Dos temas en reescritura: combinadores para cálculos con patrones e isomorfismo de Curry-Howard para la Lógica de Pruebas

Steren, Gabriela

2014-12-15

Universidad de Buenos Aires
Facultad de Ciencias Exactas y Naturales
Departamento de Computación

Dos temas en reescritura: combinadores para cálculos con patrones e isomorfismo de Curry-Howard para la Lógica de Pruebas

Tesis presentada para optar al título de Doctor de la Universidad de Buenos Aires en el área de Ciencias de la Computación

Gabriela Steren

Director de tesis: Dr. Eduardo Bonelli

Consejero de Estudios: Dr. Alejandro Ríos

Buenos Aires, 2014
Fecha de defensa: 15/12/14

Dos temas en reescritura: combinadores para cálculos con patrones e isomorfismo de Curry-Howard para la Lógica de Pruebas

Pattern matching es una herramienta fundamental de la programación funcional. Permite describir conjuntos de datos que tienen una misma forma (a través de una expresión llamada "patrón"). Esta facilidad ha comenzado a adoptarse también en otros paradigmas, y ha demostrado su utilidad para analizar datos en diversos formatos, como por ejemplo datos semi-estructurados. Los *cálculos de patrones* son lenguajes minimales basados en cálculo lambda en los que se introducen formas sofisticadas de *pattern matching* para estudiar sus fundamentos formales. La primera parte de esta tesis propone una contribución a este campo, desarrollando una lógica combinatoria para λP, un cálculo de patrones donde cualquier término puede ser un patrón. La lógica combinatoria carece de variables ligadas. Nos encontramos ante dos desafíos. Por un lado, tratar con las variables ligadas en los patrones, ya que una abstracción es un patrón válido en λP. Para esto contamos con la guía de la lógica combinatoria estándar. El segundo desafío consiste en computar, en un escenario combinatorio, la contraparte de la sustitución obtenida ante un *matching* exitoso. Esto requiere la introducción de reglas capaces de descomponer las aplicaciones. Proponemos una lógica combinatoria que logra este propósito, y estudiamos sus propiedades salientes y extensiones, incluyendo una presentación tipada y la representación de estructuras de datos.

La segunda parte de esta tesis se centra en la interpretación computacional de la Lógica de Pruebas, o LP, via el isomorfismo de Curry-Howard. LP, dada a conocer por Artemov en 1995, es un refinamiento de la lógica modal en la cual la modalidad $\Box A$ es revisitada como $[\![t]\!]A$, donde t es una expresión que testifica sobre la validez de A. Es aritméticamente correcta y completa, puede realizar todos los teoremas de $\mathsf{S4}$ y posee la capacidad de versar sobre sus propias pruebas ($\vdash A$ implica $\vdash [\![t]\!]A$, para algún t). Nuestra contribución principal es una formulación en Deducción Natural con buen comportamiento, desarrollada con el fin de develar las metáforas computacionales que surgen de esta capacidad de reflexión de LP. Esta es la primera formulación en Deducción Natural capaz de capturar a LP en su totalidad. Para esto, adoptamos la Deducción Natural Clásica de Parigot y la unimos al razonamiento hipotético. Como resultado, obtenemos una presentación en Deducción Natural de LP proposicional, para la cual se demuestran ciertas propiedades claves. Luego extendemos nuestro análisis al caso de primer orden, presentando FOHLP, una extensión de primer orden de HLP. Nuestro punto de partida es una reciente formulación de primer orden de LP, llamada FOLP, que goza de corrección aritmética y tiene una semántica de demostrabilidad exacta (la completitud es inalcanzable dado que no es finitamente axiomatizable). Presentamos una formulación en Deducción Natural llamada FOHLP, traducciones desde y hacia FOLP, una asignación de términos (cálculo lambda) y una prueba de terminación del proceso de normalización de derivaciones.

Palabras claves: Patrones, Cálculo Lambda, Lógica Combinatoria, Lógica Modal, Isomorfismo de Curry-Howard, Lógica de Pruebas

Two topics in Rewriting: Combinators for Pattern Calculi and Curry-Howard for the Logic of Proofs

Pattern matching is a basic building block on which functional programming depends, where the computation mechanism is based on finding a correspondence between the argument of a function and an expression called "pattern". It has also found its way into other programming paradigms and has proved convenient for querying data in different formats, such as semi-structured data. In recognition of this, a recent effort is observed in which pattern matching is studied in its purest form, namely by means of *pattern calculi*. These are lambda calculi with sophisticated forms of pattern matching. The first part of this two part thesis proposes to contribute to this effort by developing a combinatory logic for one such pattern calculus, namely λP. We seek to mimic the computational process of λP where arguments can be matched against *arbitrary* terms, *without* the use of variables. Two challenges must be met. On the one hand, dealing with bound variables in patterns. Indeed, an abstraction is a valid pattern in λP. Here the standard combinatory logic will provide guidance. The second is computing the counterpart, in the combinatory setting, of the substitution that is obtained in a successful match. This requires devising rules that pull applications apart, so to speak. We propose a combinatory logic that serves this purpose and study its salient properties and extensions including typed presentations and modeling data structures.

In the second part, we are concerned with the computational interpretation of a particular modal logic, the Logic of Proofs or LP, via the Curry-Howard isomorphism. LP, introduced by Artemov in 1995, is a refinement of modal logic in which the modality $\Box A$ is revisited as $[\![t]\!]A$ where t is an expression that bears witness to the validity of A. It enjoys arithmetical soundness and completeness, can realize all $\mathsf{S4}$ theorems and is capable of reflecting its own proofs ($\vdash A$ implies $\vdash [\![t]\!]A$, for some t). Esta es la primera formulación en Deducción Natural capaz de capturar a LP en su totalidad. Our main contribution is a well-behaved Natural Deduction presentation, developed with the aim of unveiling the computational metaphors which arise from the reflective capabilities of LP. This is the first Natural Deduction formulation capable of proving all LP-theorems. For that, we adopt Parigot's Classical Natural Deduction and merge it with a hypothetical reasoning which guide the construction of the inference schemes. As an outcome we obtain a Natural Deduction presentation of propositional LP for which a number of key properties are shown to hold. We then extend our analysis to the first-order case, introducing FOHLP, a first-order extension of HLP. Our point of departure is a recent first-order formulation of LP, called FOLP, which enjoys arithmetical soundness and has an exact provability semantics (completeness is unattainable given that a complete FOLP is not finitely axiomatizable). We provide a Natural Deduction presentation dubbed FOHLP, mappings to and from FOLP, a term assignment (λ-calculus) and a proof of termination of normalisation of derivations.

Keywords: Patterns, Lambda Calculus, Combinatory Logic, Modal Logic, Curry-Howard Isomorphism, Logic of Proofs

Agradecimientos

Quiero agradecer a los jurados Marcelo Coniglio, Carlos Areces y Mauricio Ayala-Rincón, a mi familia, a mi primer director Ariel Arbiser, al grupo LoReL, a la cátedra de Paradigmas de Programación, y también a Ezequiel Oliva, Facundo Nahuel Nowicky, Christian López, Francisco Soulignac, Raquel Bernator, Nilda Elías, Diego Tajer, Natalia Di Pace, Delia Kesner, Roel de Vrijer, Valeria de Paiva, Leika y Flopy, por la ayuda que me brindaron, cada quien a su manera.

Contents

Chapter 1

Introduction

The λ**-calculus** was introduced by Alonzo Church in the late 1940s [Chu36, Chu40, Chu41] as part of a larger effort to develop a system of formal logic. In 1936, Kleene [K+36], a PhD student of Church, proved the equivalence between λ-definability and Gödel-Herbrand recursiveness, and Turing showed its equivalence with his own notion of computability [Tur37]. The λ-calculus thus emerged as a concise model of computation based on the well-known concept of **function**. Abstraction of variables in an expression gives rise to such functions. Consider the expression:

$$x + 2 * x$$

By abstracting away the indeterminate "x" one produces the formal expression:

$$\lambda x.x + 2 * x \tag{1.1}$$

This expression denotes a function whose formal parameter is "x" and whose body is "$x+2*x$". An associated theory of equality over expressions is given by the fundamental notion of replacing a formal parameter by its argument:

$$(\lambda x.M)\, N =_\beta M\{x \leftarrow N\} \tag{1.2}$$

where $M\{x \leftarrow N\}$ denotes the replacement of every free occurrence of x in M by N. For example:

$$(\lambda x.x + \underline{2} * x)\underline{3} =_\beta \underline{3} + \underline{2} * \underline{3} =_\beta (\lambda x.\underline{3} + x * \underline{3})\underline{2}$$

where $\underline{2}$ and $\underline{3}$ are numerals corresponding to 2 and 3, respectively. The full set of expressions over which (1.2) is formulated, the λ-expressions, is given by the grammar:

$$M, N \quad ::= \quad x \mid M\, N \mid \lambda x.M$$

By orienting (1.2) from left to right, a notion of **reduction step** $\rightarrow_\beta$ arises. Reduction of λ-expressions may then be represented as a sequence of reduction steps. For example, $(\lambda x.\lambda y.yx)S\underline{0}$ reduces in two steps to $S\underline{0}$:

$$(\lambda x.\lambda y.yx)\underline{0}S \rightarrow_\beta (\lambda y.y\underline{0})S \rightarrow_\beta S\underline{0}$$

The expression $S\underline{0}$ cannot be **reduced** any further. Stated otherwise, there is no M such that $S\underline{0} \to_\beta M$. This expression is thus dubbed a **normal form**. Intuitively, normal forms represent the final result of a computation. This model of computation underlies the functional programming paradigm whereby the programmer specifies a number of top-level declarations of the form $f \doteq \lambda x_1.\ldots.\lambda x_n.M$ and then presents an expression (possibly containing occurrences of the user-defined functions) to be reduced to normal form (or some similar notion of result).

An independent effort at revisiting the foundations of formal logic, and motivated, in particular, by the desire to eliminate the notion of variables and substitution in formal logic systems, was the introduction of **Combinatory Logic** (*CL*) by Schönfinkel [Sch24] in 1924. A deductive theory was added to *CL* by Curry in 1930 [Cur30]. Curry, Feys, Hindley and Seldin have studied *CL* thoroughly [Cur34, Cur69, CF58, CSH72]. In *CL*, there are *CL*-expressions and an associated theory of *CL*-expression equivalence. *CL*-expressions are constructed from some number of predefined constants (**combinators**), typically K, S and I, and a binary *CL*-expression constructor Ap (application):

$$M, N ::= x \mid K \mid S \mid I \mid Ap(M, N)$$

For the sake of readability Ap is usually omitted, so that the *CL*-expression $Ap(M, N)$ is written $M\ N$. The theory of *CL*-expression equivalence is given by the following equations (application is assumed to be left-associative):

$$\begin{array}{lcl} Sxyz & \doteq & xz(yz) \\ Kxy & \doteq & x \\ Ix & \doteq & x \end{array}$$

For instance, $SKIK$ may be proved to be equivalent in this theory to K:

$$SKIK \doteq KK(IK) \doteq K$$

Soon after its introduction, *CL* and λ-calculus were noted to be very close. Indeed, Rosser [Ros35] proved that they are in fact equivalent. There are mappings between λ-calculus and *CL*, namely $\cdot_{CL}$ and $\cdot_\lambda$, which enjoy the following properties:

1. For all *CL*-expressions M and N: if $M \to_{CL} N$, then $M_\lambda \twoheadrightarrow_W N_\lambda$.
2. For all λ-expressions M and N: if $M \to_W N$, then $M_{CL} \twoheadrightarrow_{CL} N_{CL}$.
3. For every λ-expression M: $(M_{CL})_\lambda \twoheadrightarrow_\beta M$.

Here $\twoheadrightarrow_W$ stands for the reflexive transitive closure of the **weak reduction** relation [ÇH98], a restricted β-reduction in which redexes can only be contracted if their free variables are not bound by an outer λ. $\to_\beta$ means β-reduction in the λ-calculus, $\to_{CL}$ means reduction in *CL*, and $\twoheadrightarrow_\beta$ and $\twoheadrightarrow_{CL}$ denote the reflexive transitive closures of the respective reductions. These notions and results are studied in Sections 2.1 and 2.4. *CL* was thus shown to also be Turing-complete and this gave *CL* an equal standing over the λ-calculus as a theory of computability.

Nowadays, the interest in Combinatory Logic is also spawned by applications in Computer Science, particularly in relation to compilation of functional programming languages. These

applications date back to early work in the 50s [Cur52, Fit58], passing through Landin's SECD-machine, Böhm and Gross' CUCH language [BG66], the categorical combinators of Curien [Cur86, Cur93, CCM87] and the G-Machine (and variants) used in compilation of Haskell [Joh83, Aug84, Kie85, BPJR88, PJS89, HK84, PJHH+93].

1.1 Patterns in Functional Programming

Pattern matching is a basic building block on which all modern functional programmers depend on. A simple example of a program that relies on patterns is the `length` program for computing the length of a list (the code is written in Haskell).

```
length []     = 0
length (x:xs) = 1 + length xs
```

Another example is the following, which exhibits the use of nested patterns, `Salary` being a data constructor having an employee id and his/her salary as arguments:

```
update []                       f = []
update ((Salary id amount):xs) f = (Salary id (f amount)): update xs f
```

In recognition of their usefulness and in an effort to study patterns and their properties in their most essential form, a number of so called **pattern calculi** have recently emerged [CK01, Jay01, WH08, JK09, AMR06, Jay09, JGW11]. As a representative example, we briefly describe the pioneering λP [vO90] recently revisited in [KvOdV08] (cf. Section 3.1 for detailed definitions). In λP the standard functional abstraction $\lambda x.M$ is replaced by the more general $\lambda P.M$:

$$M, N, P ::= x \mid MN \mid \lambda P.M$$

The pattern P may be any term at all. In particular, it may be a variable, thus subsuming the standard functional abstraction of the λ-calculus. Examples of λP-terms are $\lambda(\lambda x.y).y$, $\lambda z.\lambda(\lambda x.x).\lambda y.y$, $(\lambda(\lambda x.y).y)(\lambda w.z)$, and of course all the terms of the λ-calculus. The pattern specifies which form the argument must have in order to match. A function can only be applied to an argument which is an instance of its pattern. Application is then performed by substituting the terms bound to the free variables in the pattern into the function body:

$$(\lambda P.M)P^\sigma \to_{\beta_P} M^\sigma$$

Here σ denotes a substitution from variables to terms, and P^σ the result of applying it to P. Note that in the case that P is a variable, we obtain $\to_\beta$. An example of a λP-reduction step is:

$$(\lambda xy.x)((\lambda x.x)(zw)) \to_{\beta_P} \lambda x.x \tag{1.3}$$

Another example is the reduction step from the term $(\lambda(\lambda x.y).y)(\lambda z.w)$. The pattern $\lambda x.y$ can be matched by any constant function, and the result of the application of $\lambda(\lambda x.y).y$ to an argument of the form $\lambda z.M$ (with $z \notin \mathsf{FV}(M)$) will be M (in this case, $M = w$). However,

the application $(\lambda(\lambda x.y).y)z$ does not reduce, since the argument z does not match the pattern $\lambda x.y$. Note that $(\lambda(\lambda x.y).y)(\lambda z.z)$ does not reduce either, since $\lambda z.z$ is not an instance of $\lambda x.y$ (variable capture is not allowed). This situation, where reduction is *permanently* blocked due to the argument of an application not matching the pattern, is known as **matching failure**. There are cases where an argument does not yet match a pattern, but can reduce to a term which does. For example, $(\lambda(\lambda x.y).y)\,((\lambda x.x)\,(\lambda z.w))$. In this case, we say that the matching is still **undecided**.

A key issue is establishing the conditions under which confluence holds for λP, since the unrestricted calculus is not confluent. For example, the following reduction starts from the same λP-expression as (1.3) but ends in a different normal form:

$$(\lambda xy.x)((\lambda x.x)(zw)) \to_{\beta_P} (\lambda xy.x)(zw) \to_{\beta_P} z$$

1.1.1 Combinators for Pattern Calculi

In the same way that *CL* shows how one may compute without variables in the λ-calculus, we seek to address a similar property for *pattern calculi*. This would entail that variables and substitution in pattern calculi may be compiled away while preserving the reduction behavior. For this purpose we fix λP as study companion and delve into the task of formulating a corresponding combinatory logic. This is the topic of the Chapter 3 of this thesis.

In order to motivate CL_P, the combinatory system we introduce in Chapter 3, recall the standard translation from the λ-calculus to *CL*:

$$\begin{array}{rcl} x_{CL} & \triangleq & x \\ (MN)_{CL} & \triangleq & M_{CL}N_{CL} \\ (\lambda x.M)_{CL} & \triangleq & [x].M_{CL} \end{array}$$

where "$\triangleq$" denotes definitional equality and $[x].M$ is defined recursively as follows:

$$\begin{array}{lcll} [x].x & \triangleq & SKK & \\ [x].M & \triangleq & KM, & \text{if } M = K, S, I \text{ or } M = y \neq x \\ [x].(MM') & \triangleq & S([x].M)([x].M') & \end{array}$$

Note that abstractions translate to terms of the form KM or SM_1M_2. Therefore, the application of an abstraction to an argument will translate to either KMN or SM_1M_2N with N being the translation of the argument. In CL_P, applications of an abstraction to an argument in λP will translate to terms of the form K_PMN or $S_PM_1M_2N$, N being the translation of the argument. The expressions K_P and S_P are combinators of CL_P. The full grammar of CL_P-expressions is:

$$M, N ::= x \mid K_M \mid S_M \mid \Pi^1_{MN} \mid \Pi^2_{MN} \mid MN$$

Subscripts of combinators in CL_P are an integral part of them. Combinators K_P and S_P will behave similarly to K and S but with one important difference: while K and S always form redexes when applied to the right number of arguments (2 and 3 respectively), K_P (resp. S_P) will check its second (resp. third) argument against the pattern P, and will only form a redex if the *match* is successful. Of importance is to note that "match" here means matching in

the combinatory setting, that is *first-order matching*, which is much simpler than in λP (more details towards the end of this section). For example, $K_{S_y} x S_y$ is a redex, but $K_{S_y} x K_y$ is not. Note that the behaviour of the standard *CL*-combinators K and S is given by that of K_P and S_P, resp., when P is any variable. For this reason, we usually abbreviate K_x, for any variable x, as K (and analogously for S).

Combinators of the form K_P and S_P alone are not expressive enough to model reduction in λP for they lack the ability to pull applications apart. Since the pattern P in $\lambda P.M$ will be translated to an application in CL_P (except for the case in which P is a variable) we need to be able to define functions which expect arguments of the form MN and take them apart, returning terms like M, N and $S(KN)M$. For instance, in order to translate $\lambda(\lambda x.y).y$, we need a term which can access the variable y from the translation of $\lambda x.y$, which is Ky. To that effect, we define the combinators Π^1_{PQ} and Π^2_{PQ}, known as **projectors**. $\Pi^1_{PQ}(MN)$ reduces to M when M matches P and N matches Q. Analogously, $\Pi^2_{PQ}(MN)$ reduces to N under the same conditions.

Some examples of the translation we shall introduce in Chapter 3 follow. λ-terms translate to *CL*-terms as per the original translation. Abstractions with non-variable patterns translate to CL_P-terms where the head combinator (the leftmost combinator) is decorated with a pattern. For example, $\lambda(\lambda x.x).y$ translates to $K_{SKK}y$, and $\lambda(\lambda x.y).y$ translates to $S_{Ky}(K(SKK))\Pi^2_{Ky}$.

Consider the λP-term $(\lambda(\lambda x.x).y)(\lambda z.z)$, which matches $\lambda z.z$ against the pattern $\lambda x.x$ and, since the match is successful, reduces in one step to y. This term can be translated as $K_{SKK}y(SKK)$. The subscript SKK in K_{SKK} is a pattern (it is, in fact, the translation of the pattern $\lambda x.x$) which must be matched by the second argument of K_{SKK}, namely SKK, the translation of $\lambda z.z$. In this case matching is successful, and thus the translated term reduces to y, just like the original λP-term.

More complex terms may require more steps to reduce. For instance, $(\lambda(\lambda x.y).y)(\lambda w.z)$, which in λP reduces to z in one step, is translated to $S_{Ky}(K(SKK))\Pi^2_{Ky}(Kz)$. The latter requires several steps to reach a normal form:

$$\begin{array}{rcl} S_{Ky}(K(SKK))\Pi^2_{Ky}(Kz) & \to & K(SKK)(Kz)(\Pi^2_{Ky}(Kz)) \\ & \to & SKK(\Pi^2_{Ky}(Kz)) \\ & \to & K(\Pi^2_{Ky}(Kz))(K(\Pi^2_{Ky}(Kz))) \\ & \to & \Pi^2_{Ky}(Kz) \to z \end{array}$$

Matching failure is also carried over to the translation: $(\lambda(\lambda x.x).y)(\lambda w.z)$ translates to $K_{SKK}y(Kz)$, which also contains a matching failure, since Kz does not match the pattern SKK.

While at first sight it may seem as if having pattern matching requires substitutions to be computed (after all, a term M matches a pattern P if and only if there is a substitution σ such that $M = P^\sigma$), actually computing a substitution is not necessary in our setting. All the matching algorithm has to do is decompose the pattern and the argument and check that, wherever there is a combinator in the pattern, the same combinator is present (in the same position) in the argument. Variables in patterns are automatically matched by any argument.

Another reason why matching is easier to compute in a combinator-based setting is the lack of binders. For example, in λP, the term $\lambda x.x$ does not match the pattern $\lambda x.y$, since there

is no substitution which applied to $\lambda x.y$ returns $\lambda x.x$ (the convention set in [vO90] does not allow variable capture). However, this means that a matching algorithm would have to explicitly check that a variable not occur free in a term. On the other hand, if we translate the $\lambda x.y$ and $\lambda x.x$ into our combinatory logic system, we obtain Ky and SKK, and now matching failure is immediate from the fact that SK does not match K.

The following depicts, in a bird's eye view, how CL_P and variations relate to λP and its variations and also to SF (discussed below). All referred results are developed in Chapter 3, the first part of this two-part thesis:

$$\begin{tikzcd}
\lambda P \arrow[d, leftrightarrow, "Cor.\ 3.3.15, Prop.\ 3.3.18"'] & & \lambda C \arrow[d, "Cor.\ 3.6.15"] \\
CL_P \arrow[rr, "(\subset)"] \arrow[dr, "Prop.\ 3.7.10"'] & & CL_P + \theta & SF \arrow[l, dotted, "Rem.\ 3.6.16"] \\
& CL_\star & &
\end{tikzcd}$$

λP can be translated into CL_P while preserving (weak) reduction (Corollary 3.3.15) and the same holds in the reverse sense (Proposition 3.3.18). λC [KvOdV08] is a variation of λP in which patterns are taken to be algebraic terms and also in which multiple pattern matching is allowed:

$$(\lambda P_1.M_1|\ldots|\lambda P_k.M_k)P_i^\sigma \to M_i^\sigma$$

Constructors may be added to CL_P together with a new set of rules ($CL_P + \theta$) and the resulting system serves as target of a reduction preserving translation from λC. The reverse direction should hold as well (the results we develop have not required us to do so though). Although CL_P is a first-order term rewriting system composed of a finite number of rule schemas, an infinite number of rules result from instantiating these rule schemas. $CL_\star$ is a variation of CL_P which is indeed a finite term rewriting system, at the cost of a more verbose system. Finally, we briefly comment on SF [JGW11], a combinatory logic similar in spirit to ours (see Section 3.8 for an in-depth comparison, and Section 3.6.3 for a partial translation from SF to $CL_P + \theta$ with additional patterns and generalized pattern matching). There is no possible translation from $CL_P + \theta$ to $CL_\star$, since multiple matching cannot be handled by an explicit matching mechanism within the setting of a TRS. For details we refer the reader to Section 3.7.4.

1.2 The Curry-Howard Isomorphism and the Logic of Proofs

A striking resemblance between logical propositions and types from the typed lambda calculus was observed by H. Curry and W. Howard while trying to develop a suitable notion of construction for the interpretation of intuitionistic mathematics. Curry [CF58] noted that there is

a close connection between the types of the axioms of the implicational fragment of propositional Intuitionistic Logic and combinators in *CL*. For example, the type of the combinator K corresponds to the axiom $A \supset B \supset A$. Tait [Tai65] discovered a close correspondence between cut-elimination and reduction of λ-terms. In 1969, Curry and Howard completed the analogy by showing the correspondence between λ-terms and intuitionistic derivations. The original notes have been summarized by Howard [How95].

From the perspective of this correspondence, a logical proposition $A \supset B$ may be viewed as the functional type expression $A \to B$ and vice versa. Similarly, consider the following Natural Deduction proof of $A \supset A$:

$$\dfrac{\dfrac{\dfrac{}{A \vdash A}\,\mathsf{Ax}}{\vdash A \supset A}\,\supset\mathsf{I} \qquad \dfrac{\dfrac{}{C \vdash C}\,\mathsf{Ax}}{\vdash C \supset C}\,\supset\mathsf{I}}{\dfrac{\vdash (A \supset A) \wedge (C \supset C)}{\vdash A \supset A}\,\wedge_{\mathsf{E1}}}\,\wedge_{\mathsf{I}}$$

It may be encoded as a (typed) λ-expression $\pi_1 \langle (\lambda x^A.x), (\lambda y^C.y) \rangle$, where $\langle \bullet, \bullet \rangle$ denotes the pair expression constructor and π_1 the first projection expression constructor (cf. Section 2.5 for further details).

This resemblance was the starting point of a deep analogy between logic and computation, most clearly exhibited by the tight correspondence between systems of Natural Deduction and typed lambda calculi. Logic and computation turn out to be two sides of the same coin. Concepts such as redexes, substitution, term reduction, exceptions, continuations, resource consumption, staged computation, abstract machines, reduction strategies, concurrency, etc. all turn out to have logical counterparts. In this thesis we focus on the particular case of modal logic.

Modal logic is a type of symbolic logic which extends propositional or first order logic to include operators expressing modality (*modal operators*). A modality represents an expression (like 'necessarily' or 'possibly') that is used to qualify the truth of a judgement. Modal logic, in its original formulation, is the study of the deductive behavior of the expressions 'it is necessary that' (represented by the "$\Box$" operator) and 'it is possible that' (represented by "$\Diamond$"). However, the term *modal logic* is often used more broadly for a family of related systems. These include logics for knowledge, for belief, for tense and other temporal expressions, for the deontic (moral) expressions such as 'it is obligatory that' and 'it is permitted that', and many others. Among the plethora of modal logic systems, we single out Lewis' $\mathsf{S4}$[1] given its close correspondence with the logic we shall study in this thesis. The language of $\mathsf{S4}$ is:

$$A, B \quad ::= \quad P \mid A \supset B \mid \Box A$$

where P ranges over propositional variables. It consists of the following axiom and inference schemes:

[1] This presentation of $\mathsf{S4}$ is due to Gödel [Göd33]

A0.	Axiom schemes of classical propositional logic
T.	$\Box A \supset A$
K.	$\Box(A \supset B) \supset (\Box A \supset \Box B)$
4.	$\Box A \supset \Box\Box A$
MP.	$\vdash A \supset B$ and $\vdash A \Rightarrow \vdash B$
Nec.	$\vdash A \Rightarrow \vdash \Box A$

In one of numerous attempts at interpreting intuitionistic truth in terms of classical provability (cf. [Art01]), early work of Orlov [Orl25] and Gödel [Göd33] suggested prefixing every subformula in Int (Intuitionistic Propositional Logic) with "$\Box$", where "$\Box$" is subject to the fundamental laws stated in $\mathsf{S4}$. Gödel then established that the abovementioned translation of formulas which are provable in Int are provable in $\mathsf{S4}$. Later, McKinsey and Tarski showed that this embedding is *faithful* (i.e. the translation of formulas which are not provable in Int are not provable in $\mathsf{S4}$). However, in order to complete the explanation of intuitionistic truth in terms of classical provability, it is necessary to relate the "$\Box$" modality and provability in Peano Arithmetics (PA):

$$\mathsf{Int} \hookrightarrow \mathsf{S4} \hookrightarrow \ldots ? \ldots \hookrightarrow \mathsf{PA} \tag{1.4}$$

However, reading "$\Box A$" as "$\exists x.\mathit{Proof}(x, \ulcorner A \urcorner)$", where "$\ulcorner A \urcorner$" denotes an appropriate numeric encoding of A, is problematic since the $\mathsf{S4}$ theorem $\Box(\neg\Box\,\bot)$, which expresses the consistency of PA, is provable in PA. This situation was observed by Gödel [Göd33], who posed two problems:

1. Uncover the modal logic of formal provability predicate $\exists x.\mathit{Proof}(x, \ulcorner A \urcorner)$.
2. Devise the exact intended provability semantics for $\mathsf{S4}$ (cf. "?" in (1.4)).

Both of these problems have been solved. In the first case we have Solovay's completeness theorem [Sol76] of Löb's logic (or GL). The second problem is solved by means of Artemov's Logic of Proofs (LP).

1.2.1 The Logic of Proofs

The Logic of Proofs is a modal logic which later gave birth to the family of Justification Logics [Art08a, Art08b]. It was introduced by Sergei Artemov in [Art95, Art01] as an answer to the fundamental question of how to fill in the abovementioned gap:

$$\mathsf{Int} \hookrightarrow \mathsf{S4} \overbrace{\hookrightarrow}^{a} \mathsf{LP} \overbrace{\hookrightarrow}^{b} \mathsf{PA} \tag{1.5}$$

LP arises essentially from skolemizing the existential quantifier which is implicit in the $\Box$ operator. That is, to replace statements of the form $\Box A$ (read as *"there is some proof of A"*) by:

$$[\![t]\!]A$$

which is read as *"t is a proof of A"*. The expression t is called a **proof polynomial**, and belongs to the set of expressions specified by the following grammar:

$$s, t \quad ::= \quad x \,|\, c \,|\, s \cdot t \,|\, !s \,|\, s + t$$

Proof polynomials are constructed from proof variables, proof constants, application, bang and sum. The axiom and inference schemes of LP are as follows:

A0. Axioms of classical propositional logic in the language of LP
A1. $[\![s]\!]A \supset A$
A2. $[\![s]\!](A \supset B) \supset ([\![t]\!]A \supset [\![s \cdot t]\!]B)$
A3. $[\![s]\!]A \supset [\![!s]\!][\![s]\!]A$
A4. $[\![s]\!]A \supset [\![s + t]\!]A$
A5. $[\![t]\!]A \supset [\![s + t]\!]A$
MP. $\vdash A \supset B$ and $\vdash A \Rightarrow \vdash B$
Nec. A axiom $\mathbf{A0} - \mathbf{A5} \Rightarrow \vdash [\![c]\!]A$

Note that if one discards the proof polynomials decorating these axioms, one obtains the axioms of S4 (**A4** and **A5** collapse to a trivial theorem). Returning to (1.5), the arrow marked with an (a) in (1.5) is Artemov's **realization theorem**:

Theorem 9.4 in [Art01] $\mathsf{S4} \vdash A \Rightarrow \mathsf{LP} \vdash A^r$, for some normal realization $\bullet^r$.

A **realization** is a function that decorates each occurrence of $\Box$ with a proof polynomial; it is said to be *normal* if each negative such occurrence is decorated with a different proof variable. This entails that each S4 theorem has an underlying statement about proofs. For instance: $\Box A \supset \Box B$ can be realized as $[\![x]\!]A \supset [\![t(x)]\!]B$, for an appropriate proof polynomial $t(x)$. The arrow marked (b) is Artemov's **arithmetical soundness and completeness theorem**. The correspondence (1.5) was later extended to a fragment of LP capturing provability in Heyting arithmetic (HA) [AI07, Das11].

A further salient property of LP is that it is endowed with a reflection mechanism.

Corollary 5.5 in [Art01] $\vdash A$ implies there exists a ground t s.t. $\vdash [\![t]\!]A$.

Here the proof polynomial t encodes a proof of A. The proof of Corollary 5.5 in [Art01] consists in analysing the given derivation of A in LP, and encoding it using proof polynomials.

1.2.2 Computational Interpretation of the Logic of Proofs

The second part of this thesis is a quest to unveil the computational metaphors which arise from the reflective capabilities of LP. If t is understood as a typing derivation of a term of type A, then a term of type $[\![t]\!]A$ should include some encoding of t. If this typing derivation is seen as a logical derivation, then any normalisation steps applied to it would produce a new typing derivation for A. Moreover, its relation to t would have to be made explicit in order for derivations to be closed under normalisation (in type systems parlance: for Type Preservation to hold). This suggests that the computational interpretation of LP should be a programming language that includes some explicit manifestation of type derivations/certificates/computation trails.

The first step in this quest is to devise a well-behaved Natural Deduction presentation. Following recent work on *judgemental reconstruction* of intuitionistic S4 [DP96, DP01b, DP01a, ML84], we introduce judgements in which a distinction is made between propositions whose *truth* is assumed from those whose *validity* is assumed:

$$v_1 : A_1 \ valid, \ldots, v_n : A_n \ valid; a_1 : B_1 \ true, \ldots, a_m : B_m \ true \vdash A \ true|s$$

This judgement expresses that: "*s is evidence that A is true, assuming that for each $i \in 1..n$, v_i is evidence that A_i is valid and assuming that for each $j \in 1..m$, a_j is evidence that B_j is true*". Such judgements are called *hypothetical judgements* [ML84]. They are abbreviated:

$$\Theta; \Gamma \vdash A|s \tag{1.6}$$

where Θ is a context of validity assumptions and Γ is a context of truth assumptions; also the qualifiers "valid" and "true" are dropped. Evidence s is a constituent part of the judgement without which the proposed reading would not be possible.

Meanings to such judgements are provided for a fragment of LP based on minimal logic and without axioms **A4** and **A5** in [AB07, BB10, BB14]. The following rather natural inference scheme is shown not to yield a well-behaved explanation for $\Box$ (cf. discussion in Section 4.2.1):

$$\frac{\Theta; \cdot \vdash B \mid s}{\Theta; \Gamma \vdash [\![s]\!]B \mid !s} \Box\mathsf{I}'$$

Indeed, it is shown to yield a system in which Type Preservation fails. Op.cit. replaces it by the inference scheme:

$$\frac{\Theta; \cdot \vdash B \mid s \quad \Theta; \Gamma \vdash s \equiv t : B}{\Theta; \Gamma \vdash [\![t]\!]B \mid !t} \Box\mathsf{I}$$

which appeals to a new judgement of proof witness equivalence. Some efforts were developed to recognize underlying programming metaphors of the resulting Natural Deduction presentation via the proof normalisation-as-reduction methodology:

- An *intensional lambda calculus* where information on *how* a result is computed (cf. Lévy labelling [Lév78]) emerges, rather than just *what* the result is [AB07].
- A calculus with *computation trails* [BB10, BB14], where the judgement $\Theta; \Gamma \vdash s \equiv t : B$ encoded and reflected in the term assignment for $\Box\mathsf{I}$, is understood as a computation trail or computation history with applications to modeling history-based access control [AF03] and history-based information flow [BN05].
- A calculus of *certified mobile units* [BF12] which enriches mobile code with certificates (representing type derivations). Such units take the form $\mathsf{box}_s\, M$, s being the certificate and M the executable. Composition of certified mobile units allows one to build mobile code out of other pieces of mobile code *together* with certificates that are also composed out of other certificates.

Unfortunately, all of the above fall short in that none address proof normalisation as computation for *full* LP. That is, classical propositional LP, including all its axioms. This is the subject matter of Chapter 4. The work plan is therefore to incorporate notions of computation which arise from the Curry-Howard interpretation of classical propositional logic into the above mentioned analysis of hypothetical reasoning over LP. One such notion is that of Parigot's Classical Natural Deduction [Par92, Par97, Sau10] (see overview in Section 2.5.3). It has a well-known computational interpretation and is formulated in the setting of (a variation of) Natural Deduction and thus proves suitable for our purposes. Judgement (1.6) is therefore replaced by:

$$\Theta; \Gamma; \Delta \vdash A \mid s$$

where Δ is a context of negated formulas. A Natural Deduction presentation for full LP involving such judgements is developed in Chapter 4, together with proofs of its salient properties.

The First-Order Case

Given a first-order language $\mathcal{L}$, the language of the First-Order Logic of Proofs (FOLP) is obtained by extending $\mathcal{L}$ with proof variables and functional symbols for operations on proofs. The set of formulas is extended with a skolemized version of the modal operator $\Box$. A crucial aspect is how parameters are understood in this skolemized version. Consider the formula $\Box A(x)$, where A has a parameter x. This parameter can play one of two roles in a *proof* of $A(x)$. It can be interpreted as a *global parameter*. Global parameters are placeholders and, as such, may be substituted by any term at all. For example, in the following derivation $\pi(y)$, where R is some binary predicate letter:

$$\dfrac{\dfrac{\forall x, y.R(x,y) \supset R(y,x)}{R(E,y) \supset R(y,E)} \qquad R(E,y)}{R(y,E)}$$

the variable y acts as a global parameter since it may be substituted for any first-order expression F in order to obtain a proof $\pi(F)$ of $A(F)$. However, parameters can also play a different role, namely that of *eigenvariables*: syntactic objects subject to generalization. For example, consider the derivation:

$$\dfrac{\pi(y)}{\forall y.\forall x.R(x,y)}$$

where the derivation $\pi(y)$ is:

$$\dfrac{\dfrac{\dfrac{\forall x.\forall y.R(x,y)}{\forall y.R(x,y)}}{R(x,y)}}{\forall x.R(x,y)}$$

The parameter y here is not meant to be substituted for; rather it acts as a fresh scoped constant. These two distinct roles have been identified in Computer Science in the context of proof assistants where reasoning over open objects is explored ([GJ02] and the citations therein). See also the discussion on proving universally quantified expressions using an extensional versus intensional approach of Miller and Tiu [MT05].

The above considerations leads to the following skolemized modal operator, proposed in [AY11], which allows both interpretations to be accounted for:

$$[\![s]\!]_{\Xi} A$$

Here Ξ is a set of variables and determines the role that a variable plays in a proof of A. Variables in Ξ play the role of global parameters in A and hence in s (which encodes a proof of A). Variables that occur in A but that are *not* in Ξ are understood as eigenvariables. These are therefore understood to be implicitly bound in A: the set of free variables of $[\![s]\!]_\Xi A$ – denoted as $\mathsf{FIV}([\![s]\!]_\Xi A)$ – is defined to be Ξ. Two additional results of the axioms associated to this new operator are realization of first order $\mathsf{S4}$ ($\mathsf{FOS4}$) and reflection (called "internalization" in [AY11]):

Theorem 2 in [AY11] $\mathsf{FOS4} \vdash A \Rightarrow \mathsf{FOLP} \vdash A^r$, for some normal realization $\bullet^r$.

Theorem 1 in [Art01] Let $\Xi = \Xi_0 \cup \ldots \cup \Xi_n$. Then $[\![x_0]\!]_{\Xi_0} A_0, \ldots, [\![x_n]\!]_{\Xi_n} A_n \vdash A \Rightarrow \exists t.\ [\![x_0]\!]_{\Xi_0} A_0, \ldots, [\![x_n]\!]_{\Xi_n} A_n \vdash [\![t(x_0, \ldots, x_n)]\!]_\Xi A$.

It should be mentioned that FOLP is known not to be recursively axiomatizable [AY01]. This does not mean that one cannot formulate a presentation of FOLP that enjoys an exact provability semantics and that is capable, at the same time, of realizing the full set of first-order $\mathsf{S4}$ theorems. This has in fact recently been accomplished by Artemov and Yavorskaya [AY11] (details supplied in Section 5.1). However, any hope of arithmetical completeness must be given up. Nonetheless, completeness with respect to a different semantics, namely Kripke semantics, has been established by Fitting [Fit14]. One could also consider quantifiers over proof variables. Such a system was studied in [Yav01] and shown not to be axiomatizable. Also related is [WWdRZ02] where the parameter x in the formula $\Box A(x)$ is assumed bound (coined "binding interpretation" in op.cit.). This system is shown to have a complete axiomatization, however it does not suffice to realize first-order modal logic.

Chapter 5 is devoted to developing a Natural Deduction presentation of FOLP. Soundness and completeness w.r.t. the system of [AY11] is proved, an analysis of proof normalisation developed and strong normalisation is also proved.

1.3 Structure of the Thesis and Summary of Contributions

In Chapter 2 we define some basic notions which shall be used throughout this work. It has the overall aim of fixing notation. The rest of the thesis, disregarding Chapter 6 (the final chapter, which suggests possible directions for future research), is structured in two parts which may be read independently.

- **Part I (Chapter 3).** We introduce CL_P, a Combinatory Logic enhanced with pattern-matching capabilities. This work was developed under the supervision of Ariel Arbiser; references to "we" in this chapter refer to both of us. The contributions of this part may be summarized as follows:
  - A combinatory logic (TRS, no bound variables) CL_P capable of simulating reduction in λP.
  - Conditions under which CL_P is confluent.
  - Type system for CL_P and proofs of Type Preservation and Strong Normalisation

- Extensions to CL_P, in which the meta-level matching of CL_P is added into the system itself.

Presentations: UNILOG 2010.

- **Part II (Chapters 4 and 5).** In the former, we introduce HLP, a refinement of Artemov's Logic of Proofs, based on judgemental reasoning. In the latter, we present a first order extension of HLP named FOHLP, capable of realizing first-order modal logic S4. This work was developed under the supervision of Eduardo Bonelli; references to "we" in these chapters refer to both of us. The contributions of this part may be summarized as follows:

  - Chapter 4
    * Formulation of a Natural Deduction presentation for classical, propositional LP.
    * A term assignment (λ-calculus) for this presentation.
    * Proof of termination of normalisation of derivations.

    Presentations: UNILOG 2013, IMLA 2013. Publications: Proceedings of IMLA 2013 [SB13] and *Logica Universalis* [BS14].
  - Chapter 5
    * Formulation of Natural Deduction presentation for classical, first-order LP.
    * A term assignment (λ-calculus) for this presentation.
    * Proof of termination of normalisation of derivations.

    Presentations: SLALM 2014. Publications: Submitted to *Journal of Logic and Computation*.

Chapter 2

Rewriting, Combinatory Logic and Lambda Calculus

This chapter introduces notions of rewriting which are to be used in the sequel of this thesis. Sections 2.1 and 2.2 define Abstract Reduction Systems and Term Reduction Systems, and introduce the central properties of termination and confluence, as well as critical pair and closure constructions. The focus is then placed on two particular rewriting systems, namely the Lambda Calculus (Section 2.3) and Combinatory Logic (Section 2.4). Finally, Section 2.5 revisits the Curry-Howard isomorphism, both in the intuitionistic case and in the classical one (with particular emphasis placed on Parigot's Classical Natural Deduction). Definitions are taken from [BN98], [BKdV03], [SU06], [Par92] and [Par97].

2.1 Abstract Reduction Systems

An abstract reduction system (ARS) is a pair $(\mathcal{A},\rightarrow)$ consisting of a set $\mathcal{A}$ and binary relation over $\mathcal{A}$ called **reduction**. An ARS models the step by step transformation of some object (for instance a term) into another. For $(x, y) \in \rightarrow$, we simply write $x \rightarrow y$ and say that x **reduces** to y, and conversely that y is a **reduct** of x. Also, we call the process of going from x to y a **reduction step**. The elements of $\mathcal{A}$ are called **objects**, or **terms**. We write $R_1 \circ R_2$ to denote the composition of relations R_1 and R_2. A term is x is a **normal form** if there is no y such that $x \rightarrow y$. Likewise, y is a **normal form of** x if y is a normal form and $x \twoheadrightarrow y$, where $\twoheadrightarrow$ denotes the reflexive-transitive closure of $\rightarrow$ (cf. Def. 2.1.3). We can now define two core properties of ARSs:

Definition 2.1.1 (Confluence and Termination) A reduction relation $\rightarrow$ is called:

- **confluent** if for every x, y_1, y_2: $(x \twoheadrightarrow y_1 \wedge x \twoheadrightarrow y_2) \supset \exists z(y_1 \twoheadrightarrow z \wedge y_2 \twoheadrightarrow z)$ (see fig. 2.1)
- **locally confluent** if for every x, y, z: $(x \rightarrow y_1 \wedge x \rightarrow y_2) \supset \exists z(y_1 \twoheadrightarrow z \wedge y_2 \twoheadrightarrow z)$
- **terminating** or **strongly normalizing** (SN) if there is no infinite sequence $x_0 \rightarrow x_1 \rightarrow \ldots$
- **weakly normalizing** (WN) if every term has a normal form.

$$\begin{tikzcd} & x \arrow[dl, two heads] \arrow[dr, two heads] & \\ y_1 \arrow[dr, dotted, two heads] & & y_2 \arrow[dl, dotted, two heads] \\ & z & \end{tikzcd} \tag{2.1}$$

Figure 2.1: Confluence

Lemma 2.1.2 (Newman's Lemma [New42]) A strongly normalizing (abstract) reduction system is confluent if it is locally confluent.

This way, termination reduces the problem of proving confluence to proving local confluence. Some other relations on terms derived from reduction are:

Definition 2.1.3

$\xrightarrow{0}$	$\triangleq$	$\{(x,x) \mid x \in \mathcal{A}\}$	identity
$\xrightarrow{i+1}$	$\triangleq$	$\xrightarrow{i} \circ \rightarrow$	$(i+1)$-fold composition $(i \geq 0)$
$\xrightarrow{=}$	$\triangleq$	$\xrightarrow{0} \cup \rightarrow$	reflexive closure
$\xrightarrow{+}$	$\triangleq$	$\bigcup_{i>0} \xrightarrow{i}$	transitive closure
$\twoheadrightarrow$	$\triangleq$	$\xrightarrow{+} \cup \xrightarrow{0}$	reflexive-transitive closure
$\xrightarrow{-1}$	$\triangleq$	$\{(y,x) \mid x \rightarrow y\}$	inverse
$\leftrightarrow$	$\triangleq$	$\rightarrow \cup \xrightarrow{-1}$	symmetric closure
$\equiv$	$\triangleq$	$\twoheadrightarrow \cup \xrightarrow{-1}$	equivalence (reflexive-symmetric-transitive) closure

2.2 Term Rewriting Systems

A **term rewriting system** (TRS) is an abstract reduction system where the objects are first-order terms, and where the reduction relation is presented in a standard schematic format of **reduction rules**.

2.2.1 First-order Terms and Substitutions

While the objects of an ARS may take any form, we will take particular interest in **(first-order) terms**, as they will be a recurring subject in this work, as well as serve as a basis to explain some core concepts. A **signature** Σ is a set of function symbols, where each $f \in \Sigma$ is associated with a non-negative integer n, the **arity** of f. For each $n \geq 0$, we denote the set of all n-ary elements of Σ by $\Sigma^{(n)}$. The elements of $\Sigma^{(0)}$ are also called **constant symbols**. For example, if we want to talk about natural numbers with 0, successor and addition, we can use the signature $\Sigma_{\mathbb{N}} = \{0, \mathtt{succ}, +\}$, where 0 is a constant symbol, $\mathtt{succ}$ is unary and $+$ is binary.

Definition 2.2.1 (Term associated to a signature) Let Σ be a signature and $\mathcal{X}$ a set of variables (disjoint from Σ). The set $T(\Sigma, \mathcal{X})$ of all Σ-terms over $\mathcal{X}$ is defined as follows:

- $\mathcal{X} \subseteq T(\Sigma, \mathcal{X})$.
- for all $n \geq 0$, all $f \in \Sigma^{(n)}$, and all $t_1, \ldots, t_n \in T(\Sigma, \mathcal{X})$, we have $f(t_1, \ldots, t_n) \in T(\Sigma, \mathcal{X})$.

For example, in the signature $\Sigma_{\mathbb{N}} = \{0, \mathtt{succ}, +\}$ from above, $+(\mathtt{succ}(x), 0)$ is a term which contains the variable x. For the constant symbol 0, we have written the corresponding term 0 instead of 0() (i.e., 0 applied to an empty list of terms). Binary function symbols (such as +) are sometimes written in infix notation, e.g., $(x + y) + z$ instead of $+(+(x, y), z)$. If g is a unary function symbol, we abbreviate the term $g(g(g(\ldots g(t) \ldots)))$ to $g^n(t)$ (n-fold application). We sometimes refer to $T(\Sigma, \mathcal{X})$ as $\mathrm{Ter}(\Sigma)$.

Definition 2.2.2 The set of variables in a term t, written $\mathrm{Var}(t)$, is defined inductively as:

$$\begin{aligned} \mathrm{Var}(x) &\triangleq \{x\} \\ \mathrm{Var}(f(t_1, \ldots, t_n)) &\triangleq \bigcup_{i=1}^{n} \mathrm{Var}(t_i) \end{aligned}$$

Definition 2.2.3 (Size and depth) The **size** $|t|$ of a term t is defined as follows:

$$\begin{aligned} |t| &\triangleq 1, && \text{if } t \text{ is a variable or constant symbol} \\ |f(t_1, \ldots, t_n)| &\triangleq 1 + \sum_{i=1}^{n} |t_i| \end{aligned}$$

The **depth** of a term is defined as follows:

$$\begin{aligned} \mathtt{depth}(t) &\triangleq 0, \text{ if } t \text{ is a variable or constant symbol} \\ \mathtt{depth}(f(t_1, \ldots, t_n)) &\triangleq 1 + \mathtt{max}\{\mathtt{depth}(t_i) \mid 1 \leq i \leq n\} \end{aligned}$$

The structure of a term can be illustrated by representing it as a tree, where function symbols are nodes and their children the arguments of the function. To refer to positions in a term, we use a numbering of the tree's nodes by strings of positive integers. We use the standard notion of **position**, defined as follows.

Definition 2.2.4 The set of positions of a term t is a set $\mathtt{Pos}(t)$ of strings over the alphabet of positive integers, which is inductively defined as follows:

- $\mathtt{Pos}(x) \triangleq \{\epsilon\}$, where ϵ denotes the empty string.
- $\mathtt{Pos}(f(t_1, \ldots, t_n)) \triangleq \{\epsilon\} \cup \bigcup_{i=1}^{n} i\,p | p \in \mathtt{Pos}(t_i)$.

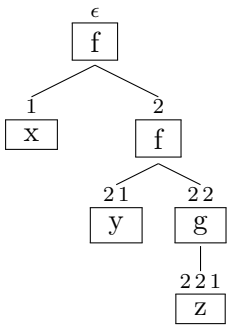

Figure 2.2: Positions of the term $t = f(x, f(y, g(z)))$

The position ϵ is called the **root position** of the term t, and the function or variable symbol at this position is called the **root symbol** of t. For $p \in \mathtt{Pos}(t)$, the **subterm** of t at position p, denoted $t|_p$, is defined inductively as:

$$\begin{array}{rcl} t|_\epsilon & \triangleq & t \\ f(t_1, \ldots, t_n)|_{i\,p} & \triangleq & t_i|_p, \quad \text{if } 1 \leq i \leq n \end{array}$$

Note that, for $p = i\,p'$, $p \in \mathtt{Pos}(t)$ implies that t is of the form $f(t_1, \ldots, t_n)$ with $1 \leq i \leq n$.

Definition 2.2.5 (Substitution) Let Σ be a signature and $\mathcal{X}$ be a countably infinite set of variables. A $T(\Sigma, \mathcal{X})$**-substitution** – or simply **substitution**, if the set of terms is clear from the context – is a function $\sigma : \mathcal{X} \to T(\Sigma, \mathcal{X})$ such that $\sigma(x) \neq x$ for only finitely many $x \in \mathcal{X}$. The (finite) set of variables that $\sigma(x)$ does not map to themselves is called the **domain** of σ: $\mathsf{dom}(\sigma) \triangleq \{x \in \mathcal{X} \mid \sigma(x) \neq x\}$. If $\mathsf{dom}(\sigma) = \{x_1, \ldots, x_n\}$, then we write σ as

$$\sigma = \{x_1 \leftarrow \sigma(x_1), \ldots, x_n \leftarrow \sigma(x_n)\}$$

The **restriction** of σ to S, denoted $\sigma|_S$, is defined as: $\sigma|_S \triangleq \{x \leftarrow M \mid x \leftarrow M \in \sigma \wedge x \in S\}$. A substitution σ is **idempotent** $\sigma(\sigma(x)) = \sigma(x)$ if for every $x \in \mathsf{dom}(\sigma)$.

Any $T(\Sigma, \mathcal{X})$-substitution σ can be **extended** to a mapping $\widehat{\sigma} : T(\Sigma, \mathcal{X}) \to T(\Sigma, \mathcal{X})$, thereby replacing all the occurrences of variables in its argument term by their respective σ-images. For example, let $t = f(0, x)$ and $t' = f(y, f(x, y))$, and let $\sigma = \{x \leftarrow g(y), y \leftarrow 0\}$. Then $\widehat{\sigma}(t) = f(0, g(y))$ and $\widehat{\sigma}(t') = f(0, f(g(y), 0))$. For the sake of simplicity, $\widehat{\sigma}$ is also denoted by σ.

Remark 2.2.6 To avoid clutter, we will sometimes use the notation t^σ to refer to $\sigma(t)$. When a substitution is not given a name, we denote its application to a term by writing the substitution next to term. E.g. $s\{x \leftarrow t\}$ means s^σ where $\sigma = \{x \leftarrow t\}$.

Lemma 2.2.7 (Substitution Lemma) Let s, t, u in $T(\Sigma, \mathcal{X})$ and x, y in $\mathcal{X}$. If $x \neq y$ and $x \notin \mathrm{Var}(u)$ then $s\{x \leftarrow t\}\{y \leftarrow u\} = s\{y \leftarrow u\}\{x \leftarrow t\{y \leftarrow u\}\}$

Definition 2.2.8 (Matching) A term s is said to **match** another term t if there is a substitution σ such that $s = t^\sigma$, meaning that s and t^σ are syntactically identical. When a term s matches a term t, we call s an **instance** of t.

2.2.2 Term Rewriting Systems

As mentioned above, a TRS is an abstract reduction system where the objects are first-order terms, and where the reduction relation is presented by means of reduction rules. Reduction rules are presented schematically in the sense that 1) arbitrary substitutions of a term for a variable are allowed; and 2) a reduction step thus obtained can be performed at arbitrary positions within a more complex term.

Example 2.2.9 Consider the following reduction rule:

$$x + 0 \to x$$

This rule allows us to perform, among others, the following reduction steps:

$$\begin{array}{lcl} x+0 & \to & x \\ 1+0 & \to & 1 \\ (2+0)+y & \to & 2+y \end{array}$$

We say that rewriting is **closed under substitution** and can be performed in any *context*. A **context** is a term containing zero, one or more occurrences of a special constant symbol $\Box$, denoting holes, i.e., a term over the extended signature $\Sigma \cup \{\Box\}$. If C is a context containing exactly n holes, and $t_1, \ldots, t_n$ are terms, then $C[t_1, \ldots, t_n]$ denotes the result of replacing the holes of C from left to right by $t_1, \ldots, t_n$. An important special case is when there is exactly one occurrence of $\Box$ in C. Then C is called a **one-hole context**, also denoted by $C[]$; the notation $C[]$ is used exclusively for one-hole contexts. If the term t can be written as $t = C[s]$, then the term s is said to be a **subterm** of t, notation $s \subseteq t$. Since $\Box$ is itself a context, the trivial context, we also have $t \subseteq t$. Other subterms s of t than t itself are called **proper subterms** of t, notation $s \subseteq t$.

Note that the same term s may occur more than once as a subterm in a term t. That is, there may be more than one **occurrence** of s in t.

Definition 2.2.10 (Reduction rule) A reduction rule for a signature Σ is a pair $\langle l, r\rangle$ of terms of $\mathrm{Ter}(\Sigma)$ such that l is not a variable and $\mathrm{Var}(r) \subseteq \mathrm{Var}(l)$. It will be written as $l \to r$. Often a reduction rule will get a name, e.g. ρ, and we write $\rho : l \to r$. An **instance** of ρ is obtained by applying a substitution σ. The result is a reduction step $l^\sigma \to_\rho r^\sigma$. The left-hand side l^σ is called a **redex**, more precisely a ρ-redex. The right-hand side r^σ is called its **reduct**.

Example 2.2.11 Consider a reduction rule $\rho : f(g(x), y) \to f(x, x)$. Then a substitution σ, with $\sigma(x) = 0$ and $\sigma(y) = g(x)$, yields the reduction step $f(g(0), g(x)) \to_\rho f(0, 0)$ with redex $f(g(0), g(x))$ and reduct $f(0, 0)$.

A redex r is somehow tied to the reduction rule according to which it is a redex. It may happen, however, that r can be considered both as a ρ_1-redex and as a ρ_2-redex, with respect to different rules ρ_1 and ρ_2.

Example 2.2.12 Consider the following rules:

$$\begin{array}{lcl} \rho_1 : f(a,x) & \to & x \\ \rho_2 : f(y,b) & \to & a \end{array}$$

With these rules, the term $f(a,b)$ is both a ρ_1-redex and a ρ_2-redex, and it may reduce to either b (via ρ_1) or a (via ρ_2).

Definition 2.2.13 A **term rewriting system** is a pair $\mathcal{R} = (\Sigma, R)$ of a signature Σ and a set of reduction rules R for Σ.

The one-step reduction relation of $\mathcal{R}$, denoted by $\to$ (or by $\to_{\mathcal{R}}$, when we want to be more specific), is defined as the union $\bigcup\{\to_\rho \mid \rho \in R\}$. So we have $t \to_{\mathcal{R}} s$ when $t \to_\rho s$ for some reduction rule $\rho \in R$.

In a natural way a term rewriting system $\mathcal{R} = (\Sigma, R)$ gives rise to a corresponding abstract reduction system, namely $(\mathrm{Ter}(\Sigma), \to_{\mathcal{R}})$. As a matter of fact, we will identify the two, with the effect that $\mathcal{R}$ is considered as an ARS by itself. All ARS definitions and results from Section 2.1 carry over to the TRS world.

Definition 2.2.14 A reduction rule is called **left-linear** (resp. **right-linear**) if its left-hand side (resp. right-hand side) is a linear term (i.e. no variable occurs in it more than once). A TRS is called **left-linear** if all of its reduction rules are.

2.2.3 Overlap and Critical Pairs

One term can contain several occurrences of redexes, thereby offering a choice of which redex to contract first. This may lead to non-confluence, as we have seen in example 2.2.12. In this section we will address the question of when the interaction of redexes is harmless, and when it is harmful.

In the sequel, we will always assume that different rules do not share variables. If that were not the case, variables can be renamed in order to ensure this. We say that two reduction rules $l_1 \to r_1$ and $l_2 \to r_2$ **overlap** if there is a non-variable subterm of l_1 that can be unified with l_2. Then we have $l_1 = C[s]|_p$ and $s^\sigma = l_2^\sigma$ for some non-variable term s and substitution σ. The trivial overlap of a reduction rule with itself at the root position can be ignored.

Note that overlaps of a rule with itself are possible, but require different redex positions in the term. It is also possible for a redex to contain another without overlap. This happens when there are two rules $\rho_1 : l_1 \to r_1$ and $\rho_2 : l_2 \to r_2$, and a term $t = l_1^\sigma$ where σ substitutes a variable in l_1 by a ρ_2-redex. This situation is called **nesting**, and it is harmless in that it does not constitute an obstacle for confluence.

Example 2.2.15 Consider the TRS with the following two reduction rules:

$$\begin{array}{lcl} \rho_1 : f(g(x),y) & \to & x \\ \rho_2 : g(a) & \to & b \end{array}$$

This TRS is not confluent. The term $f(g(a), x)$ can be reduced in two ways. We have $f(g(a), x) \to_{\rho_1} a$ and also $f(g(a), x) \to_{\rho_2} f(b, x)$, where the terms a and $f(b, x)$ are clearly normal forms, and hence without a common reduct. This is a case of overlap between ρ_1 and ρ_2. On the other hand, the term $f(g(b), g(a))$ has only 1 normal form: $g(b)$, which can be reached in two ways, by reducing either of the nested redexes first and the other one later.

Definition 2.2.16 (Critical Pair) Consider a pair of overlapping reduction rules $\rho_1 : l_1 \to r_1$ and $\rho_2 : l_2 \to r_2$. By definition of overlap (assuming ρ_1, ρ_2 are given in the appropriate order) there are a context D, a non-variable term s, and substitutions σ, δ such that $l_1 = D[s]$ and $s^\sigma = l_1^\delta$. The pair of one-step reducts of the outer redex $l_1^\sigma = D^\sigma[l_2^\delta]$ that arises from this overlap, $\langle D^\sigma[r_2^\delta], r_1^\sigma \rangle$, is called a **critical pair**.

Note that in a case of root overlap, this definition gives rise to two critical pairs. For example, in the TRS with reduction rules $\{a \to a, a \to b\}$, both $\langle a, b \rangle$ and $\langle b, a \rangle$ are critical pairs.

Definition 2.2.17 A critical pair $\langle s, t \rangle$ is called **joinable** if s and t have a common reduct, which can be reached in 0 or more reduction steps.

Clearly, if a non-joinable critical pair exists, then the TRS is not confluent. On the other hand, convergence of all critical pairs is not always sufficient for confluence, but it will guarantee local confluence. The latter is the content of the Critical Pair Lemma of Knuth and Bendix [KB70] and Huet [Hue80].

Lemma 2.2.18 (Critical Pair Lemma) A TRS is locally confluent if and only if all its critical pairs are joinable.

Definition 2.2.19 A TRS is **orthogonal** if it is left-linear and has no critical pairs. It is well-known [BKdV03] that every orthogonal TRS is confluent.

We conclude the section with a word on conditional TRSs. A **conditional term rewriting system** (CTRS) $\mathcal{C}$ consists of a first-order signature Σ together with a set of conditional reduction rules over Σ. The terms of $\mathcal{C}$ are just the usual first-order terms over Σ. The format of the conditional rules may vary, giving rise to different forms of CTRSs [DO90, Han97]. A **conditional reduction rule** has the form

$$\rho : t \to s \Leftarrow C_1, \ldots, C_n$$

Here $C_1, \ldots, C_n$ are the conditions, $n \geq 0$. A condition C_i takes the form $P_i(x_{i1}, \ldots, x_{ik})$, where P_i is any fixed predicate on terms and the $x_{i1}, \ldots, x_{ik}$ are variables in t. We require that the rule that remains when ρ is stripped of its conditions is a reduction rule in the usual, unconditional, sense. Note that if $n = 0$, then ρ has no conditions and we obtain a standard unconditional reduction rule.

Example 2.2.20 The following is a CTRS which uses $\to$ and $\twoheadrightarrow$ as predicates for its conditions.

$$
\begin{array}{lcll}
P(S(x)) & \to & x & \\
S(P(x)) & \to & x & \\
\mathit{positive}(0) & \to & \mathit{false} & \\
\mathit{positive}(S(0)) & \to & \mathit{true} & \\
\mathit{positive}(S(x)) & \to & \mathit{true} & \Leftarrow \mathit{positive}(x) \to \mathit{true} \\
\mathit{positive}(P(x)) & \to & \mathit{false} & \Leftarrow \mathit{positive}(x) \to \mathit{false} \\
x - 0 & \to & x & \\
x - S(y) & \to & P(x - y) & \\
x - P(y) & \to & S(x - y) & \\
\mathit{abs}(x) & \to & x & \Leftarrow \mathit{positive}(x) \twoheadrightarrow \mathit{true} \\
\mathit{abs}(x) & \to & 0 - x & \Leftarrow \mathit{positive}(x) \twoheadrightarrow \mathit{false}
\end{array}
$$

A CTRS with a pattern-matching predicate will be introduced in Section 3.6.2.

2.3 The λ-Calculus

We recall from the introduction the syntax of the pure λ-Calculus:

$$M, N ::= x \mid MN \mid \lambda x.M$$

where x ranges over a countably infinite set of variables $\mathcal{X}$. In $\lambda x.M$, $\lambda x.$ is a **binder**, and free occurrences of x in M become bound. The set of terms of the λ-calculus is denoted $\mathrm{Ter}(\lambda)$. A term in which all variables are bound is said to be **closed** or **ground**. Otherwise it is **open**. A closed term of pure λ-calculus is called a **combinator**.

Definition 2.3.1 The set of free variables in a term M (denoted as $\mathsf{FV}(M)$) is defined as follows:

$$
\begin{array}{lcl}
\mathsf{FV}(X) & \triangleq & \{x\} \\
\mathsf{FV}(MN) & \triangleq & \mathsf{FV}(M) \cup \mathsf{FV}(N) \\
\mathsf{FV}(\lambda x.M) & \triangleq & \mathsf{FV}(M) \backslash \{x\}
\end{array}
$$

Terms which are the same except for renaming of bound variables are not distinguished (such terms are called **α-equivalent**). Thus $\lambda x.x$ and $\lambda y.y$ are the same **identity function**.

In writing terms we freely use parentheses to remove ambiguity. We further adopt the conventions that application is left-associative and that the scope of a binder extends as far to the right as possible. For example fgh means $(fg)h$ and $\lambda x.\lambda y.Ma$ means $\lambda x.(\lambda y.(Ma))$.

Definition 2.3.2 (Substitution) The result of substituting N for the free occurrences of x in M, notation $M\{x \leftarrow N\}$, is defined inductively by:

$$
\begin{array}{lcll}
x\{x \leftarrow N\} & \triangleq & N & \\
y\{x \leftarrow N\} & \triangleq & y, & \text{if } y \neq x \\
(M_1 M_2)\{x \leftarrow N\} & \triangleq & (M_1\{x \leftarrow N\})(M_2\{x \leftarrow N\}) & \\
(\lambda y.M)\{x \leftarrow N\} & \triangleq & \lambda y.(M\{x \leftarrow N\}) &
\end{array}
$$

In the last clause of Definition 2.3.2 we assume that $x \neq y$ and y not free in N. If this were not the case, variable y can be renamed.

λ-Contexts are λ-terms containing some 'empty places', that is, occurrences of the constant $\Box$, also called **holes**. An analogous notion of context was introduced in Section 2.2 for first-order TRSs. A λ-context is generally denoted by C. If C is a λ-context containing n holes, and $M_1, \ldots, M_n \in \mathrm{Ter}(\lambda)$, then $C[M_1, \ldots, M_n]$ denotes the result of replacing the holes in C from left to right by $M_1, \ldots, M_n$. In this act, variables occurring free in $M_1, \ldots, M_n$ may become bound in $C[M_1, \ldots, M_n]$. In general we will only need contexts containing precisely one hole.

Example 2.3.3 $C = \lambda x.x(\lambda y.\Box)$ is a λ-context. If $M = xy$, then $C[M] = \lambda x.x(\lambda y.xy)$.

2.3.1 Reduction

The λ-calculus in its simplest form has only one rule, which describes how to substitute an argument into the body of a function:

$$\beta \ : \ (\lambda x.M)N \to M\{x \leftarrow N\}$$

Here $M\{x \leftarrow N\}$ means "substitute N for free occurrences of x in M". The smallest reflexive, symmetric, transitive, substitutive relation on terms including $\to_\beta$, written $\equiv_\beta$, is Church's notion of λ-conversion. As for TRSs, an instance of the left hand side of rule β is called a **redex**. All the notions defined for ARSs and TRSs carry over to the λ-Calculus, which is in fact a higher order rewrite system (see [Nip95]).

There are non-normalizing terms, of which perhaps the simplest is $(\lambda x.xx)(\lambda x.xx)$. We have the cyclic reduction $(\lambda x.xx)(\lambda x.xx) \to_\beta (\lambda x.xx)(\lambda x.xx)$ as the only available step. A consequence of the following result is confluence of the λ-Calculus.

Theorem 2.3.4 (Church-Rosser) If $M_1 \equiv_\beta M_2$, then there is a term N such that $M_1 \twoheadrightarrow_\beta N$ and $M_2 \twoheadrightarrow_\beta N$.

An immediate consequence of this is that the normal form of a normalizing term is unique.

Intensional and extensional equality. β-reduction uses a concept of function equality that is called **intensional**. That is, it does not include the assumption used in most of mathematics that functions with identical graphs are necessarily equal. This difference is overcome by a second relation, η-reduction, which identifies terms having the same applicative behaviour (**extensional** equality). η-reduction is defined by the following rule:

$$\eta \ : \ (\lambda x.Mx) \to M, \qquad \text{if } x \notin \mathsf{FV}(M)$$

Note that the introduction of the η rule affects the equational theory: $\equiv_{\beta\eta}$ is not the same as $\equiv_\beta$. For example, $\lambda x.yx \equiv_{\beta\eta} y$, but $\lambda x.yx \not\equiv_\beta y$.

$$\frac{}{\Gamma, x^A \vdash x : A} \qquad \frac{\Gamma, x^A \vdash M : B}{\Gamma \vdash \lambda x^A.M : A \to B} \qquad \frac{\Gamma \vdash M : A \to B \quad \Gamma \vdash N : A}{\Gamma \vdash MN : B}$$

Figure 2.3: Typing rules of $\lambda^{\to}$

2.3.2 Simply Typed λ-Calculus

The simply typed λ-calculus has been formulated in two ways. The first, originated by Curry [Cur34]; see also [CF58] and [CSH72], is called **implicit typing**: the terms are the same as in the untyped calculus and each term has a set of possible types assigned to it. The second approach, originated by Church [Chu40], is called **explicit typing**: terms are annotated with type information which uniquely determines a type for the term. In the following, we will follow Church's approach.

We assume given a non-empty set of type variables $\{\alpha_1, \ldots\}$. The set of **types**, $\textsc{Typ}$, is defined as:

$$A, B \quad ::= \quad \alpha \,|\, A \to B$$

Definition 2.3.5

- The set of **pseudo-terms** (or terms which are not necessarily typable) is defined by the following grammar:
$$M, N ::= x \mid MN \mid \lambda x^A.M$$
where x ranges over a countably infinite set of variables $\mathcal{X}$ and $A \in \textsc{Typ}$.
- A **typing context** is a finite set of type-decorated variables $\{x_1^{A_1}, \ldots, x_n^{A_n}\}$, with $x_i \neq x_j$ for $i \neq j$.
- A **typing judgement** is an expression of the form $\Gamma \vdash M : A$, with Γ a typing context, M a term and $A \in \textsc{Typ}$.
- The **typability relation** $\Vdash$, also known as **derivability** of typing judgements, is defined by the rules shown in Figure 2.3.5.
- We say that a typing judgement $\Gamma \vdash M : A$ is **derivable** if $\Gamma \vdash M : A \in \Vdash$.
- If $\Gamma \vdash M : A$ is derivable, then we say that M has type A in Γ. We say that M is **typable** if there are Γ and A such that $\Gamma \vdash M : A \in \Vdash$.
- The set of **terms** $\mathrm{Ter}(\lambda^{\to})$ is defined as the set of all typable pseudo-terms.
- The simply typed λ-calculus à la Church ($\lambda^{\to}$ for short) is the triple $(\mathrm{Ter}(\lambda^{\to}), \mathsf{Typ}, \Vdash)$.

Example 2.3.6 Let A, B, C be arbitrary types. Then:

- $\vdash \lambda x^A.x : A \to A$

$$\begin{array}{ll} Sxyz & \to xz(yz) \\ Kxy & \to x \\ Ix & \to x \end{array}$$

Table 2.1: Combinatory logic, applicative notation

- $\vdash \lambda x^A.\lambda y^B.x \colon A \to B \to A$
- $\vdash \lambda x^{A\to B\to C}.\lambda y^{A\to B}.\lambda z^A.xz(yz) \colon (A \to B \to C) \to (A \to B) \to A \to C$

Free and bound variables, closed terms, subterms, substitution and the set of contexts are defined in the obvious way by analogy to the type-free calculus. But the types of terms have to fit: that is, $M\{x^A \leftarrow N\}$ is defined only if $\Gamma \vdash N \colon A$ for some Γ; likewise $C[M]$ is defined only if $C[M] \in \mathrm{Ter}(\lambda^{\to})$.

The notions of reduction in the typed λ-calculus are the obvious analogues of the notions that we introduced in the type-free case.

Strong normalization of simply typed λ-calculus was proved independently within the scope of a stronger system by Dragalin [Dra68], Gandy [Gan80], Hinata [Hin67], Hanatani [HM66] and Tait [Tai67].

2.4 Combinatory Logic

Recall from the introduction that *CL*-expressions are constructed from some number of predefined constants (**combinators**) and a binary *CL*-expression constructor *Ap* (application):

$$M, N ::= x \mid K \mid S \mid I \mid Ap(M, N)$$

The constants I, K, S are called the **combinators**. Application associates to the left. The reduction rules are recalled in Table 2.1, where, as mentioned, $Ap(M, N)$ is abbreviated $M\,N$. The combinator I acts as a "universal" identity operator. Universal here means without domain restrictions. Application is defined between any pair of terms. Thus I can even be applied to itself, and we get $II \to I$. The combinator K may be thought of as a builder of constant functions. Applying K to an argument M yields the constant function $K\ M$, which gives the same value M on any argument N. The combinator S has a slightly more complicated behaviour; it is related to substitution, and it is relevant for achieving combinatorial completeness (Theorem 2.4.6).

CL is known to be confluent. This follows from the fact that *CL* is an orthogonal TRS (see Definition 2.2.19). However, *CL* is not strongly normalizing, as illustrated by the following reduction cycle:

Example 2.4.1 $SII(SII) \to I(SII)(I(SII)) \to SII(I(SII)) \to SII(SII)$

Additional Comments on Combinators In the applicative notation the combinators S, K, I may be followed by an arbitrary number of arguments $M_1, \ldots, M_n$ ($n \geq 0$). Therefore one may look at these constants as operators with variable arity. But note that at least three

arguments are needed before one can use the reduction rule for S, and at least two for K; for example, $S\ M_1\ M_2\ M_3\ M_4\ M_5\ M_6 \to M_1\ M_3\ (M_2\ M_3)M_4\ M_5\ M_6$.

Apart from I, K and S, many more operators can be defined in CL. For example, composition of functions is represented by the term $B \triangleq S(KS)K$. We have $Bxyz = S(KS)Kxyz \to KSx(Kx)yz \to S(Kx)yz \to Kxz(yz) \to x(yz)$.

Another example, the term $D \triangleq SII$, was already encountered in Example 2.4.1. Given an arbitrary argument, D copies it and applies it to itself. We have $Dx = SIIx \to Ix(Ix) \to x(Ix) \to xx$

Likewise, defining C as $S(BBS)(KK)$, we have $Cxyz \to xzy$. If f is seen as a function which takes two arguments, then Cf is the function which behaves like f if the two arguments are given in reverse order.

Terms such as B, D, C, and in fact all closed CL-terms, are called combinators. The above examples of combinators are well-known examples, which were already introduced by Schönfinkel, and can be found in the books by Curry and Feys [CF58], Barendregt [Bar84] or Hindley and Seldin [HS86]. Note that each of the above combinators is a solution for F in an explicit definition of the form $Fx_1 \dots x_n \to M$, with M a CL-term containing only variables from the set $\{x_1, \dots, x_n\}$. Not only these, but all such explicitly definable functions, can be represented within CL as combinators. This phenomenon is called **combinatorial completeness**, and forms the essence of CL. One of Schönfinkel's additional contributions to CL is that combinatorial completeness could already be attained by the clever choice of a small set of combinators. The combinators I, K, S were tailor-made for the purpose of combinatorial completeness (Theorem 2.4.6) gives the real motivation for their choice. As a matter of fact, the combinator I may be dropped in order to achieve an even simpler formulation of CL, since I can be defined in terms of S and K as SKK.

Relationship between *CL* and the λ-Calculus. There is a close correspondence between the λ-Calculus and Combinatory Logic, which we now make precise. In what follows, we write $\mathrm{Ter}(CL)$ for the set of CL terms, $\to_{CL}$ for the one-step reduction relation generated by the reduction rules given in Table 2.1 and $\twoheadrightarrow_{CL}$ for its reflexive-transitive closure.

Definition 2.4.2 (From combinators to λ-terms) Let $M \in \mathrm{Ter}(CL)$. Its corresponding λ-term, denoted t_λ, is defined as follows:

$$\begin{array}{lcl} x_\lambda & \triangleq & x \\ K_\lambda & \triangleq & \lambda x.\lambda y.x \\ S_\lambda & \triangleq & \lambda x.\lambda y.\lambda z.xz(yz) \\ I_\lambda & \triangleq & \lambda x.x \\ (tt')_\lambda & \triangleq & t_\lambda t'_\lambda \end{array}$$

Proposition 2.4.3 For $M, N \in \mathrm{Ter}(CL)$, if $M \twoheadrightarrow_{CL} N$ then $M_\lambda \twoheadrightarrow_\beta N_\lambda$.

The converse implication, however, is false. In particular, KI is a normal form in CL, while $(KI)_\lambda = (\lambda x.\lambda y.x)(\lambda z.z)$ contains a β-redex, and reduces in one step to $\lambda y.(\lambda z.z)$. For this reason, the reduction of CL is normally called **weak reduction**.

Definition 2.4.4 (From λ-terms to combinators) The mapping from λ-terms to combinators, as shown on Section 1.1.1, is defined as follows:

$$\begin{array}{rcl} x_{CL} & \triangleq & x \\ (MN)_{CL} & \triangleq & M_{CL}N_{CL} \\ (\lambda x.M)_{CL} & \triangleq & [x].M_{CL} \end{array}$$

with $[x].M$ defined recursively as follows:

$$\begin{array}{llll} 1) & [x].x & \triangleq & I \\ 2) & [x].M & \triangleq & KM, \qquad \text{if } M \text{ is a constant or a variable other than } x \\ 3) & [x].(MN) & \triangleq & S([x].M)([x].N) \end{array}$$

Note that the variable x does not occur in the CL-term denoted by $[x].M$.

Proposition 2.4.5 [See [BKdV03]] Let $M, N \in \mathrm{Ter}(CL)$.

(i) $([x].M)x \twoheadrightarrow M$.

(ii) $([x].M)N \twoheadrightarrow M\{x \leftarrow N\}$.

We can finally state the combinatorial completeness of CL.

Theorem 2.4.6 (Combinatorial completeness) Given a CL-term M with all its variables in $\{x_1 \ldots x_n\}$, there is a CL-term F such that $Fx_1 \ldots x_n \twoheadrightarrow M$.

2.5 The Curry-Howard Isomorphism

We introduce Natural Deduction for propositional Intuitionistic Logic (Int), then show the correspondence between its implicational fragment and simply typed λ-Calculus. Finally, we will introduce the $\lambda\mu$-calculus, an extension of the λ-calculus which has a correspondence with Classical Propositional Logic. For further details see [Gen35, Pra65, SU06].

2.5.1 Natural Deduction for Intuitionistic Logic

Intuitionistic logic is a system of symbolic logic that differs from classical logic by replacing the traditional concept of *truth* with the concept of *constructive provability*. Developed by Arend Heyting [Hey30], Int encompasses the principles of logical reasoning which were used by L. E. J. Brouwer in developing his intuitionistic mathematics, beginning in [Bro07].

In order to understand intuitionism, one should forget the classical, Platonic notion of *truth*. Now our judgements about statements are no longer based on any predefined value of that statement, but on the existence of a proof or **construction** of that statement. Before informally explaining the constructive semantics of propositional connectives we first fix our language. We assume an infinite set of propositional variables $P, Q, \ldots$, and define the set of formulas by induction:

$$A, B ::= \bot \mid P \mid A \supset B \mid A \wedge B \mid A \vee B$$

Our basic connectives are: implication $\supset$, disjunction $\vee$, conjunction $\wedge$, and the constant $\bot$ (false). For all formulas A, B, $\neg A$ and $A \Leftrightarrow B$ are abbreviations for $A \supset \bot$ and $(A \supset B) \wedge (B \supset A)$ respectively. Negation $\neg A$ is defined as shorthand for $A \supset \bot$. Implication is assumed right associative. Negation has the highest precedence, and implication the lowest priority, with no preference between $\wedge$ and $\vee$. That is, $\neg A_1 \vee A_2 \supset A_3$ means $((\neg A_1) \vee A_2) \supset A_3$. We forget about outermost parentheses.

Returning to the informal explanation of the constructive semantics of propositional connectives, we recall the BHK-interpretation (for Brouwer, Heyting and Kolmogorov). The algorithmic flavor of this definition will later lead us to the Curry-Howard Isomorphism (for further details see [TS00] and [TVD88]).

- A construction of $A \wedge B$ consists of a construction of A and a construction of B.
- A construction of $A_1 \vee A_2$ consists of a number $i \in \{1, 2\}$ and a construction of A_i.
- A construction of $A \supset B$ is a method (function) transforming every construction of A into a construction of B.
- There is no possible construction of $\bot$.

Intuitionistic negation $\neg A$ is stronger than just "there is no construction for A". Both $P \supset \neg\neg P$ and $\neg\neg P \supset P$ are classical tautologies, however only the former holds in Int.

We now present Natural Deduction for Int.

Definition 2.5.1

- A **context** Γ is a finite subset of A. The formulas in Γ represent hypotheses. These hypotheses may be labeled (e.g. x^A instead of just A) in order to allow the presence of more than one instance of the same formula.
- The relation $\Gamma \vdash A$ is defined by the rules in Figure 2.4.
- We write Γ, A (or Γ, x^A) instead of $\Gamma \cup \{A\}$ (or $\Gamma \cup \{x^A\}$). We also write $\vdash A$ instead of $\emptyset \vdash A$.
- A formal **proof** of $\Gamma \vdash A$ is a finite tree, whose nodes are labelled by pairs $\langle \Gamma', A' \rangle$, which will also be written $\Gamma' \vdash A'$, satisfying the following conditions:
  - The root label is $\Gamma \vdash A$.
  - All the leaves are labelled by axioms.
  - The label of each father node is obtained from the labels of the children using one of the rules.

$$\frac{}{\Gamma, A \vdash A}\,\mathsf{Ax} \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \supset B}\,\supset\mathsf{I} \qquad \frac{\Gamma \vdash A \supset B \quad \Gamma \vdash A}{\Gamma \vdash B}\,\supset\mathsf{E}$$

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B}\,\vee_{\mathsf{I1}} \qquad \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B}\,\vee_{\mathsf{I2}} \qquad \frac{\Gamma \vdash A \vee B \quad \Gamma, A \vdash B \quad \Gamma, B \vdash B}{\Gamma \vdash B}\,\vee_{\mathsf{E}}$$

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B}\,\wedge_{\mathsf{I}} \qquad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A}\,\wedge_{\mathsf{E1}} \qquad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash B}\,\wedge_{\mathsf{E2}} \qquad \frac{\Gamma \vdash \bot}{\Gamma \vdash A}\,\bot E$$

Figure 2.4: Intuitionistic propositional calculus

- If there is a proof of $\vdash A$, then we say that A is a **theorem** of the intuitionistic propositional calculus. Sample proofs of various theorems are given in Fig. 2.5.

There is a natural interest in proofs without "roundabouts". One arrives at such proofs from arbitrary proofs by means of **proof normalization** rules that eliminate detours in a proof. More concretely, consider the proof tree in Fig. 2.5(a). It proves that $A \supset A$ is derivable. However, it does so by first showing $A \supset A$, then inferring $(A \supset A) \wedge (B \supset B)$, and then, finally, concluding $A \supset A$. A more direct proof tree, which does not make an excursion via $(A \supset A) \wedge (B \supset B)$, is in Fig. 2.5(b). Note that the detour in the former proof tree is signified by an introduction rule immediately followed by the corresponding elimination rule, for example $\wedge_{\mathsf{I}}$ and $\wedge_{\mathsf{E1}}$. In fact, the above style of detour-elimination is possible whenever an introduction rule is immediately followed by the corresponding elimination rule. A similar situation occurs in the proof tree of Fig. 2.5(c), where the proof may be simplified as exhibited in Fig. 2.5(d).

These normalization rules take the following general form (symmetric cases are omitted: they can be obtained by replacing $\wedge_{\mathsf{E1}}$ and $\vee_{\mathsf{I1}}$ by $\wedge_{\mathsf{E2}}$ and $\vee_{\mathsf{I2}}$ respectively):

$$\frac{\dfrac{\dfrac{\pi_1}{\Gamma \vdash A} \quad \dfrac{\pi_2}{\Gamma \vdash B}}{\Gamma \vdash A \wedge B}\,\wedge_{\mathsf{I}}}{\Gamma \vdash A}\,\wedge_{\mathsf{E1}} \qquad \rightarrow \qquad \frac{\pi_1}{\Gamma \vdash A}$$

$$\frac{\dfrac{\dfrac{\dfrac{\Gamma, x^B \vdash B}{\pi_1}}{\Gamma, x^B \vdash A}}{\Gamma \vdash B \supset A}\,\supset\mathsf{I} \quad \dfrac{\pi_2}{\Gamma \vdash B}}{\Gamma \vdash A}\,\supset\mathsf{E} \qquad \rightarrow \qquad \frac{\dfrac{\pi_2}{\Gamma \vdash B}}{\dfrac{\pi_1}{\Gamma \vdash A}}$$

$$\frac{\dfrac{\dfrac{\pi_1}{\Gamma \vdash A}}{A \vee B}\,\vee_{\mathsf{I1}} \quad \dfrac{\dfrac{\Gamma, x^A \vdash A}{\pi_2}}{\Gamma, x^A \vdash C} \quad \dfrac{\dfrac{\Gamma, y^B \vdash B}{\pi_3}}{\Gamma, y^B \vdash C}}{\Gamma \vdash C}\,\vee_{\mathsf{E}} \qquad \rightarrow \qquad \frac{\dfrac{\pi_1}{\Gamma \vdash A}}{\dfrac{\pi_2}{\Gamma \vdash C}}$$

$$\dfrac{\dfrac{\dfrac{}{A \vdash A}\,\mathsf{Ax}}{\vdash A \supset A}\,\supset\mathsf{I} \qquad \dfrac{\dfrac{}{B \vdash B}\,\mathsf{Ax}}{\vdash B \supset B}\,\supset\mathsf{I}}{\dfrac{\vdash (A \supset A) \wedge (B \supset B)}{\vdash A \supset A}\,\wedge_{\mathsf{E1}}}\,\wedge_{\mathsf{I}}$$

(a)

$$\dfrac{\dfrac{}{A \vdash A}\,\mathsf{Ax}}{\vdash A \supset A}\,\supset\mathsf{I}$$

(b)

$$\dfrac{\dfrac{\dfrac{\dfrac{}{A \supset A, B \vdash A \supset A}\,\mathsf{Ax}}{A \supset A \vdash B \supset A \supset A}\,\supset\mathsf{I}}{\vdash (A \supset A) \supset B \supset A \supset A}\,\supset\mathsf{I} \qquad \dfrac{\dfrac{}{A \vdash A}\,\mathsf{Ax}}{\vdash A \supset A}\,\supset\mathsf{I}}{\vdash B \supset A \supset A}\,\supset\mathsf{E}$$

(c)

$$\dfrac{\dfrac{\dfrac{}{B, A \vdash A}\,\mathsf{Ax}}{B \vdash A \supset A}\,\supset\mathsf{I}}{\vdash B \supset A \supset A}\,\supset\mathsf{I}$$

(d)

Figure 2.5: Normalizing proofs

The first rule states that if we, somewhere in a proof, infer A and B, and then use $\wedge$-introduction to infer $A \wedge B$ followed by $\wedge$-elimination to infer A, we might as well avoid the detour and replace this proof simply by the subproof of A. The second rule says that if we have obtained a proof of A from assumption B, which we use alongside $\supset$-introduction to get a proof of $B \supset A$, and we also have a proof of B; then, instead of inferring A by $\supset$-elimination, we can use the original proof of A where we plug in the proof of B in all the places where the assumption B occurs. In other words, the resulting proof is obtained by replacing in π_1 the occurrences of the hypothesis x^B by the proof π_2. If there are several occurrences of $A \vdash A$ in π_1, the subproof π_2 is replicated as many times as necessary; if there is no occurrence of x^B in π_1, the subproof π_2 is erased. The reading of the third rule is similar.

The process of eliminating proof detours of the above kind, is called *proof normalization*, and a proof tree with no detours is said to be in *normal form*.

2.5.2 The Curry-Howard Isomorphism

As mentioned in the Introduction, there is a precise correspondence between proofs in intuitionistic propositional logic and typable $\lambda^{\rightarrow}$-terms. We review this for the implicational fragment of intuitionistic propositional logic ($\mathsf{IPC}^{\supset}$).

If we take the set of propositional variables equal to U (the set of type variables), and assume the type constructor $\rightarrow$ to be associated to $\supset$, then A (the set of propositional formulas in the implicational fragment of intuitionistic propositional logic) and $\textsc{Typ}$ (the set of simple types) are identical. Moreover, if we label all hypotheses used in a Natural Deduction proof, then the contexts used in the proof are also identical to those used for typing.

$\lambda^{\rightarrow}$	$\mathsf{IPC}^{\supset}$
term variable	assumption
term	construction (proof)
type variable	propositional variable
type	formula
type constructor	connective
inhabitation	provability
typable term	construction for a proposition
redex	construction representing proof tree with redundancy
reduction	normalization

Figure 2.6: The Curry-Howard Isomorphism

Proposition 2.5.2 (Curry-Howard Isomorphism)

(i) $\Gamma \vdash M : A$ derivable in $\lambda^{\rightarrow} \Rightarrow \Gamma \vdash A$ derivable in $\mathsf{IPC}^{\supset}$.

(ii) $\Gamma \vdash A$ derivable in $\mathsf{IPC}^{\supset} \Rightarrow \exists M \in \mathrm{Ter}(\lambda^{\rightarrow})$ s.t. $\Gamma \vdash M : A$ derivable in $\lambda^{\rightarrow}$.

Thus $\lambda^{\rightarrow}$ and $\mathsf{IPC}^{\supset}$ may be viewed as different names for essentially the same thing.

This correspondence also extends to proof normalisation. A *redex* is a proof tree containing an application of an introduction rule immediately followed by an application of the corresponding elimination rule. Reduction on terms corresponds to proof normalization. A term in normal form corresponds to a construction representing a proof tree in normal form. Type Preservation states that reducing a construction of a formula yields a construction for the same formula. The Church-Rosser Theorem (confluence of the λ-calculus) states that the order of normalization will not affect the form of the proof without detours. This, and other corresponding notions, are depicted in Fig. 2.6.

The perfect correspondence between reduction and normalization and the related concepts, justifies the name *isomorphism* rather than simply *bijection*. In fact, reduction has been studied extensively for the λ-calculus and its variants, while normalization has been studied independently in proof theory.

2.5.3 Classical Natural Deduction

Parigot [Par92] introduced Classical Natural Deduction (CND) in order to avoid the problems which arose from trying to use $\neg\neg A \rightarrow A$ as a type for control operators while restricting typing judgements to a single conclusion (i.e. one type). In addition to serving as Natural Deduction presentation for classical propositional logic, CND was the base for an algorithmic calculus which will be introduced in Section 2.5.4.

The formulas of CND are constructed as follows:

$$A, B ::= P(s_1, \ldots, s_n) \mid A \supset B \mid \forall x.A \mid \forall X.A$$

$$A \vdash A$$

$$\frac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \supset B, \Delta} \supset I \qquad \frac{\Gamma \vdash A \supset B, \Delta \quad \Gamma' \vdash A, \Delta'}{\Gamma \cup \Gamma' \vdash B, \Delta \cup \Delta'} \supset E$$

$$\frac{\Gamma, A \vdash \Delta}{\Gamma \vdash \neg A, \Delta} \neg I \qquad \frac{\Gamma \vdash \neg A, \Delta \quad \Gamma' \vdash A, \Delta'}{\Gamma \cup \Gamma' \vdash \Delta \cup \Delta'} \neg E$$

$$\frac{\Gamma \vdash A\{x \leftarrow y\}, \Delta}{\Gamma \vdash \forall x.A, \Delta} \forall \mathsf{I}_1 \qquad \frac{\Gamma \vdash \forall x.A, \Delta}{\Gamma \vdash A\{x \leftarrow T\}, \Delta} \forall \mathsf{E}_1$$

$$\frac{\Gamma \vdash A\{X \leftarrow Y\}, \Delta}{\Gamma \vdash \forall X.A, \Delta} \forall \mathsf{I}_2 \qquad \frac{\Gamma \vdash \forall X.A, \Delta}{\Gamma \vdash A\{X \leftarrow T\}, \Delta} \forall \mathsf{E}_2$$

Figure 2.7: Classical (second-order) Natural Deduction schemes

where s_i range over first-order terms, x ranges over individuals and X ranges over propositions. Letters Γ, Δ denote sets of formulas. Sequents $\Gamma \vdash \Delta$ are interpreted as usual: Γ is seen as a conjunction of premises, and Δ as a disjunction of conclusions. The inference schemes are given in Fig. 2.7 (where y (resp. Y) is not free in the conclusion of the first order (resp. second-order) introduction rule for "$\forall$"). The formulas explicitly mentioned in the rules are called **active**. The one which bears the connective is called the **main formula** of the rule. Weakening is managed implicitly: non-occurring active formulas are allowed in the premises of the rules.

The algorithmic interpretation of the system will follow the proof normalization procedure. As above, detours are understood as "obstacles" to the subformula property, in which the premises of each rule in the calculus consist of subformulas of the formulas occurring in the conclusion of the rule. We distinguish between logical detours and structural detours. One has a **logical detour** when the main formula of an elimination rule R_1 is active in the preceding rule R_2, and R_2 is an introduction rule; one has a **structural detour** when the main formula of an elimination rule R_1 is not active in the preceding rule R_2.

Definition 2.5.3 (Classical proof normalization) Logical and structural detours have corresponding normalization rules. We will show the logical normalization rules for the $\supset$ connective (the rules for $\neg$ and $\forall$ are analogous):

$$\frac{\dfrac{\dfrac{\dfrac{A \vdash A}{\pi_1}}{\Gamma_1, A \vdash B, \Delta_1}}{\Gamma_1 \vdash A \supset B, \Delta_1} \supset I \qquad \dfrac{\pi_2}{\Gamma_2 \vdash A, \Delta_2}}{\Gamma_1 \cup \Gamma_2 \vdash B, \Delta_1 \cup \Delta_2} \supset E \quad \rightarrow \quad \dfrac{\dfrac{\pi_2}{\Gamma_2 \vdash A, \Delta_2}}{\dfrac{\pi_1}{\Gamma_1 \cup \Gamma_2 \vdash B, \Delta_1 \cup \Delta_2}}$$

Analogously to the proof normalisation rule for Intuitionistic Logic, the resulting proof is

obtained by replacing in π_1 the occurrences of the axiom $A \vdash A$ by the proof π_2.

The rule for structural normalization is as follows:

$$\dfrac{\dfrac{\dfrac{\pi_1}{\Gamma_1 \vdash A \supset B, \Delta_1}}{\Gamma_2 \vdash A \supset B, \Delta_2} \quad \dfrac{\pi_3}{\Gamma_3 \vdash A, \Delta_3}}{\Gamma_2 \cup \Gamma_3 \vdash B, \Delta_2 \cup \Delta_3} \supset E \quad \rightarrow \quad \dfrac{\dfrac{\dfrac{\pi_1}{\Gamma_1 \vdash A \supset B, \Delta_1} \quad \dfrac{\pi_3}{\Gamma_3 \vdash A, \Delta_3}}{\Gamma_1 \cup \Gamma_3 \vdash B, \Delta_1 \cup \Delta_3} \supset E}{\pi_2}}{\Gamma_2 \cup \Gamma_3 \vdash B, \Delta_2 \cup \Delta_3}$$

The final proof is obtained by recursively replacing in π_2 each subproof π_1 – whose conclusion contains the formula $A \supset B$ on its right-hand side –, by the proof obtained by $\supset E$ from the conclusions of π_1 and π_3.

2.5.4 The $\lambda\mu$-Calculus

The $\lambda\mu$-calculus [Par92] is an extension of the lambda calculus that introduces two new operators: the "μ" operator and the "[ ]" operator. According to the Curry-Howard isomorphism, lambda calculus on its own can express theorems in intuitionistic logic only, but several classical logical theorems can't be written at all. However with these new operators one is able to write terms that have the type of, for example, the law of excluded middle ($A \vee \neg A$), or Peirce's law ($(((A \supset B) \supset A) \supset A)$. This calculus is introduced as the term assignment for a presentation of second-order CND.

Semantically, these operators correspond to continuations found in some functional programming languages (introduced by Adriaan van Wijngaarden for Algol 60 in [vW64]). A formulas-as-types application of the $\lambda\mu$-calculus to the Scheme programming language can be found in [Gri90]. A different application of the $\lambda\mu$-calculus in programming can be found in [Sau10], which makes use of the four reduction rules shown in [Par97] in conjunction with a new type system to model the computational behavior of streams.

The (typed) $\lambda\mu$-Calculus uses two different types of variables: λ-variables x, y, z and μ-variables α, β, γ. λ-variables are used to label formulas on the left-hand side of a sequent, while μ-variables label formulas on the right-hand side (except for at most one formula on the right-hand side, which may not be labelled). Different formulas cannot have the same label.

Definition 2.5.4 Named and unnamed $\lambda\mu$-terms are defined by mutual recursion as follows:

$$\begin{array}{rcll} T, U & ::= & x \mid \lambda x^A.U \mid T\ U \mid \mu\alpha^A.N & \text{(unnamed terms)} \\ N & ::= & [\alpha^A]U & \text{(named terms)} \end{array}$$

As in the λ-calculus, application is assumed to be left-associative and have the highest precedence; unnecessary parentheses may be dropped.

Definition 2.5.5 (Typing rules) The typing rules for the $\lambda\mu$ calculus are those shown in Fig. 2.8, where each Γ is a set of formulas labelled by λ-variables, and each Δ is a set of formulas labelled by μ-variables.

Logical rules:

$$x : A^x \vdash A$$

$$\frac{U : \Gamma, A^x \vdash B, \Delta}{\lambda x^A.U : \Gamma \vdash A \supset B, \Delta} \qquad \frac{T : \Gamma \vdash A \supset B, \Delta \quad U : \Gamma' \vdash A, \Delta'}{(T\ U) : \Gamma \cup \Gamma' \vdash B, \Delta \cup \Delta'}$$

$$\frac{U : \Gamma \vdash A\{y \leftarrow x\}, \Delta}{U : \Gamma \vdash \forall x.A, \Delta} \qquad \frac{U : \Gamma \vdash \forall x.A, \Delta}{U : \Gamma \vdash A\{x \leftarrow T\}, \Delta}$$

$$\frac{U : \Gamma \vdash A\{Y \leftarrow X\}, \Delta}{U : \Gamma \vdash \forall X.A, \Delta} \qquad \frac{U : \Gamma \vdash \forall X.A, \Delta}{U : \Gamma \vdash A\{X \leftarrow T\}, \Delta}$$

Naming rules:

$$\frac{T : \Gamma \vdash A, \Delta}{[\alpha^A]T : \Gamma \vdash A^\alpha, \Delta} \qquad \frac{N : \Gamma \vdash A^\alpha, \Delta}{\mu\alpha^A.N : \Gamma \vdash A, \Delta}$$

Figure 2.8: Axiom and inference schemes of $\lambda\mu$.

Note that the logical rules are essentially the ones of typed λ-calculus. We can also define "$\neg A$" as "$A \supset \bot$", assuming that the formula $\bot$ (which can be defined as $\forall X.X$) and hypotheses of the form $\bot^\alpha$ are implicit in the rules.

Remark 2.5.6 The type system for the $\lambda\mu$-calculus can be seen as a system with at most one conclusion, if each named formula in the conclusions is replaced by its negation in the hypotheses. Then the "μ" operator becomes in a certain sense an algorithmic version of the classical absurdity rule.

Example 2.5.7 Proof of $\neg\neg A \supset A$:

$$\cfrac{\cfrac{y : \neg\neg A^y \vdash \neg\neg A \quad \cfrac{\cfrac{\cfrac{x : A^x \vdash A}{[\alpha^A]x : A^x \vdash A^\alpha}}{\mu\alpha^A.[\alpha^A]x : A^x \vdash A}}{\lambda x^A.\mu\alpha^A.[\alpha^A]x : \vdash \neg A, A}}{(y\ \lambda x^A.\mu\alpha^A.[\alpha^A]x) : \neg\neg A^y \vdash A}}{\lambda y^{\neg\neg A}.(y\ \lambda x^A.\mu\alpha^A.[\alpha^A]x) : \vdash \neg\neg A \supset A}$$

Definition 2.5.8 Following the proofs as terms correspondence, the normalization rules can be translated into term reduction rules [Par97]:

$$
\begin{array}{lll}
R_1: & (\lambda x^A.T)U & \to \; T\{x \leftarrow U\} \\
R_2: & (\mu\alpha^{A\supset B}.N)U & \to \; \mu\beta^B.N(\!|\, [\alpha^{A\to B}](\bullet) \leftarrow [\beta^B](\bullet)U \,|\!) \\
S_1: & [\beta^A]\mu\alpha^A.N & \to \; N\{\alpha^A \leftarrow \beta^A\} \\
S_2: & \mu\alpha^A.[\alpha^A]U & \to \; U, \qquad \text{if } \alpha^A \notin \mathsf{FV}(U)
\end{array}
$$

where structural substitution $n(\!|\, [\alpha^{A\to B}](\bullet) \leftarrow [\beta^B](\bullet)U \,|\!)$ is defined as follows:

$$
\begin{array}{llll}
x & (\!|\, [\alpha^{A\to B}](\bullet) \leftarrow [\beta^B](\bullet)U \,|\!) & \triangleq & x \\
\lambda x^C.T & (\!|\, [\alpha^{A\to B}](\bullet) \leftarrow [\beta^B](\bullet)U \,|\!) & \triangleq & \lambda x^C.(T(\!|\, [\alpha^{A\to B}](\bullet) \leftarrow [\beta^B](\bullet)U \,|\!)) \\
(T\ T') & (\!|\, [\alpha^{A\to B}](\bullet) \leftarrow [\beta^B](\bullet)U \,|\!) & \triangleq & (T(\!|\, [\alpha^{A\to B}](\bullet) \leftarrow [\beta^B](\bullet)U \,|\!)\ (T'(\!|\, [\alpha^{A\to B}](\bullet) \leftarrow [\beta^B](\bullet)U \,|\!))) \\
[\alpha^{A\supset B}]T & (\!|\, [\alpha^{A\to B}](\bullet) \leftarrow [\beta^B](\bullet)U \,|\!) & \triangleq & [\beta^B](T\ U) \\
[\gamma^C]T & (\!|\, [\alpha^{A\to B}](\bullet) \leftarrow [\beta^B](\bullet)U \,|\!) & \triangleq & [\gamma^C](T(\!|\, [\alpha^{A\to B}](\bullet) \leftarrow [\beta^B](\bullet)U \,|\!)), \text{ if } \gamma^C \neq \alpha^{A\supset B} \\
\mu\alpha^{A\supset B}.N & (\!|\, [\alpha^{A\to B}](\bullet) \leftarrow [\beta^B](\bullet)U \,|\!) & \triangleq & \mu\alpha^{A\supset B}.N \\
\mu\gamma^C.N & (\!|\, [\alpha^{A\to B}](\bullet) \leftarrow [\beta^B](\bullet)U \,|\!) & \triangleq & \mu\gamma^C.(N(\!|\, [\alpha^{A\to B}](\bullet) \leftarrow [\beta^B](\bullet)U \,|\!)), \text{ if } \gamma^C \neq \alpha^{A\supset B}
\end{array}
$$

Rule R_1 corresponds to β-reduction in the λ-calculus. R_2 corresponds to structural reduction. In [Sau10], the above rules are named β, μ, ρ and θ respectively.

$\lambda\mu$ satisfies the standard properties of the λ-calculus, in particular confluence of both the typed and untyped calculus [Par92] and Type Preservation [Par92]. SN also holds for the typed $\lambda\mu$-calculus [Par97].

Proposition 2.5.9 Typed $\lambda\mu$ is SN.

Chapter 3

Combinatory Logics for Lambda Calculi with Patterns

In this chapter we explore a combinatory logic system for the λ-calculus with patterns λP. As discussed in the Introduction, our aim is to achieve a combinator-based rewriting system that can simulate the abstraction mechanism of the λP-calculus.

We start in Section 3.1 with a detailed overview of λP. We then introduce the CL_P rewriting system in Section 3.2. Section 3.3 treats the translation procedure for representing λP-terms and simulating the use of abstractions. Section 3.4 presents the equational theory associated to CL_P, and briefly discusses extensionality. In Section 3.5 we introduce a type system for CL_P, for which we prove Type Preservation and Strong Normalisation, and consider the addition of constructors to the calculus, keeping the same spirit. Some variants of the system are presented in Section 3.6; particularly, in Section 3.6.2, we present a way to define variants of the calculus with different notions of pattern-matching without losing confluence. In Section 3.7, we introduce the $CL_\star$-calculus, a variant of CL_P in which pattern matching is handled explicitly by means of a matching operator. Finally, in Section 3.8, we discuss other pattern calculi and how CL_P relates to them.

3.1 The λP-Calculus

We recall the set of terms of the λP-calculus, the λP-terms, from the Introduction:

$$M, N, P ::= x \mid MN \mid \lambda P.M$$

In the abstraction $\lambda P.M$, P is the *pattern* and M the *body*. Applications are left-associative, as per the usual convention.

Definition 3.1.1 (Free variables of λP-terms) Free variables of terms (extended as expected to sets of terms) are defined as follows:

$$\begin{array}{lcl} \mathsf{FV}(x) & \triangleq & \{x\} \\ \mathsf{FV}(MN) & \triangleq & \mathsf{FV}(M) \cup \mathsf{FV}(N) \\ \mathsf{FV}(\lambda P.M) & \triangleq & \mathsf{FV}(M) \backslash \mathsf{FV}(P) \end{array}$$

The notions of **substitution**, **domain** and **range** are analogous to those presented in Definition 2.2.5.

Definition 3.1.2 (Simultaneous substitution over λP-terms) The application of a substitution $\sigma = \{x_i \leftarrow N_i\}_{i=1,\ldots,n}$ to a term M, denoted $\sigma(M)$ or M^σ, is defined as follows:

$$\begin{array}{lcll} \sigma(x) & \triangleq & N_i, & \text{if } \ x = x_i \\ \sigma(x) & \triangleq & x, & \text{if } \ x \neq x_1, \ldots, x_n \\ \sigma(MN) & \triangleq & \sigma(M)\sigma(N) & \\ \sigma(\lambda P.M) & \triangleq & \lambda P.\sigma(M) & \end{array}$$

We assume the expected **free variable convention** over σ: $(\mathsf{dom}(\sigma) \cup \mathsf{FV}(\mathsf{ran}(\sigma))) \cap \mathsf{FV}(P) = \emptyset$.

One-hole λP-contexts are λP-terms with a unique "hole" which can occur anywhere in the term except inside abstraction patterns:

$$C ::= \Box \mid MC \mid CM \mid \lambda P.C$$

Reduction is given by the β_P relation generalizing the original β-reduction as follows:

$$(\lambda P.M)P^\sigma \rightarrow_{\beta_P} M^\sigma$$

We use $M \rightarrow_{\beta_P} N$ to denote that M reduces in one β_P-step to N, that is: $M = C[(\lambda P.R)P^\sigma]$ and $N = C[R^\sigma]$ where C is some context. In this case $(\lambda P.R)P^\sigma$ is called a β_P-**redex** and R^σ the β_P-**reduct**. Note that reduction can occur on either side of an application, and only on the right side (body) of an abstraction. This calculus does not allow reduction inside the patterns.

As exemplified in the Introduction, if arbitrary terms were permitted as patterns, the calculus would not be confluent. Two restrictions have been defined in order to ensure confluence of the calculus [KvOdV08]. We discuss first a more general condition, RPC, and then a syntax-driven restriction called RPC^+. The first requires the auxiliary notion of **simultaneous reduction** $\multimap$ allowing to contract an arbitrary set of pairwise non-overlapping redexes simultaneously:

Definition 3.1.3 (Simultaneous Reduction) The relation $M \multimap N$ is inductively defined as follows:

$$\frac{}{M \multimap M} \qquad \frac{M \multimap M' \quad N \multimap N'}{MN \multimap M'N'} \qquad \frac{M \multimap M'}{\lambda P.M \multimap \lambda P.M'}$$

$$\frac{M \multimap M' \quad N_1 \multimap N_1' \quad \ldots \quad N_k \multimap N_k'}{(\lambda P.M)(P\{x_1 \leftarrow N_1, \ldots x_k \leftarrow N_k\}) \multimap \ M'\{x_1 \leftarrow N_1', \ldots x_k \leftarrow N_k'\}}$$

For instance, $I(I(IK)) \multimap IK$ – with I defined as $\lambda x.x$ and K as $\lambda x.\lambda y.x$ – by simultaneously contracting the outer and inner redexes, however it is not the case that $IIK \multimap K$ since the redex IK has been created by the first step. The following is the first of the two conditions ensuring confluence in λP:

Definition 3.1.4 (Rigid Pattern Condition) A set of patterns satisfies the rigid pattern condition (RPC), if for any pattern P in it and for any substitution of terms $N_1, ..., N_n$ for the free variables $x_1, ..., x_n$ of P we have:

$$P\{x_1 \leftarrow N_1, \ldots, x_n \leftarrow N_n\} \multimapinv P' \Rightarrow P' = P\{x_1 \leftarrow N'_1, \ldots, x_n \leftarrow N'_n\} \wedge N_i \multimapinv N'_i (1 \leq i \leq n)$$

Theorem 6 in [KvOdV08] β_P is confluent if all patterns satisfy RPC.

The second condition is as follows:

Definition 3.1.5 (RPC^+) The set RPC^+ consists of all λ-terms which:

1. are linear: (free) variables occur at most once;
2. are in normal form: they contain no β-pattern redex; and
3. have no active variables: they have no subterms of the form xM with x free.

For example, $\lambda x.x$, $\lambda x.y$ and $\lambda x.\lambda y.x$ satisfy RPC^+ (and can thus be used as patterns), but $\lambda(\lambda x.y).y$, $\lambda x.\lambda y.xy$ and $(\lambda y.x)z$ do not.

Theorem 7 in [KvOdV08] The set RPC^+ satisfies RPC and thus yields a confluent calculus.

3.2 The CL_P-Calculus

The set of CL_P-terms is described by the grammar:

$$M, N ::= x \mid K_M \mid S_M \mid \Pi^1_{MN} \mid \Pi^2_{MN} \mid MN$$

where x ranges over a given countably infinite set of variables $\mathcal{X}$. Application is left-associative, as usual. K_M, S_M, Π^1_{MN} and Π^2_{MN} will be the **combinators**, or primitive functions of our calculus. The combinators K_M and S_M are pattern-decorated versions of the CL combinators. Π^1_{MN} and Π^2_{MN}, which will be called **projectors**, have been introduced in order to extract information from an application by means of decomposition. Although the grammar admits arbitrary terms as subindices in K, S, Π^1 and Π^2, they will shortly be circumscribed to a subset that we shall dub **patterns** (Def. 3.2.5). We will use the letters P and Q to refer to patterns.

The aim of the set of CL_P-terms is to mimic the 3 steps required for λP-reduction:

1. matching the pattern against the argument, which if successful
2. yields bindings of the free variables of the pattern to subterms of the argument,
3. which are then applied to the body.

In CL_P, the combinators of the form S_P and K_P serve for the third phase, the projectors for the second phase, and the subscripts for the first phase.

Remark 3.2.1 Note that we have an infinite number of combinators, one S_P, K_P, Π^1_{PQ}, Π^2_{PQ} for each possible P and Q. For example, the CL_P-terms K_x, K_{K_y} and K_{S_y} are all different combinators.

Definition 3.2.2 (Free variables of CL_P-terms) Free variables are defined as follows:

$$\begin{array}{lcl} \mathsf{FV}(x) & \triangleq & \{x\} \\ \mathsf{FV}(MN) & \triangleq & \mathsf{FV}(M) \cup \mathsf{FV}(N) \\ \mathsf{FV}(K_P) = \mathsf{FV}(S_P) = \mathsf{FV}(\Pi^1_{PQ}) = \mathsf{FV}(\Pi^2_{PQ}) & \triangleq & \emptyset \end{array}$$

In the sequel of this chapter we use **term** for CL_P-term where there is no ambiguety. We will also adopt the following conventions:

- All free variables are considered to be different from the names of the variables in the patterns, and the latter different from each other, even if their names clash. There are no bound variables in this calculus; the presence of a variable in a pattern only serves to indicate that any term is a valid match.[1]
- We consider that two patterns are equal if they differ in nothing but the names of their variables, and we will work modulo renaming of variables in patterns (Def. 3.2.6).

Variables in terms are used in the same way as in CL. Variables within patterns – present as subscripts – play a slightly different role, as they represent a subterm of the pattern which can be matched (Definition 3.2.3) by any given term. They have no correspondence with any variables in the term in which they are involved, and as such they do not act as binders. In other words, the same variable cannot appear as both a subterm and (part of) a subscript within a term.

Definition 3.2.3 Substitution over CL_P-terms and patterns is defined as follows:

$$\begin{array}{lclcl} x^\sigma & \triangleq & \sigma(x) & \qquad & (\Pi^1_{QR})^\sigma \triangleq \Pi^1_{QR} \\ K^\sigma_P & \triangleq & K_P & & (\Pi^2_{QR})^\sigma \triangleq \Pi^2_{QR} \\ S^\sigma_P & \triangleq & S_P & & (MN)^\sigma \triangleq M^\sigma N^\sigma \end{array}$$

A term M is an **instance** of a pattern P (also, M **matches** the pattern P) if $\exists \sigma . P^\sigma = M$.

Remark 3.2.4 Note that substitutions *do not affect* the subscripts of combinators: each combinator K_P, S_P, Π^1_{QR}, Π^2_{QR} is a constant for all patterns P, QR. Patterns are affected by substitutions when used as terms, but not when used as subscripts. For example, the term Kx will be affected by substitutions which affect the variable x, but the term Π^1_{Kx} will not.

We denote the unification relation as $\mathring{=}$. For example, $P \mathring{=} M$ should be read as "P unifies with M". Similarly, $P \not\mathring{=} M$ should be read as "P does not unify with M". The purpose of our use of unification in the definition that follows is to ensure that no unwanted redexes are formed (that is, that terms which match a pattern do not reduce to terms which no longer match it).

[1] An alternative would be to omit variables in patterns, replacing each variable by a $\star$. We do not use this approach in CL_P as we are interested in providing a TRS formulation.

Definition 3.2.5 (Patterns) The set of **patterns** is defined as follows:

$$
\begin{array}{lll}
P, Q, P_1, \cdots P_n & ::= x & \\
& \mid\ K_P P_1 \cdots P_n & \text{where } n < 2 \vee P_2 \not\equiv P \\
& \mid\ S_P P_1 \cdots P_n & \text{where } n < 3 \vee P_3 \not\equiv P \\
& \mid\ \Pi^1_{PQ} P_1 \cdots P_n & \text{where } n = 0 \vee P_1 \not\equiv PQ \\
& \mid\ \Pi^2_{PQ} P_1 \cdots P_n & \text{where } n = 0 \vee P_1 \not\equiv PQ
\end{array}
$$

with $n \geq 0$ and $\mathsf{FV}(P_i) \cap \mathsf{FV}(P_j) = \emptyset$ for all $1 \leq i, j \leq n$ s.t. $i \neq j$ (this means, by recursion, that all patterns are linear).

A pattern of the form $P_1 P_2$ is called an **application pattern**. We abbreviate K_x and S_x, as K and S, respectively. Examples of patterns are: K, S, $K_{K_{S_{K^x}}}$ SKK, $S_S KK$, Π^1_{KS}, $\Pi^2_{K_K x}$. The terms $K_S SK$, $S_K KSS$, $\Pi^1_{KS} K$, $\Pi^2_{K_K x}(KS)$ are also patterns because, while they have the number of arguments required to form a redex (or more), they contain pattern-matching failures[2] which prevent them from unifying with the left-hand side of any reduction rules: more precisely, K does not unify with S, KS nor K_K. On the other hand, xy, yS, $K_S SS$, $S_K KSK$, $\Pi^1_{KS}(KS)$ and $\Pi^2_{K_K x}(K_K S)$ are not patterns, as they all have subterms which unify with the left-hand side of a reduction rule. And, of course, non-linear terms like $Sx(Kx)$ or Syy are not patterns either.

The patterns defined by this syntax are those which conform to a restriction named RPC^{++}, which will be explained in detail in Section 3.2.1, once the reduction rules have been introduced.

Definition 3.2.6 (α-equivalence over CL_P-terms) The fact that variable names within a pattern are irrelevant induces an α-equality relation between patterns, which is defined as follows:

$$
\begin{array}{llll}
x & =_\alpha & y & \\
K_P & =_\alpha & K_Q, & \text{if } P =_\alpha Q \\
S_P & =_\alpha & S_Q, & \text{if } P =_\alpha Q \\
\Pi^1_{PQ} & =_\alpha & \Pi^1_{P'Q'}, & \text{if } P =_\alpha P' \text{ and } Q =_\alpha Q' \\
\Pi^2_{PQ} & =_\alpha & \Pi^2_{P'Q'}, & \text{if } P =_\alpha P' \text{ and } Q =_\alpha Q' \\
PQ & =_\alpha & P'Q', & \text{if } P =_\alpha P' \text{ and } Q =_\alpha Q' \\
P & \neq_\alpha & Q, & \text{in any other case.}
\end{array}
$$

Equality between arbitrary CL_P-terms is defined in a similar way, except that different variables are treated as different terms. Only the subscripts of combinators are subject to α-conversion.

Definition 3.2.7 We use the following function to measure the size of a term:

$$
\begin{array}{llll}
|x| & \triangleq & 1 & \\
|K_P| = |S_P| & \triangleq & 1 + |P|, & \text{for every pattern } P \\
|\Pi^1_{QR}| = |\Pi^2_{QR}| & \triangleq & 1 + |Q| + |R|, & \text{for every patterns } Q,\ R \text{ such that } QR \text{ is a pattern} \\
|MN| & \triangleq & |M| + |N| &
\end{array}
$$

[2]Not unifying with the left-hand-side of a reduction rule by itself does not guarantee a matching failure, since a term which does not unify with another may reduce to a term which does. However, this is not possible for patterns, since the definition is recursive and $P_1, \ldots, P_n$ must be patterns too.

3.2.1 Reduction in CL_P

The aim of this subsection is to introduce a rewriting system based on the above combinators, which will simulate λP in the sense of combinatorial completeness, as well as mimic the pattern matching of this calculus.

Definition 3.2.8 (W_P-reduction) **W_P-reduction** (denoted as $\rightarrow_{W_P}$) is defined as the following TRS over the signature given by the syntax presented at the beginning of Section 3.2:

$$\begin{array}{lcll} K_P x P & \rightarrow & x, & x \notin \mathsf{FV}(P) \\ S_P x y P & \rightarrow & xP(yP), & x, y \notin \mathsf{FV}(P) \\ \Pi^1_{PQ}(PQ) & \rightarrow & P & \\ \Pi^2_{PQ}(PQ) & \rightarrow & Q & \end{array}$$

where P and Q range over patterns, and the application PQ – where used – is also a pattern.

These rules are schematic, since P and Q range over an infinite set of patterns. We have, in fact, an infinite number of rules captured by a finite number of schemas. Before presenting some examples, three easily verifiable properties of reduction: reduction is closed over CL_P, it does not create new free variables and is well-defined over α-equivalence classes of terms.

Lemma 3.2.9 Suppose M is a CL_P-term and $M \rightarrow_{W_P} N$. Then:

1. $N \in CL_P$.
2. $\mathsf{FV}(N) \subseteq \mathsf{FV}(M)$.
3. $M =_\alpha M'$ implies there exists a term N' such that $M' \rightarrow_{W_P} N'$ and $N =_\alpha N'$.

Example 3.2.10 We will now show some reduction and pattern-matching examples.

- $K_P x^\sigma P^\sigma \rightarrow_{W_P} x^\sigma$ (for any substitution σ).
- $S_{Kx}\Pi^1_{Ky}\Pi^2_{KS}(KS) \rightarrow_{W_P} \Pi^1_{Ky}(KS)(\Pi^2_{KS}(KS)) \rightarrow_{W_P} K(\Pi^2_{KS}(KS)) \rightarrow_{W_P} KS$.
  The substitutions used here are $\{x \leftarrow S\}$, $\{y \leftarrow S\}$ and $\emptyset$ respectively.
- $S_{Kx}\Pi^1_{Ky}\Pi^2_{KS}(KS) \rightarrow_{W_P} \Pi^1_{Ky}(KS)(\Pi^2_{KS}(KS)) \rightarrow_{W_P} \Pi^1_{Ky}(KS)S \rightarrow_{W_P} KS$ is another possible reduction path.
- $\Pi^2_{S\Pi^1_{Kx}}(S\Pi^1_{Ky})(KS_K) \rightarrow_{W_P} \Pi^1_{Ky}(KS_K) \rightarrow_{W_P} K$. No substitution is involved in the first step, since $S\Pi^1_{Kx}$ is the same as $S\Pi^1_{Ky}$ due to α-equivalence between the subscripts. In the second step, we use the substitution $\{y \leftarrow S_K\}$.
- $\Pi^2_{K_K S}(KS)$ does not reduce, as KS does not match $K_K S$.
- $\Pi^2_{S\Pi^1_{Kx}}(S\Pi^1_{KS})$ does not reduce, since Π^1_{KS} does not match Π^1_{Kx} and thus $S\Pi^1_{KS}$ does not match $S\Pi^1_{Kx}$. Note that, although KS matches Kx, the same does not hold for combinators which have these patterns as their subscripts. This is because substitutions do not affect subscripts.

Note that $P^\sigma Q^\sigma$ is the same as $(PQ)^\sigma$ by definition. This means that for a term of the form $\Pi^1_P M$ to be a redex, both P and M must be applications[3], and M must be an instance of P. A term is said to be *active* if it is used as the left-hand side of an application. The reason why we have restricted the syntax of our terms so that Π^1 and Π^2 may only have an application as their pattern should now be clear: according to the rules, an active projector with a non-application pattern (i.e. a variable or combinator) would never execute (that is, the projector applied to an argument would not reduce).

Proposition 3.2.11 CL_P is an extension of CL.

Proof.- There is a direct mapping from CL to CL_P: variables translate to themselves, and the combinators S and K translate as S_x and K_x respectively (this mapping is univocal modulo α-conversion). Since a variable can be matched by any term, the reduction rules for S_x and K_x behave in the same way as the reduction rules for S and K in CL. □

The $\boldsymbol{RPC^{++}}$ restriction and confluence in $\boldsymbol{CL_P}$

In order to prove confluence (Corollary 3.2.17) and just as in λP, it is necessary to impose restrictions over the patterns. Without them, confluence would not hold: for instance, the term $\Pi^1_{xy}(KSK)$ would reduce to two different normal forms: KS and $\Pi^1_{xy}S$. Following Van Oostrom's RPC^+ restriction for λP, we call this set of restrictions $\boldsymbol{RPC^{++}}$. The aim of RPC^{++}, just like RPC^+, is to define a syntax-based, easily verifiable set of restrictions that can guarantee the confluence of the calculus.

Definition 3.2.12 The set of **application subterms** $\mathtt{AS}(M)$ in a CL_P-term M is defined as follows:

$$\begin{array}{lcll}
\mathtt{AS}(x) & \triangleq & \emptyset, & \\
\mathtt{AS}(M) & \triangleq & \emptyset, & \text{if } M \text{ is a combinator} \\
\mathtt{AS}(MN) & \triangleq & \{MN\} \cup \mathtt{AS}(M) \cup \mathtt{AS}(N) &
\end{array}$$

The set of **application patterns** $\mathtt{AP}(M)$ in a term M is defined as follows:

$$\begin{array}{lcl}
\mathtt{AP}(x) & \triangleq & \emptyset \\
\mathtt{AP}(K_P) & \triangleq & \mathtt{AS}(P) \cup \mathtt{AP}(P) \\
\mathtt{AP}(S_P) & \triangleq & \mathtt{AS}(P) \cup \mathtt{AP}(P) \\
\mathtt{AP}(\Pi^1_{PQ}) & \triangleq & \mathtt{AS}(PQ) \cup \mathtt{AP}(P) \cup \mathtt{AP}(Q) \\
\mathtt{AP}(\Pi^2_{PQ}) & \triangleq & \mathtt{AS}(PQ) \cup \mathtt{AP}(P) \cup \mathtt{AP}(Q) \\
\mathtt{AP}(MN) & \triangleq & \mathtt{AP}(M) \cup \mathtt{AP}(N)
\end{array}$$

Definition 3.2.13 ($\boldsymbol{RPC^{++}}$) A term M satisfies RPC^{++} if every $N \in (\mathtt{AS}(M) \cup \mathtt{AP}(M))$:

1. is linear, i.e. no variable appears more than once,

[3] We do not allow terms like Π^1_x or Π^2_x, since their presence could easily break the confluence of the calculus. For example, the term $\Pi^1_x KKK$ would reduce to two distinct normal forms: KK and $\Pi^1_x K$.

2. has no active variables, i.e. no subterms of the form xN' with $x \in \mathcal{X}$,

3. does not unify[4] with the left-hand-side of a W_P-rule.

We do not require anything of non-application patterns (except, of course, that their subscripts – when present – satisfy RPC^{++}). A variable, for example, will trivially satisfy the RPC^{++} restriction. The second and third conditions imply that application patterns will have no active variables (i.e. they will have no subterms of the form xM), and that they will be normal forms. These two results, along with linearity (our first condition), constitute a translation of RPC^{+}, minus the requirement for all patterns to be λ-terms, to CL_P.

Remark 3.2.14 M is a pattern if and only if M satisfies RPC^{++}. This can be easily verified by looking at the syntax of the patterns in Definition 3.2.5 and the definition of RPC^{++}.

Our patterns are also rigid in that an instance of a given pattern may only reduce to another instance of the same pattern (see Lemma 3.6.6).

We briefly show with examples that our restrictions are well-motivated: breaking any of the last two conditions can result in a calculus that is not even locally confluent. (See for example the derivations starting from the terms $K_{xy}S(KKz)$, $K_{Kxz}y(KSS)$ and $K_{\Pi^1_{KK}xy}S(\Pi^1_{KK}(KK)z)$).

While the use of non-linear patterns may not lead to two different normal forms, it does break the confluence of the calculus. See Klop's standard example [Klo76, KvOdV08], and define D as the term $S(K(S(K(K_{Sxx}E))))S$, where E is some term chosen to indicate equality.

This is the most general definition we will use for the set of patterns for CL_P. It is also possible to work with proper subsets of this set, in order to avoid dealing with unification and multiple levels of subscripts. See Section 3.6 for further details.

Lemma 3.2.15 W_P is orthogonal in CL_P.

Proof.- Left-linearity of the rules is immediate: wherever variables appear explicitly in a rule schema, they are required to be fresh with respect to the pattern; and patterns are required to be linear by RPC^{++}. The absence of critical pairs is a consequence of the restrictions that require all patterns – and their applicative subterms – not to unify with the left-hand-side of a W_P-rule. Since each rule of the TRS unifies with a rule in the original formulation, and all patterns satisfy RPC^{++}, this implies that no instance of a pattern, nor any of its subterms, matches the left side of any rule. □

Remark 3.2.16 Note that orthogonality follows crucially from the fact that the patterns involved in the rules satisfy RPC^{++}.

The following corollary states the confluence of the calculus for every set of patterns satisfying RPC^{++}. It follows from the fact that no new patterns appear upon reduction and the previous lemma.

Corollary 3.2.17 Let Φ be any subset of the CL_P-patterns, and let $CL_P(\Phi)$ be the CL_P-calculus restricted to the terms whose patterns belong to Φ. Then, $CL_P(\Phi)$ is confluent.

[4]Remember we are working modulo renaming of variables in patterns, so the pattern Ky will unify with yS even though K does not unify with S.

3.3 Translations between λP and CL_P

In order to prove that W_P-reduction represents an abstraction mechanism, we will define translations between the two systems and then show the relationship between their respective reduction relations.

3.3.1 Translation from λP to CL_P

Definition 3.3.1 (From λP-terms to combinators) Let M be a λP-term. Its corresponding combinator, denoted M_{CL}, is defined as follows:

$$\begin{array}{rcl} x_{CL} & \triangleq & x \\ (MN)_{CL} & \triangleq & M_{CL}N_{CL} \\ (\lambda M.N)_{CL} & \triangleq & \lambda^* M_{CL}.N_{CL} \end{array}$$

with λ^* defined recursively as follows:

$$\begin{array}{lcll} 1)\,\lambda^* x.x & \triangleq & SKK & \\ 2)\,\lambda^* P.M & \triangleq & K_P M, & \text{if } \mathsf{FV}(P) \cap \mathsf{FV}(M) = \emptyset \\ 3)\,\lambda^* PQ.x & \triangleq & S_{PQ}(K\lambda^* P.x)\Pi^1_{PQ}, & \text{if } x \in (\mathsf{FV}(P) \backslash \mathsf{FV}(Q)) \\ 4)\,\lambda^* PQ.x & \triangleq & S_{PQ}(K\lambda^* Q.x)\Pi^2_{PQ}, & \text{if } x \in \mathsf{FV}(Q) \\ 5)\,\lambda^* P.MN & \triangleq & S_P(\lambda^* P.M)(\lambda^* P.N), & \text{if } \mathsf{FV}(P) \cap \mathsf{FV}(MN) \neq \emptyset \end{array}$$

This translation extends to substitutions σ in λP as expected: $\sigma_{CL} \triangleq \{x \leftarrow M_{CL} \mid x \leftarrow M \in \sigma\}$.

Rules 1, 2 and 5 are inspired in the original translation from λ-calculus to CL. Rules 3 and 4 emerge from the necessity of decomposing application patterns. An abstraction of the form $\lambda PQ.x$, whose pattern expects an application, should be transformed into a term which executes projections until the location of x is found (either inside P or inside Q). After projecting, say, to the left (if $x \in \mathsf{FV}(P)$), the CL_P-derivation continues by letting the projected argument match with the pattern P: this is done by translating the term $\lambda P.x$ recursively. A term of the form $\lambda PQ.x$ should finally locate the corresponding instance of x inside the matching argument. Thus, the resulting translation will be a composition of S's, K's and projectors.

The fact that the function is well defined can be derived from a simple observation (all recursive calculations of λ^* are carried out over smaller terms). Furthermore, it can be proved with a straightforward case by case analysis that any term of the form $\lambda P.M$ fits the hypotheses of one – and only one – of these 5 rules.

Clauses 1 to 5 are general enough to handle the full syntax without the RPC^{++} restriction. On the other hand, restricting the domain to terms with patterns will resolve the apparent lack of symmetry of rules 3 and 4: since a free variable never appears more than once in a pattern, their conditions can be simplified to $x \in \mathsf{FV}(P)$ and $x \in \mathsf{FV}(Q)$ respectively. Otherwise, without the restriction, one can always choose either case and the result will behave in the same way.

Remark 3.3.2 The following rule may be added as a shortcut to optimize reductions (and reduce the size of the translated term):

$$6)\,\lambda^* x.Mx \quad \triangleq \quad M, \quad \text{if } x \notin \mathsf{FV}(M)$$

Clause 6 is optional, since all terms that can be translated with this rule can also be translated via rule 5, but it can greatly reduce the amount of reduction steps required to reach a normal form (whenever one exists). One may prove that the resulting terms are functionally equivalent, by showing that:

$$S_x(\lambda^* x.M)(\lambda^* x.x)N \quad \twoheadrightarrow_{W_P} MN, \quad \text{if } x \notin \mathsf{FV}(M)$$

Indeed:

$$\begin{array}{lll} S_x(\lambda^* x.M)(\lambda^* x.x)N & =_{2)} & S_x(K_x M)(\lambda^* x.x)N \\ & =_{1)} & S_x(K_x M)(SKK)N \\ & = & S(KM)(SKK)N \\ & \to_{W_P} & (KMN)(SKKN) \\ & \to_{W_P} & M(SKKN) \\ & \twoheadrightarrow_{W_P} & MN \end{array}$$

Nevertheless, keeping this rule would result in a non-deterministic definition, unless the clauses are followed in a prescribed order by verifying the conditions of rule 6 before attempting to apply rule 5.

Just as in CL, it can be easily proved that $SKKN \twoheadrightarrow_{W_P} N$ for every term N. For this reason, we will allow the term SKK to be abbreviated as I to represent the identity function. Note that this only makes sense when the patterns involved are variables, otherwise pattern matching may fail. More generally, in CL_P we have for any pattern P:

$$S_P KKP^\sigma \to_{W_P} KP^\sigma(KP^\sigma) \to_{W_P} P^\sigma$$

The expression I_P is used to denote the term $S_P KK$.

Remark 3.3.3 The definition of abstraction λ^* extends the one for CL, i.e. for $M \in CL$ and $x \in \mathcal{X}$, $\lambda^* x.M$ coincides with the classical notion.

Remark 3.3.4 While it is true that every abstraction in λP fits the hypotheses of one of the λ^* rules, terms which do not satisfy the RPC^+ restriction may translate to terms with ill-formed patterns. For example:

$$(\lambda xx.x)_{CL} = \lambda^* xx.x =_{4)} S(K\lambda^* x.x)\Pi^2_{xx} =_{1)} S(KI)\Pi^2_{xx}$$

but xx is **not** a pattern in CL_P.

Lemma 3.3.5 If P is a CL_P-pattern, then so is $\lambda^* x.P$.

Proof.- By induction on P.

- If $P = x$, then $\lambda^* x.P = SKK$, which is a CL_P-pattern.
- If $x \notin \mathsf{FV}(P)$, then $\lambda^* x.P = KP$, also a CL_P-pattern.
- If $P \neq x$ and $x \in \mathsf{FV}(P)$, then P must be an application $P_1 P_2$ (with P_1 and P_2 patterns) and $\lambda^* x.P = S(\lambda^* x.P_1)(\lambda^* x.P_2)$. By IH, both $\lambda^* x.P_1$ and $\lambda^* x.P_2$ are CL_P-patterns. Therefore, so is $S(\lambda^* x.P_1)(\lambda^* x.P_2)$.

□

Proposition 3.3.6 Every λP-pattern which satisfies RPC^+ (Def. 3.1.5) translates into a CL_P-pattern.

Proof.- By induction on the pattern. Keep in mind that, since it satisfies RPC^+, it must be a λ-term, and be either a variable or an abstraction of the form $\lambda x.P$ with P a pattern satisfying RPC^+(an application would contain either an active variable or a redex).

- If the pattern is a variable, then its translation is also a variable, which is a CL_P-pattern.
- If it is an abstraction $\lambda x.P$, then its translation is $\lambda^* x.P_{CL}$. Since P satisfies RPC^+, then by IH P_{CL} is a CL_P-pattern. Thus, by Lemma 3.3.5, so is $\lambda^* x.P_{CL}$.

□

On the other hand, some λP-terms which do **not** satisfy RPC^+ can still be translated to CL_P-patterns. Consider for example the term $(\lambda x.\lambda y.x)w$. This term does not satisfy RPC^+, as it is not a normal form. However, when we translate it to CL_P (using rules 2 and 6 of λ^*), we obtain the term $K_y w$, which satisfies RPC^{++}. In this sense, we can say that the RPC^{++} restriction is more general than RPC^+ (it allows a strictly larger set of patterns).

Even without rule 6, there are CL_P-patterns which are not a direct CL_P translation of any λP-patterns. For example, there is no λP-term M such that $M_{CL} = \Pi^1_{KS}$ (it is immediate from the definition of $-_{CL}$ that the result of this translation will never be a single projector), and yet the term Π^1_{KS} satisfies RPC^{++}.

Another alternative is to replace rule 1 by the more general rule:

$$1') \quad \lambda^* P.P \quad \triangleq \quad I_P = S_P KK, \quad \text{if } \mathsf{FV}(P) \neq \emptyset$$

and since I_P is the identity restricted to the set of terms matching P, the process will yield a more efficient translation. Its condition ensures it does not overlap with rule 2, but it will still overlap with rule 5, resulting in a non-deterministic system, just like the system that results of including rule 6. Naturally, precedences among the rules may be defined in order to regain determinism.

Remark 3.3.7 λP-patterns which satisfy RPC^+ will translate into a more restricted set of patterns, since they are λ-terms and therefore translate into CL-terms. The grammar for the patterns which result from such a translation is: $P ::= x \mid K \mid S \mid KP \mid SP \mid SPP$, maintaining linearity as usual. We will refer to this new set as RPC^+-patterns.

Further results related to the translation

Lemma 3.3.8 ($-_{CL}$ preserves free variables) $\mathsf{FV}(M) = \mathsf{FV}(M_{CL})$.

Proof.- By induction on the definition of M_{CL} (free variables may only appear in x_{CL}, $(MN)_{CL}$ and rules 2 and 5 of λ^*). □

Lemma 3.3.9 (Commutation of $-_{CL}$ and substitution) $(M^\sigma)_{CL} = (M_{CL})^{\sigma_{CL}}$.

Proof.- By induction on the size of M. Our IH is that $(M^\sigma)_{CL} = (M_{CL})^{\sigma_{CL}}$ for every λP substitution σ and for every λP-term M which is strictly smaller than the term we are analyzing.

- $(x^\sigma)_{CL} = (\sigma(x))_{CL} = \sigma_{CL}(x) = x^{\sigma_{CL}} = (x_{CL})^{\sigma_{CL}}$.
- $((MN)^\sigma)_{CL} = (M^\sigma N^\sigma)_{CL} = (M^\sigma)_{CL}(N^\sigma)_{CL} =_{\text{IH}} (M_{CL})^{\sigma_{CL}}(N_{CL})^{\sigma_{CL}} = (M_{CL}N_{CL})^{\sigma_{CL}} = ((MN)_{CL})^{\sigma_{CL}}$.
- For $((\lambda P.M)^\sigma)_{CL}$ we will need to analyze all the possible cases:
  - $\lambda P.M$ is of the form $\lambda x.x$. Then
    $$((\lambda x.x)^\sigma)_{CL} = (\lambda x.x)_{CL} = SKK = (SKK)^{\sigma_{CL}} = ((\lambda x.x)_{CL})^{\sigma_{CL}}$$
  - $\lambda P.M$ is such that $\mathsf{FV}(P) \cap \mathsf{FV}(M) = \emptyset$. Then
    $$\begin{array}{rcl} ((\lambda P.M)^\sigma)_{CL} & = & (\lambda P.M^\sigma)_{CL} \\ & =_{\text{IH}} & \lambda^* P_{CL}.(M^\sigma)_{CL} \\ & = & \lambda^* P_{CL}.(M_{CL})^{\sigma_{CL}} \\ & = & K_{P_{CL}}((M_{CL})^{\sigma_{CL}}) \\ & = & (K_{P_{CL}} M_{CL})^{\sigma_{CL}} \\ & = & (\lambda^* P_{CL}.M_{CL})^{\sigma_{CL}} \\ & = & ((\lambda P.M)_{CL})^{\sigma_{CL}}. \end{array}$$
    Here we have used Lemma 3.3.8, and the usual variable convention over σ.
  - $\lambda P.M$ is of the form $\lambda QR.x$ with $x \in \mathsf{FV}(Q)\backslash\mathsf{FV}(R)$. In this case the term $\lambda QR.x$ has no free variables, and – by Lemma 3.3.8 – nor does $(\lambda QR.x)_{CL}$. Therefore, $((\lambda QR.x)^\sigma)_{CL} = (\lambda QR.x)_{CL} = ((\lambda QR.x)_{CL})^{\sigma_{CL}}$.
  - $\lambda P.M$ is of the form $\lambda QR.x$ with $x \in \mathsf{FV}(R)$. Analogous to the previous one.
  - $\lambda P.M$ is of the form $\lambda P.NT$ with $\mathsf{FV}(P) \cap \mathsf{FV}(NT) \neq \emptyset$. In this case $(\lambda P.NT)^\sigma = \lambda P.N^{\sigma'}T^{\sigma'}$ where $\sigma' = \sigma'|_{\mathsf{FV}(NT)\backslash\mathsf{FV}(P)}$. Since σ' is also a substitution in λP, we will be able to use it in our inductive hypothesis. Thus we have:
    $$\begin{array}{rcl} ((\lambda P.NT)^\sigma)_{CL} & = & (\lambda P.N^{\sigma'}T^{\sigma'})_{CL} \\ & = & \lambda^* P_{CL}.(N^{\sigma'})_{CL}(T^{\sigma'})_{CL} \\ & = & S_{P_{CL}}(\lambda^* P_{CL}.(N^{\sigma'})_{CL})(\lambda^* P_{CL}.(T^{\sigma'})_{CL}) \\ & = & S_{P_{CL}}(\lambda P.N^{\sigma'})_{CL}(\lambda P.T^{\sigma'})_{CL} \\ & = & S_{P_{CL}}((\lambda P.N)^{\sigma'})_{CL}((\lambda P.T)^{\sigma'})_{CL} \\ & =_{\text{IH}} & S_{P_{CL}}((\lambda P.N)_{CL})^{\sigma'_{CL}}((\lambda P.T)_{CL})^{\sigma'_{CL}} \\ & = & (S_{P_{CL}})^{\sigma'_{CL}}((\lambda P.N)_{CL})^{\sigma'_{CL}}((\lambda P.T)_{CL})^{\sigma'_{CL}} \\ & = & (S_{P_{CL}}((\lambda P.N)_{CL})((\lambda P.T)_{CL}))^{\sigma'_{CL}} \\ & = & (S_{P_{CL}}(\lambda^* P_{CL}.N_{CL})(\lambda^* P_{CL}.T_{CL}))^{\sigma'_{CL}} \\ & = & (\lambda^* P_{CL}.N_{CL}T_{CL})^{\sigma'_{CL}} \\ & = & ((\lambda P.NT)_{CL})^{\sigma'_{CL}} \\ & = & ((\lambda P.NT)_{CL})^{\sigma_{CL}} \end{array}$$
    Note that, since $\sigma' = \sigma|_{\mathsf{FV}(NT)\backslash\mathsf{FV}(P)} = \sigma|_{\mathsf{FV}(\lambda P.NT)}$, then $\sigma'_{CL} = \sigma_{CL}|_{\mathsf{FV}(\lambda P.NT)} = \sigma_{CL}|_{\mathsf{FV}((\lambda P.NT)_{CL})}$.

□

Lemma 3.3.10 If $M_{CL} = P^\sigma$, then there is a substitution σ' in λP such that $\sigma = \sigma'_{CL}$.

Proof.- By induction on the pattern P, using the syntax for CL_P-patterns defined in Section 3.2.1.

- If $P = x$ with $x \in \mathcal{X}$, then $\sigma = \{x \leftarrow M_{CL}\}$. Take $\sigma' = \{x \leftarrow M\}$.
- Otherwise, P must be of the form $TP_1 \cdots P_n$, where T has no free variables and is therefore unaffected by σ, and $\mathsf{FV}(P_i) \cap \mathsf{FV}(P_j) = \emptyset \; \forall i \neq j$. It also holds that every P_i is a pattern by definition. Thus, $M_{CL} = TP_1^\sigma \cdots P_n^\sigma =_{\text{IH}} TP_1^{(\sigma_1)_{CL}} \cdots P_n^{(\sigma_n)_{CL}}$. Take $\sigma' = \bigcup_{i=1}^n \sigma_i|_{\mathsf{FV}(P_i)}$. Since P is linear, the substitution σ' is well-defined.

□

Proposition 3.3.11 ($-_{CL}$ preserves pattern matching) For every λP-term P such that P_{CL} is a CL_P-pattern, for every λP-term M: $(\exists \sigma \text{ s.t. } M = P^\sigma) \Leftrightarrow (\exists \sigma' \text{ s.t. } M_{CL} = P_{CL}^{\sigma'})$, where σ is a substitution in λP and σ' is a substitution in CL_P.

Proof.- The $\Rightarrow$-direction is a direct consequence of Lemma 3.3.9 (take $\sigma' = \sigma_{CL}$). The $\Leftarrow$-direction is a consequence of Lemmas 3.3.10 and 3.3.9. Take any σ s.t. $\sigma' = \sigma_{CL}$ (by Lemma 3.3.10, such a σ exists). □

3.3.2 Translation from CL_P to λP

Any CL_P-term M can be translated to a term M_λ in λP.

Definition 3.3.12 (From combinators to λP-terms) Let M be a CL_P-term. Its corresponding λP term, denoted M_λ, is defined as follows:

$$\begin{array}{lcll}
x_\lambda & \triangleq & x & \\
(K_P)_\lambda & \triangleq & \lambda x.\lambda P_\lambda.x, & x \text{ fresh} \\
(S_P)_\lambda & \triangleq & \lambda x.\lambda y.\lambda P_\lambda.xP_\lambda(yP_\lambda), & x \text{ and } y \text{ fresh} \\
(\Pi^1_{PQ})_\lambda & \triangleq & \lambda P_\lambda Q_\lambda.P_\lambda & \\
(\Pi^2_{PQ})_\lambda & \triangleq & \lambda P_\lambda Q_\lambda.Q_\lambda & \\
(MN)_\lambda & \triangleq & M_\lambda N_\lambda &
\end{array}$$

This translation can be extended to substitutions in the same way as the previous one:

$$\sigma_\lambda \quad \triangleq \quad \{x \leftarrow M_\lambda \mid x \leftarrow M \in \sigma\}$$

Note that the domains of substitutions are preserved by both translations ($-_{CL}$ and $-_\lambda$).

It can be observed that, just like the previous translation, $-_\lambda$ preserves free variables: For every CL_P-term M, $\mathsf{FV}(M) = \mathsf{FV}(M_\lambda)$. This is immediate from the first 2 lines of the above definition, the definitions of free variables in CL_P and λP and the fact that none of the lines of the definition of $-_\lambda$ introduces nor eliminates free variables (the variables introduced in the third and fourth lines are bound).

Remark 3.3.13 $-_\lambda$ is not the inverse of $-_{CL}$. Expanding the definitions shows that $((\lambda x.x)_{CL})_\lambda$ is not the same as $\lambda x.x$, and that $((S_P)_\lambda)_{CL}$ is not the same as S_P. What does hold, nevertheless, is that $(M_{CL})_\lambda \equiv_{\beta_P} M$ (or, if we consider rule 6 for λ^*, $(M_{CL})_\lambda \equiv_{\beta_P\eta} M$). In fact, if we use the original definition of λ^* (without accelerator rules), we have that $(M_{CL})_\lambda \twoheadrightarrow_{\beta_P} M$. We prove this later as Proposition 3.3.20.

3.3.3 Relationship between $\to_{W_P}$ and $\to_{\beta_P}$

We now analyze the connection between the W_P reduction in CL_P and the original β_P reduction in λP. We begin by proving that W_P can provide an abstraction mechanism (Corollary 3.3.15). We will also prove the following results:

- For all CL_P-terms M, N: if $M \to_{W_P} N$ then $M_\lambda \overset{+}{\to}_{\beta_P} N_\lambda$ (Prop. 3.3.18).
- For every λP-term M: $(M_{CL})_\lambda \twoheadrightarrow_{\beta_P} M$ (Prop. 3.3.20).
- For all CL_P-terms M, N: if $M \equiv_{W_P} N$ then $M_\lambda \equiv_{\beta_P} N_\lambda$ (Lem. 3.3.22).
- The $-_{CL}$ translation preserves higher-order unification and matching (Prop. 3.3.23).
- For all λP-terms M, N, if $M \equiv_{\beta_P} N$ then $M_{CL} \equiv_{W_P} N_{CL}$. (Cor. 3.3.24).

In order to prove *Abstraction Simulation* (Corollary 3.3.15), we need the following two lemmas:

1. $(M^\sigma)_{CL} = (M_{CL})^{\sigma_{CL}}$ for every λP-term M.
2. $(\lambda^* P.M)P^\sigma \twoheadrightarrow_{W_P} M^\sigma$ if $\mathsf{dom}(\sigma) \subseteq \mathsf{FV}(P)$.

The first of this lemmas has already been proved as Lemma 3.3.9. We now show the proof of the second lemma.

Lemma 3.3.14 $(\lambda^* P.M)P^\sigma \twoheadrightarrow_{W_P} M^\sigma$ if $\mathsf{dom}(\sigma) \subseteq \mathsf{FV}(P)$.

Proof.- By induction on the definition of $\lambda^* P.M$.

1. $\lambda^* P.M = \lambda^* x.x : (\lambda^* x.x)x^\sigma = SKKx^\sigma \to_{W_P} Kx^\sigma(Kx^\sigma) \to_{W_P} x^\sigma$.
2. $\mathsf{FV}(P) \cap \mathsf{FV}(M) = \emptyset : (\lambda^* P.M)P^\sigma = K_P M P^\sigma \to_{W_P} M = M^\sigma$. Given that $\mathsf{FV}(P) \cap \mathsf{FV}(M) = \emptyset$, then $\mathsf{dom}(\sigma) \cap \mathsf{FV}(M) = \emptyset$.
3. $\lambda^* P.M = \lambda^* QR.x$ with $x \in \mathsf{FV}(Q) \backslash \mathsf{FV}(R)$:

$$\begin{array}{ll}
(\lambda^* QR.x)QR^\sigma & = \\
S_{QR}(K\lambda^* Q.x)\Pi^1_{QR}(Q^\sigma R^\sigma) & \to_{W_P} \\
K(\lambda^* Q.x)(Q^\sigma R^\sigma)\Pi^1_{QR}(Q^\sigma R^\sigma) & \to_{W_P} \\
(\lambda^* Q.x)\Pi^1_{QR}(Q^\sigma R^\sigma) & \to_{W_P} \\
(\lambda^* Q.x)Q^\sigma & \twoheadrightarrow_{W_P(IH)} \\
x^\sigma. &
\end{array}$$

4. $\lambda^* P.M = \lambda^* QR.x$ with $x \in \mathsf{FV}(R)$: this is analogous to the previous case.
5. $\lambda^* P.M = \lambda^* P.NT$ with $\mathsf{FV}(P) \cap \mathsf{FV}(NT) \neq \emptyset$:

$$\begin{array}{ll} (\lambda^* P.NT)P^\sigma & = \\ S_P(\lambda^* P.N)(\lambda^* P.T)P^\sigma & \to_{W_P} \\ (\lambda^* P.N)P^\sigma((\lambda^* P.T)P^\sigma) & \twoheadrightarrow_{W_P(IH)} \\ N^\sigma((\lambda^* P.T)P^\sigma) & \twoheadrightarrow_{W_P(IH)} \\ N^\sigma T^\sigma & = \\ (NT)^\sigma. & \end{array}$$

Rule 6 in the definition of λ^* is a shortcut for a subset of the terms that would fit rule 5, but it can also be used to simulate the β_P-reduction (in 0 steps). The proof is simple:

6. $\lambda^* P.M = \lambda^* x.Mx$ with $x \notin \mathsf{FV}(M)$: $(\lambda^* x.Mx)x^\sigma = Mx^\sigma = M^\sigma x^\sigma = (Mx)^\sigma$, since $\mathsf{dom}(\sigma) = \{x\}$ and $x \notin \mathsf{FV}(M)$.

□

Corollary 3.3.15 (Abstraction Simulation) For every CL_P-representable term M, CL_P-representable pattern P and λP substitution σ s.t. $\mathsf{dom}(\sigma) \subseteq \mathsf{FV}(P)$: $((\lambda P.M)P^\sigma)_{CL} \twoheadrightarrow_{W_P} (M^\sigma)_{CL}$.

Proof.- By Lemma 3.3.9, $((\lambda P.M)P^\sigma)_{CL} = (\lambda P.M)_{CL}(P_{CL})^{\sigma_{CL}} = (\lambda^* P_{CL}.M_{CL})(P_{CL})^{\sigma_{CL}}$, and $(M^\sigma)_{CL} = (M_{CL})^{\sigma_{CL}}$. The result holds by Lemma 3.3.14. □

With this, we have finally proved that $((\lambda P.M)P^\sigma)_{CL} \twoheadrightarrow_{W_P} (M^\sigma)_{CL}$ whenever both sides of the $\twoheadrightarrow_{W_P}$ are well-formed CL_P-terms and the domain of σ is contained in FV(P). This result is an extension of its counterpart in CL: not only does $-_{CL}$ translate abstraction to λ^*, but it also applies the translation to the pattern.

Note that, just as in CL, the implication $M \to_{\beta_P} N \supset M_{CL} \twoheadrightarrow_{W_P} N_{CL}$ does not hold.

Other relevant results regarding the translations are listed below.

Lemma 3.3.16 $(M^\sigma)_\lambda = (M_\lambda)^{\sigma_\lambda}$.

Proof.- By induction on M.

- If $M = x$ with $x \in \mathcal{X}$, then $(M^\sigma)_\lambda = (x^\sigma)_\lambda = (\sigma(x))_\lambda = \sigma_\lambda(x) = x^{\sigma_\lambda} = x_\lambda^{\sigma_\lambda} = (M_\lambda)^{\sigma_\lambda}$.
- If M is a combinator or constructor, then M_λ is a ground term (by definition of the $-_\lambda$ translation, all combinators translate to terms with no free variables, and constructors translate to themselves). Therefore $(M^\sigma)_\lambda = M_\lambda = (M_\lambda)^{\sigma_\lambda}$.
- If $M = M_1 M_2$, then

$$
\begin{array}{ll}
(M^\sigma)_\lambda & = \\
((M_1 M_2)^\sigma)_\lambda & = \\
(M_1^\sigma M_2^\sigma)_\lambda & = \\
(M_1^\sigma)_\lambda (M_2^\sigma)_\lambda & =_{\text{IH}} \\
((M_1)_\lambda)^{\sigma_\lambda} ((M_2)_\lambda)^{\sigma_\lambda} & = \\
((M_1)_\lambda (M_2)_\lambda)^{\sigma_\lambda} & = \\
((M_1 M_2)_\lambda)^{\sigma_\lambda} & = \\
(M_\lambda)^{\sigma_\lambda}. &
\end{array}
$$

□

Lemma 3.3.17 $(\sigma \circ \sigma')_\lambda = \sigma_\lambda \circ \sigma'_\lambda$.

Proof.- It is enough to prove that this holds whenever the compositions of the substitutions are applied to free variables (since only free variables will be affected). Whether the variable is contained in the domain of σ' or not, it can be easily proved that the result of applying either side of the equation to the variable is the same. □

Reduction within CL_P functions as a weak reduction with respect to $\rightarrow_{\beta_P}$ (See section 2.4 for the original CL-analogue).

Proposition 3.3.18 (Weak Reduction) If $M \rightarrow_{W_P} N$ then $M_\lambda \overset{+}{\rightarrow}_{\beta_P} N_\lambda$

Proof.- By induction on the context of the reduction $M \rightarrow_{W_P} N$. In every case we make use of the result of Lemma 3.3.16 and the variable convention (that is, CL_P-patterns have no variables in common with other terms). Also keep in mind that the bound variables x and y introduced by the $-_\lambda$ translation are fresh variables, and that this translation does *not* introduce free variables that were not present in the original term.

We also assume that the domains of the substitutions are restricted to the variables of the terms to which they are originally applied. For instance, if we introduce a term as P^σ, that σ will be such that $\mathsf{dom}(\sigma) \subseteq \mathsf{FV}(P)$. This is a safe assumption, since for every substitution σ' it holds that $P^{\sigma'} = P^{\sigma'|_{\mathsf{FV}(P)}}$.

- $M = K_P M' P^\sigma$ and $N = M'$:

$$
\begin{array}{ll}
M_\lambda & = \\
(K_P)_\lambda M'_\lambda P^\sigma_\lambda & = \\
(\lambda x.\lambda P_\lambda.x) M'_\lambda (P_\lambda)^{\sigma_\lambda} & \rightarrow_{\beta_P} \\
(\lambda P_\lambda.M'_\lambda)(P_\lambda)^{\sigma_\lambda} \rightarrow_{\beta_P} (M'_\lambda)^{\sigma_\lambda} & = \\
M'_\lambda & = \\
N_\lambda. &
\end{array}
$$

  Here we are assuming that the substitution σ_λ does not affect the free variables in M'_λ. We can safely assume this, since M' has no variables in common with P, and thus M'_λ shares no free variables with P_λ; and also because the translation preserves the domain of the substitution: $dom(\sigma_\lambda) = \mathsf{dom}(\sigma)$. Similar considerations will be applied to the following cases.

- $M = S_P M' M'' P^\sigma$ and $N = M' P^\sigma (M'' P^\sigma)$:

$$\begin{array}{ll} M_\lambda & = \\ (S_P)_\lambda M'_\lambda M''_\lambda (P^\sigma)_\lambda & = \\ (\lambda x.\lambda y.\lambda P_\lambda.xP_\lambda(yP_\lambda))M'_\lambda M''_\lambda (P_\lambda)^{\sigma_\lambda} & \to_{\beta_P} \\ (\lambda y.\lambda P_\lambda.M'_\lambda P_\lambda(yP_\lambda))M''_\lambda (P_\lambda)^{\sigma_\lambda} & \to_{\beta_P} \\ (\lambda P_\lambda.M'_\lambda P_\lambda(M''_\lambda P_\lambda))(P_\lambda)^{\sigma_\lambda} & \to_{\beta_P} \\ M'_\lambda (P_\lambda)^{\sigma_\lambda}(M''_\lambda (P_\lambda)^{\sigma_\lambda}) & = \\ M'_\lambda (P^\sigma)_\lambda (M''_\lambda (P^\sigma)_\lambda) & = \\ N_\lambda. & \end{array}$$

- $M = \Pi^1_{PQ}(PQ)^\sigma$ and $N = P^\sigma$:

$$\begin{array}{ll} M_\lambda & = \\ (\Pi^1_{PQ})_\lambda((PQ)^\sigma)_\lambda & = \\ (\lambda P_\lambda Q_\lambda.P_\lambda)((PQ)_\lambda)^{\sigma_\lambda} & \to_{\beta_P} \\ (P_\lambda)^{\sigma_\lambda} & = \\ (P^\sigma)_\lambda & = \\ N_\lambda. & \end{array}$$

- $M = \Pi^2_{PQ}(PQ)^\sigma$ and $N = Q^\sigma$. This case is analogous to the previous one.
- $M = M'M''$, $M' \to_{W_P} M'''$ and $N = M'''M''$: $M_\lambda = M'_\lambda M''_\lambda \underset{IH}{\overset{+}{\to}}_{\beta_P} M'''_\lambda M''_\lambda = N_\lambda$.
- $M = M'M''$, $M'' \to_{W_P} M'''$ and $N = M''M'''$. This case is analogous to the previous one.

□

Corollary 3.3.19 If $M_\lambda \in \mathrm{SN}_{\lambda P}$ then $M \in \mathrm{SN}_{CL_P}$.

Proof.- This is an immediate consequence of Proposition 3.3.18, as an infinite derivation in CL_P starting from M would result in an infinite derivation in λP starting from M_λ. □

Proposition 3.3.20 $(M_{CL})_\lambda \twoheadrightarrow_{\beta_P} M$.

Proof.- By induction on M.

- If $M = x \in \mathcal{X}$, then $(M_{CL})_\lambda = (x_{CL})_\lambda = x = M$. Trivially, $M \twoheadrightarrow_{\beta_P} M$, as $\twoheadrightarrow_{\beta_P}$ is reflexive.
- If $M = M'M''$, then $(M_{CL})_\lambda = (M'M''_{CL})_\lambda = (M'_{CL}M''_{CL})_\lambda = (M'_{CL})_\lambda(M''_{CL})_\lambda$. By induction hypothesis, $(M'_{CL})_\lambda \twoheadrightarrow_{\beta_P} M'$ and $(M''_{CL})_\lambda \twoheadrightarrow_{\beta_P} M''$. Thus, $(M'_{CL})_\lambda(M''_{CL})_\lambda \twoheadrightarrow_{\beta_P} M'M'' = M$.
- If $M = \lambda P.M'$, then $M_{CL} = \lambda^* P_{CL}.M'_{CL}$. We will analyze all the possible cases.
  1. $M = \lambda x.x$: $M_{CL} = \lambda^* x.x = SKK$, and then

$$\begin{array}{ll} (M_{CL})_\lambda & = \\ (SKK)_\lambda & = \\ (\lambda z.\lambda y.\lambda x.zx(yx))(\lambda v.\lambda w.v)(\lambda t.\lambda u.t) & \twoheadrightarrow_{\beta_P} \\ \lambda x.(\lambda v.\lambda w.v)x((\lambda t.\lambda u.t)x) & \twoheadrightarrow_{\beta_P} \\ \lambda x.(\lambda w.x)(\lambda u.x) & \twoheadrightarrow_{\beta_P} \\ \lambda x.x & = M. \end{array}$$

2. $\mathsf{FV}(P) \cap \mathsf{FV}(M') = \emptyset$: $M_{CL} = \lambda^* P_{CL}.M'_{CL} = K_{P_{CL}} M'_{CL}$ and

$$\begin{array}{ll} (M_{CL})_\lambda & = \\ (K_{P_{CL}} M'_{CL})_\lambda & = \\ (\lambda x.\lambda (P_{CL})_\lambda.x)(M'_{CL})_\lambda & \twoheadrightarrow_{\beta_P} \\ & \quad IH \\ (\lambda x.\lambda P.x)M' & \twoheadrightarrow_{\beta_P} \\ \lambda P.M' & = \ M. \end{array}$$

3. $M' = x \in \mathsf{FV}(P)$. In this case, since P cannot be x (otherwise we would be on the first case), nor a different variable (since $x \in \mathsf{FV}(P)$), then P must translate to an application in CL_P. This is because, outside of variables, only applications may contain free variables in CL_P. Therefore $P_{CL} = RQ$ for some CL_P-terms R and Q (if P satisfies RPC^+, then RQ will be a CL_P-pattern). There are 2 possibilities:
   - $x \in \mathsf{FV}(R) \backslash \mathsf{FV}(Q)$: $M_{CL} = \lambda^* RQ.x = S_{RQ}(K\lambda^* R.x)\Pi^1_{RQ}$ and

     $$\begin{array}{ll} (M_{CL})_\lambda & = \\ (\lambda w.\lambda y.\lambda (RQ)_\lambda.w(RQ)_\lambda(y(RQ)_\lambda))((\lambda v.\lambda u.v)(\lambda^* R.x)_\lambda)(\lambda (RQ)_\lambda.R_\lambda) & \twoheadrightarrow_{\beta_P} \\ \lambda (RQ)_\lambda.(\lambda v.\lambda u.v)(\lambda^* R.x)_\lambda (RQ)_\lambda((\lambda (RQ)_\lambda.R_\lambda)(RQ)_\lambda) & \twoheadrightarrow_{\beta_P} \\ \lambda (RQ)_\lambda.((\lambda^* R.x)_\lambda)R_\lambda & = \\ \lambda (RQ)_\lambda.((\lambda^* R.x)R)_\lambda. & \end{array}$$

     We know by Lemma 3.3.14 that $(\lambda^* R.x)R \twoheadrightarrow_{W_P} x$. Using Proposition 3.3.18, we can conclude that $((\lambda^* R.x)R)_\lambda \twoheadrightarrow_{\beta_P} x_\lambda = x$.
     Therefore $\lambda (RQ)_\lambda.((\lambda^* R.x)R)_\lambda \twoheadrightarrow_{\beta_P} \lambda (RQ)_\lambda.x = \lambda (P_{CL})_\lambda.x \twoheadrightarrow_{\beta_P} \lambda P.x = M$.
     IH
   - If $x \in \mathsf{FV}(Q)$, then $M_{CL} = \lambda^* RQ.x = S_{RQ}(K\lambda^* Q.x)\Pi^2_{RQ}$. This case is analogous to the previous one.
4. M does not fit into any of the previous cases. This means that M' translates into an application which shares at least one free variable with P. Then M_{CL} is of the form $\lambda^* P_{CL}.LN = S_{P_{CL}}(\lambda^* P_{CL}.L)(\lambda^* P_{CL}.N)$. So:

   $$\begin{array}{ll} (M_{CL})_\lambda & = \\ (\lambda x.\lambda y.\lambda (P_{CL})_\lambda.x(P_{CL})_\lambda(y(P_{CL})_\lambda))(\lambda^* P_{CL}.L)_\lambda(\lambda^* P_{CL}.N)_\lambda & \twoheadrightarrow_{\beta_P} \\ \lambda (P_{CL})_\lambda.(\lambda^* P_{CL}.L)_\lambda (P_{CL})_\lambda(((\lambda^* P_{CL}.N)_\lambda)(P_{CL})_\lambda) & = \\ \lambda (P_{CL})_\lambda.((\lambda^* P_{CL}.L)(P_{CL}))_\lambda(((\lambda^* P_{CL}.N)P_{CL})_\lambda) & \twoheadrightarrow_{\beta_P} \\ & \quad L.\ 3.3.14,\ P.\ 3.3.18 \\ \lambda (P_{CL})_\lambda.L_\lambda N_\lambda & = \\ \lambda (P_{CL})_\lambda.(LN)_\lambda & = \\ \lambda (P_{CL})_\lambda.(M'_{CL})_\lambda & \twoheadrightarrow_{\beta_P} \\ & \quad IH \\ \lambda P.M' & = \ M. \end{array}$$

□

Remark 3.3.21 The above proposition holds without the inclusion of accelerator rules. If we introduce rule 6, the $\twoheadrightarrow_{\beta_P}$ relation must be replaced by $\equiv_{\beta_P \eta}$ since, for example, $((\lambda x.yx)_{CL})_\lambda$ will become y, which is η-equivalent but not β_P-equivalent to $\lambda x.yx$.

While the translations we presented above can simulate abstractions correctly, there is still room for optimization. An alternate definition can be used in order to improve the efficiency

(and, at the same time, reduce the size of the translated terms by decreasing the number of subindices) by avoiding the replication of subindices in each recursive call. See Definition B.1.1 in Appendix B.1 for more details on this optimization.

Lemma 3.3.22 If $M \equiv_{W_P} N$ then $M_\lambda \equiv_{\beta_P} N_\lambda$.

Proof.- By confluence, there exists a term L such that $M \twoheadrightarrow_{W_P} L$ and $N \twoheadrightarrow_{W_P} L$. Then, by Proposition 3.3.18, $M_\lambda \twoheadrightarrow_{\beta_P} L_\lambda$ and $N_\lambda \twoheadrightarrow_{\beta_P} L_\lambda$. □

Proposition 3.3.23

1. If σ unifies the equation $M_{CL} \equiv_{W_P} N_{CL}$, then σ_λ unifies the equation $M \equiv_{\beta_P} N$.
2. If σ is idempotent, then so is σ_λ.
3. If $M_{CL} \equiv_{W_P} N^\sigma_{CL}$, then $M \equiv_{\beta_P} N^{\sigma_\lambda}$ (higher-order matching is preserved).

Proof.- We prove items 1 and 2, 3 being similar to 1.

1. We know that $(M_{CL})^\sigma \equiv_{W_P} (N_{CL})^\sigma$. Then, by Lemmas 3.3.16 and 3.3.22,

$$\begin{array}{ll} ((M_{CL})_\lambda)^{\sigma_\lambda} & = \\ ((M_{CL})^\sigma)_\lambda & \equiv_{\beta_P} \\ ((N_{CL})^\sigma)_\lambda & = \\ ((N_{CL})_\lambda)^{\sigma_\lambda}. & \end{array}$$

   By Proposition 3.3.20, $(M_{CL})_\lambda \twoheadrightarrow_{\beta_P} M$ and $(N_{CL})_\lambda \twoheadrightarrow_{\beta_P} N$. This means that $((M_{CL})_\lambda)^{\sigma_\lambda} \twoheadrightarrow_{\beta_P} M^{\sigma_\lambda}$ and $((N_{CL})_\lambda)^{\sigma_\lambda} \twoheadrightarrow_{\beta_P} N^{\sigma_\lambda}$. Therefore $M^{\sigma_\lambda} \equiv_{\beta_P} N^{\sigma_\lambda}$.
2. Since $\sigma \circ \sigma = \sigma$, then by Lemma 3.3.17: $\sigma_\lambda \circ \sigma_\lambda = \sigma_\lambda$.

□

Corollary 3.3.24 If $M \equiv_{\beta_P} N$ then $M_{CL} \equiv_{W_P} N_{CL}$.

Proof.- By item 3 of Proposition 3.3.23, using the identity substitution as σ. □

3.4 The CL_P-Theory

Recall that the λ-calculus as well as CL can be formulated as equational theories (see for example [Bar84, Bar92, BN98, BKdV03] for details about equational theories). The theory associated to CL_P is defined as interpreting all rewriting rules by an infinite set of equalities.

$$\begin{array}{lcll} K_P x P & \doteq & x, & x \notin \mathsf{FV}(P) \\ S_P x y P & \doteq & xP(yP), & x, y \notin \mathsf{FV}(P) \\ \Pi^1_{PQ}(PQ) & \doteq & P & \\ \Pi^2_{PQ}(PQ) & \doteq & Q & \end{array}$$

where P and Q range over CL_P-patterns, and the application PQ – where used – is also a pattern.

With this interpretation, the terms M and N will be considered equal whenever $M \equiv_{W_P} N$. The CL_P-theorems are the provable equalities which follow from the above clauses, closed under applicative congruence and substitution.

Definition 3.4.1 A theory $\mathcal{T}_1$ is a **proper consistent extension** of a theory $\mathcal{T}_2$ if:

- $\mathcal{T}_1$ is consistent,
- all $\mathcal{T}_2$-theorems are also $\mathcal{T}_1$-theorems, and
- some $\mathcal{T}_1$-theorems are not $\mathcal{T}_2$-theorems.

Remark 3.4.2 Restricting the language to any family of patterns satisfying RPC^{++}, the equational CL_P-theory is a proper consistent extension of the equational CL-theory.

Being a proper extension is straightforward. Consistency follows by taking any two different normal forms (such as x and y), then by confluence they will not be convertible.

3.4.1 Extensional CL_P

There is the question of whether the CL_P-theory is maximally consistent or not (sometimes known as Post consistent or saturated)[5]. We will see that, as in CL, the answer is again negative. One way to extend the theory is by adding extensionality [Bar84, Bar92].

Let us consider incorporating the following rewriting rule to the original formulation of W_P, to be called η (in which every pattern is a variable):

$$S(Kx)(SKK) \quad \to \quad x$$

It is clear that, unlike its counterpart in the λ-calculus, the above rule does not require a specific variable not to be free in M: this has already been taken care of in the translation $\lambda^* x.Mx = M$ (see the comment after translation rule 6 in subsection 3.3.1).

Remark 3.4.3 This extension can simulate the traditional η-rule. That is, $(\lambda x.Mx)_{CL} \twoheadrightarrow_{W_P+\eta} M_{CL}$.

We can see that if we use rule 6 of λ^*, then $(\lambda x.Mx)_{CL} = \lambda^* x.M_{CL}x = M_{CL}$, which reduces to M_{CL} in 0 steps. On the other hand, if we do not allow rule 6, then $(\lambda x.Mx)_{CL} = \lambda^* x.M_{CL}x = S(KM_{CL})(SKK)$, which reduces to M_{CL} in 1 step of the η-rule.

Note that our definition of RPC^{++} needs to be updated so that the clause that states patterns must not unify with the left-hand-side of a W_P-rule reads "$W_P + \eta$" rather than just W_P. Otherwise, orthogonality would not be valid anymore, as there would be an infinite number of critical pairs:

$$\Pi^j_{SPQ}(S(KM)(SKK)) \quad j = 1, 2$$

[5] That is, without adding specific reduction rules involving new combinators, which is always a possibility, but it is not the focus of this section.

where P, Q are patterns such that, for some substitution σ, $P^\sigma = KM$ and $Q^\sigma = SKK$. In all these terms, both η and projection are applicable and it is not possible to close the diagram. Thus this system would not even be weakly confluent.

The problem originates from the fact that a pattern like Syz above has instances which may η-reduce in the root (because $S(Kx)(SKK)$ may match with it). Therefore, in order to have a sound extensionality in the calculus, a new restriction should be added to RPC^{++}. We discuss such a condition below.

We will say that the pattern P is **η-forbidden** if there is an application QR such that QR is a subterm of P and $QR \doteq S(Kx)(SKK)$. That is, if any subterm of P is an application which unifies with the left-hand-side of the η rule. Let us call this new restriction for the set of patterns RPC^{++}_η.

Remark 3.4.4 If a set Φ of patterns satisfies RPC^{++}, then the set $\{P \in \Phi \mid P$ is not η-forbidden$\}$ satisfies RPC^{++}_η.

Let us write CL_η for $CL{+}\eta$ extending the original formulation, and $CL_{P\eta}(\Phi)$ for the rewriting system consisting of $CL_P(\Phi) + \eta$, i.e. in which, as usual, all patterns belong to Φ.

Proposition 3.4.5 For any set Φ of patterns satisfying RPC^{++}_η, the rewriting system $CL_{P\eta}(\Phi)$ is confluent.

Proof.- The resulting system is left-linear and does not have critical pairs, hence it is orthogonal. □

The $CL_{P\eta}$-theory extends CL_P with extensionality, in which the η-rule is interpreted as an equality. Then we have, as before:

Corollary 3.4.6 Restricting the language to any family of patterns satisfying RPC^{++}_η, the equational theory $CL_{P\eta}(\Phi)$ is a proper consistent extension of both the CL_η theory and the $CL_P(\Phi)$ theory.

Proof.-

- Proper: $S(K\Pi^1_{SK})(SKK) \doteq \Pi^1_{SK}$ is a theorem in $CL_{P\eta}(\Phi)$ but not in CL_η nor $CL_P(\Phi)$.
- Consistent: S and K are two different normal forms with respect to $CL_{P\eta}(\Phi)$.
- Extension: the set of axioms $CL_{P\eta}(\Phi)$ contains the axioms of both CL_η and $CL_P(\Phi)$.

□

3.5 A Simple Type System for CL_P

We now describe a type system for CL_P, named $CL_P^{\rightarrow}$, and prove some of its salient properties. **Type expressions** are defined by following grammar:

$$A, B \quad ::= \quad \alpha \mid A \rightarrow B$$

$$\frac{x^A \in \Gamma}{\Gamma \vdash x\colon A}\,(\textsc{Var}) \qquad \frac{\Gamma' \vdash P\colon A}{\Gamma \vdash K_P\colon B \to A \to B}\,(\mathrm{K})$$

$$\frac{\Gamma' \vdash P\colon A}{\Gamma \vdash S_P\colon (A \to B \to C) \to (A \to B) \to A \to C}\,(\mathrm{S})$$

$$\frac{\Gamma' \vdash P\colon A \to B \quad \Gamma' \vdash Q\colon A}{\Gamma \vdash \Pi^1_{PQ}\colon B \to A \to B}\,(\textsc{Pi-1}) \qquad \frac{\Gamma' \vdash P\colon A \to B \quad \Gamma' \vdash Q\colon A}{\Gamma \vdash \Pi^2_{PQ}\colon B \to A}\,(\textsc{Pi-2})$$

$$\frac{\Gamma \vdash M\colon A \to B \quad \Gamma \vdash N\colon A}{\Gamma \vdash MN\colon B}\,(\to\text{-}\textsc{Elim})$$

Figure 3.1: Typing rules for CL_P

where α ranges over a non-empty set of type variables $\mathcal{A}$. As usual, arrows in types are right-associative.

We will work with the following definitions regarding contexts. A **context** is a finite set of variables with type decorations $\{x_1^{A_1}, \ldots, x_n^{A_n}\}$, with $x_i \neq x_j$ for $i \neq j$ (this is analogous to the definition of typing contexts for $\lambda^{\to}$ shown in Section 2.3.2). The **domain** of Γ, denoted $\mathsf{dom}(\Gamma)$, is defined as $\{x \mid \exists A(x^A \in \Gamma)\}$. Its **range**, denoted $\mathsf{ran}(\Gamma)$, is defined as $\{A \mid \exists x(x^A \in \Gamma)\}$. The intersection between two contexts $(\Gamma \cap \Gamma')$ is defined as: $\{x^A \mid x^A \in \Gamma \wedge x^A \in \Gamma'\}$. Similarly, the union of two contexts $(\Gamma \cup \Gamma')$ is defined as: $\{x^A \mid x^A \in \Gamma \vee x^A \in \Gamma'\}$, and is only defined if for every x, A and B: $x^A \in \Gamma \wedge x^B \in \Gamma' \supset A = B$. A substitution σ is **compatible** with a context Γ if for all types A, B and every $x \in \mathsf{dom}(\sigma)$: $(x^A \in \Gamma$ and $\Gamma \vdash x^\sigma\colon B \to A = B)$.

We also work with **type substitutions**, total functions from type variables to types. If $\delta(\alpha) = A$, then we often write $\alpha \leftarrow A \in \delta$. The domain of δ is $\{\alpha \mid \delta(\alpha) \neq \alpha\}$. Substitution over types (A^δ) and contexts is defined as:

$$\begin{array}{rcl} \alpha^\delta & \triangleq & \delta(\alpha) \\ (A \to B)^\delta & \triangleq & A^\delta \to B^\delta \\ \Gamma^\delta & \triangleq & \{x^{A^\delta} \mid x^A \in \Gamma\} \end{array}$$

The **typing rules** for $CL_P^{\to}$ are given in Fig. 3.1. Note that the patterns themselves must be typable for the term that contains them to have a type. We do not require them to be typable in the same context as said term, but there has to be some context Γ' in which the pattern can be assigned the expected type. That is, the type that will be expected of the arguments which shall match it. This is because the pattern may have variables that will disappear once a substitution is applied. For example, the term $K_{SKx}S(SKK)$ is a ground term, which is typable in the empty context, in which its subterm SKK has type $A \to A$. However, the pattern SKx is not typable in the empty context because typing information is needed for the variable x. Hence the need for the context Γ', which assigns the required types to the variables in the pattern (an easy way to obtain one such Γ' is to extend Γ with the variables in the patterns, each with the type of the term that will match with it). Since patterns are linear and all their variables are

fresh by convention, no conflicts may arise from extending the contexts in this manner.

Remark 3.5.1 This type system is conservative with respect to the traditional one for simply typed CL: terms which are untypable in CL are also untypable in CL_P, and terms which have a type in CL are assigned the same type in CL_P. This follows from the observation that restricting the rules to terms with no projectors and only variables as their patterns, results in the original type system (all statements of the form $\Gamma' \vdash P : A$ trivially hold for $\Gamma' = \{P^A\}$ when P is a variable). It is not, however, conservative with respect to type inhabitation, as mentioned above.

We now exhibit some examples.

Example 3.5.2 We can prove that $\emptyset \vdash S_P KK : A \to A$, for any given type A and pattern P such that P can be assigned type A in some context.

$$\dfrac{\dfrac{\dfrac{\Gamma' \vdash P : A}{\emptyset \vdash S_P : (A \to (C \to A) \to A) \to (A \to C \to A) \to A \to A}\,(\text{S}) \quad \dfrac{\dfrac{}{\{x_1^{C \to A}\} \vdash x_1 : C \to A}\,(\textsc{Var})}{\emptyset \vdash K_{x_1} : A \to (C \to A) \to A}\,(\text{K})}{\emptyset \vdash S_P K : (A \to C \to A) \to A \to A}\,(\to\text{-}\textsc{Elim}) \quad \dfrac{\dfrac{}{\{x_2^C\} \vdash x_2 : C}\,(\textsc{Var})}{\emptyset \vdash K_{x_2} : A \to C \to A}\,(\text{K})}{\emptyset \vdash S_P KK : A \to A}\,(\to\text{-}\textsc{Elim})$$

Example 3.5.3 The term $\Pi^2_{Sx} SK$ can be assigned the type $A \to B \to A$, as shown below. We split the derivation into three parts.
Part 1:

$$\dfrac{\dfrac{\dfrac{}{\{x_1^A\} \vdash x_1 : A}\,(\textsc{Var})}{\{x^{A \to B \to A}\} \vdash S_{x_1} : (A \to B \to A) \to (A \to B) \to A \to A}\,(\text{S}) \quad \dfrac{}{\{x^{A \to B \to A}\} \vdash x : A \to B \to A}\,(\textsc{Var})}{\emptyset \vdash \Pi^2_{Sx} : ((A \to B) \to A \to A) \to A \to B \to A}\,(\textsc{Pi-2})$$

Part 2:

$$\dfrac{\dfrac{\dfrac{}{\{x_2^A\} \vdash x_2 : A}\,(\textsc{Var})}{\{x^{A \to B \to A}\} \vdash S_{x_2} : (A \to B \to A) \to (A \to B) \to A \to A}\,(\text{S}) \quad \dfrac{\dfrac{}{\{x_3^B\} \vdash x_3 : B}\,(\textsc{Var})}{\emptyset \vdash K_{x_3} : A \to B \to A}\,(\text{K})}{\emptyset \vdash SK : (A \to B) \to A \to A}\,(\to\text{-}\textsc{Elim})$$

Part 3:

$$\dfrac{\dfrac{\cdots}{\emptyset \vdash \Pi^2_{Sx} : ((A \to B) \to A \to A) \to A \to B \to A}\,(\textsc{Pi-2}) \quad \dfrac{\cdots}{\emptyset \vdash SK : (A \to B) \to A \to A}\,(\to\text{-}\textsc{Elim})}{\emptyset \vdash \Pi^2_{Sx}(SK) : A \to B \to A}\,(\to\text{-}\textsc{Elim})$$

Note that these typing rules allow us to prove that $\{x^A\} \vdash K_K xK : A$. However, the term $K_K xS$ is untypable: it cannot be assigned a type in any context. This is because K_K can only be assigned types of the form $C \to (A \to B \to A) \to C$ for some types A, B and C, while S cannot have a type of the form $A \to B \to A$ for any types A and B. This is an example of how (mismatched) patterns may affect the typability of a term: if we removed the subscripts, the term KxS would, of course, be typable.

Remark 3.5.4 Note that every type is inhabited (for example, the term $\Pi^2_{Kx}(SKK)$ has any given type in the empty context). While this result makes it impossible to establish a Curry-Howard isomorphism with a consistent logic, it does not invalidate the use of this $CL^{\rightarrow}_{P}$ for typing purposes, as it still satisfies important properties like Subterm Typability, Weakening and Strengthening, Type Substitution, Type Preservation and even Strong Normalization of typable terms.

3.5.1 Main Results

We now show some properties of $CL^{\rightarrow}_{P}$. The first is the Inversion Lemma whose proof is immediate from the syntax-driven nature of the typing rules.

Lemma 3.5.5 (Inversion Lemma) If $\Gamma \vdash M : D$ is derivable, then:

- if $M = x \in \mathcal{X}$ then $x^D \in \Gamma$.
- if $M = K_P$ for some pattern P then there exist A, B, Γ' s.t. $D = A \rightarrow B \rightarrow A$ and $\Gamma' \vdash P : B$.
- if $M = S_P$ for some pattern P then there exist A, B, C, Γ' s.t. $D = (A \rightarrow B \rightarrow C) \rightarrow (A \rightarrow B) \rightarrow A \rightarrow C$ and $\Gamma' \vdash P : A$.
- if $M = \Pi^1_{PQ}$ for some application pattern PQ then there exist A, B, Γ' s.t. $D = B \rightarrow A \rightarrow B$, $\Gamma' \vdash P : A \rightarrow B$ and $\Gamma' \vdash Q : A$.
- if $M = \Pi^2_{PQ}$ for some application pattern PQ then there exist A, B, Γ' s.t. $D = B \rightarrow A$, $\Gamma' \vdash P : A \rightarrow B$ and $\Gamma' \vdash Q : A$.
- if $M = M_1 M_2$ for some terms M_1 and M_2 then there exists A s.t. $\Gamma \vdash M_1 : A \rightarrow D$ and $\Gamma \vdash M_2 : A$.

Corollary 3.5.6 If a term is typable in an context Γ, so are all its subterms.

Proof.- By structural induction on the original term, using the last item of the Inversion Lemma when the term is an application. □

Corollary 3.5.7 If $\Gamma \vdash M : A$ then $\mathsf{FV}(M) \subseteq \mathsf{dom}(\Gamma)$.

Proof.- By structural induction on M. □

The following lemma will allow for weakening and strengthening of type derivations, which will be needed in order to prove the Term Substitution property.

Lemma 3.5.8 (Weakening and Strengthening) If $\Gamma \vdash M : A$ and $\mathsf{FV}(M) \subseteq \mathsf{dom}(\Gamma \cap \Gamma')$, then $\Gamma' \vdash M : A$.

Proof.- By structural induction on M, using the Inversion Lemma (IL).

- $M = x \in \mathcal{X}$. By IL, $x^A \in \Gamma$ and $\mathsf{FV}(x) = \{x\}$. Then $x \in \mathsf{dom}(\Gamma \cap \Gamma')$ and $x^A \in \Gamma$, and thus $x^A \in \Gamma'$, and we obtain $\Gamma' \vdash x\colon A$ by Var.
- $M \in \{K_P, S_P, \Pi^1_{QR}, \Pi^2_{QR}\}$ for some patterns P, Q, R such that QR is a pattern. Then $\mathsf{FV}(M) = \emptyset$. It is immediate from the formulation of the rules K, S, Pi-1 and Pi-2 that if $\Gamma \vdash M\colon A$ is derivable for some context Γ, then this must also hold for any other context Γ' (the rules require nothing of the context Γ which is used to type M).
- $M = M_1 M_2$. By IL, there is a C s.t. $(\Gamma \vdash M_1 : C \to A$ and $\Gamma \vdash M_2 : C)$ are derivable, $\mathsf{FV}(M_1) \subseteq \mathsf{FV}(M) \subseteq \Gamma'$, and $\mathsf{FV}(M_2) \subseteq \mathsf{FV}(M) \subseteq \Gamma'$. By IH, we can derive $\Gamma' \vdash M_1\colon C \to A$ and $\Gamma' \vdash M_2\colon C$, and then obtain $\Gamma' \vdash M_1 M_2\colon A$ by ($\to$-Elim).

□

Corollary 3.5.9 If $\Gamma \vdash M : A$ and $\Gamma \subseteq \Gamma'$, then $\Gamma' \vdash M : A$.

Proof.- We know by Corollary 3.5.7 that if $\Gamma \vdash M : A$, then $\mathsf{FV}(M) \subseteq \mathsf{dom}(\Gamma)$. Also, because $\Gamma \subseteq \Gamma'$, $\Gamma \cap \Gamma' = \Gamma$ thus $\mathsf{dom}(\Gamma \cap \Gamma') = \mathsf{dom}(\Gamma)$. It is now immediate from Lemma 3.5.8 that $\Gamma' \vdash M\colon A$. □

Lemma 3.5.10 If $\mathsf{FV}(M) = \emptyset$ and $\Gamma \vdash M\colon A$, then $\Gamma' \vdash M\colon A$.

Proof.- If $\mathsf{FV}(M) = \emptyset$, then $\mathsf{FV}(M) \subseteq \mathsf{dom}(\Gamma \cap \Gamma')$. Then, by Lemma 3.5.8, if $\Gamma \vdash M\colon A$ then $\Gamma' \vdash M\colon A$. □

Lemma 3.5.11 If $\Gamma \vdash P^\sigma\colon A$ then there is a context Γ's.t. $\Gamma' \vdash P\colon A$.

Proof.- By structural induction on P.

- If $P = x$ with $x \in \mathcal{X}$, take $\Gamma' = \{x^A\}$
- If $P \in \{K_{P'}, S_{P'}, \Pi^1_{QR}, \Pi^2_{QR}\}$ for some patterns P', Q, R such that QR is a pattern, then $\mathsf{FV}(P) = \emptyset$ and $P^\sigma = P$. Take $\Gamma' = \Gamma$.
- If $P = P_1 P_2$, then $P^\sigma = P_1^\sigma P_2^\sigma$. By Inversion Lemma, $\Gamma \vdash P_1^\sigma : C \to A$ and $\Gamma \vdash P_2^\sigma : C$ for some type C. Then, by IH, $\exists \Gamma_1, \Gamma_2 (\Gamma_1 \vdash P_1 : C \to A$ and $\Gamma_2 \vdash P_2 : C)$. Take $\Gamma' = \Gamma_1 \cup \Gamma_2$. Γ' is well-defined, as P_1 and P_2 have no free variables in common (because P is a pattern and thus it is linear), which means that no variables need appear simultaneously in Γ_1 and Γ_2. Since $\Gamma_1 \subseteq \Gamma'$ and $\Gamma_2 \subseteq \Gamma'$, then by Corollary 3.5.9, $\Gamma' \vdash P_1 : C \to A$ and $\Gamma' \vdash P_2 : C$. This, by ($\to$-Elim), means that $\Gamma' \vdash P\colon A$.

□

The following result states that if M can be typed in Γ extended with the adequate types for each variable in the domain of σ, then M^σ can be typed in Γ with the same type.

Lemma 3.5.12 (Term Substitution) If σ is compatible with Γ and $\Gamma \cup \{x_i^{A_i} \mid \exists M_i (x_i \leftarrow M_i \in \sigma \wedge \Gamma \vdash M_i\colon A_i)\} \vdash M\colon A$, then $\Gamma \vdash M^\sigma\colon A$.

Proof.- Let $\Gamma_1 = \{x_i^{A_i} \mid \exists M_i(x_i \leftarrow M_i \in \sigma \wedge \Gamma \vdash M_i : A_i)\}$. We need to prove that if $\Gamma \cup \Gamma_1 \vdash M : A$ then $\Gamma \vdash M^\sigma : A$. By Lemma 3.5.8, this is equivalent to proving that if $\Gamma \cup \Gamma' \vdash M : A$, then $\Gamma \vdash M^\sigma : A$, where $\Gamma' = \{x_i^{A_i} \mid x_i \in \mathsf{FV}(M) \wedge \exists M_i(x_i \leftarrow M_i \in \sigma \wedge \Gamma \vdash M_i : A_i)\}$. That is, Γ' is Γ_1 restricted to the free variables of M. Note that the compatibility requirement over σ and Γ is necessary: otherwise $\Gamma \cup \Gamma_1$ would not be well-defined.

We will prove our new statement by induction on M.

- $M = x \in \mathcal{X}$:
  - If there is a term M_i s.t. $x \leftarrow M_i \in \sigma$, then $\Gamma' = \{x^{A_i}\}$ where $\Gamma \vdash x^\sigma : A_i$. Since $\Gamma \cup \Gamma' \vdash x : A$ by hypothesis, then by Inversion Lemma $x^A \in \Gamma \cup \Gamma'$. And since $x^{A_i} \in \Gamma'$, then $A_i = A$. Hence, since $\Gamma \vdash x^\sigma : A_i$, this means that $\Gamma \vdash x^\sigma : A$.
  - Otherwise, $x^\sigma = x$, $\Gamma' = \emptyset$ and $\Gamma \cup \Gamma' \vdash x : A$. Then we can prove $\Gamma \vdash x : A$, which is the same as $\Gamma \vdash x^\sigma : A$.
- $M \in \{K_P, S_P, \Pi^1_{QR}, \Pi^2_{QR}\}$ for some patterns P, Q, R and QR. Then $M^\sigma = M$ and $\mathsf{FV}(M) = \emptyset$. We know that $\Gamma \cup \Gamma' \vdash M : A$ by hypothesis. Then, by Lemma 3.5.10, M has type A in every context, particularly Γ'.
- $M = M_1 M_2$. By Inversion Lemma, $\Gamma \cup \Gamma' \vdash M_1 : C \to A$ and $\Gamma \cup \Gamma' \vdash M_2 : C$ for some type C. Then, by IH, $\Gamma \vdash M_1^\sigma : C \to A$ and $\Gamma \vdash M_2^\sigma : C$ (resorting to Lemma 3.5.8 to remove unwanted variables from Γ' before each application of the IH). Using the $\to$-Elim rule, we can conclude that $\Gamma \vdash M_1^\sigma M_2^\sigma : A$.

□

Note that Lemma 3.5.12 generalizes the Term Substitution lemma for *CL*, in which the domain of the substitution σ is only one variable.

Lemma 3.5.13 If $\Gamma \vdash M : A$ then $\Gamma^\delta \vdash M : A^\delta$.

Proof.- By induction on the derivation of $\Gamma \vdash M : A$.

- If $\Gamma \vdash x : A$ with $x^A \in \Gamma$, then $x : A^\Delta \in \Gamma^\Delta$, thus $\Gamma^\Delta \vdash x : A^\Delta$.
- If $\Gamma \vdash K_P : B \to D \to B = A$ for some types B, D where $\Gamma' \vdash P : D$ for some context Γ', then by IH, $\Gamma'^\Delta \vdash P : D^\Delta$. So, applying the K-rule, $\Gamma^\Delta \vdash K_P : B^\Delta \to D^\Delta \to B^\Delta = A^\Delta$.
- If $\Gamma \vdash S_P : (D \to B \to C) \to (D \to B) \to D \to C = A$ for some types B, D and C, with $\Gamma' \vdash P : D$ for some context Γ', then by IH: $\Gamma'^\Delta \vdash P' : D^\Delta$. Using the S-rule, we can conclude $\Gamma^\Delta \vdash S_P : (D^\Delta \to B^\Delta \to C^\Delta) \to (D^\Delta \to B^\Delta) \to D^\Delta \to C^\Delta = A^\Delta$.
- If $\Gamma \vdash \Pi^1_{QR} : B \to D \to B = A$, the argument is similar to the previous cases.
- If $\Gamma \vdash \Pi^2_{QR} : B \to D = A$, it is also similar to the previous cases.
- If $\Gamma \vdash MN : A$ with $\Gamma \vdash M : C \to A$ and $\Gamma \vdash N : C$ for some type C, then, by IH, $\Gamma^\Delta \vdash M : C^\Delta \to A^\Delta$ and $\Gamma^\Delta \vdash N : C^\Delta$. Using the $\to$-Elim-rule, we conclude that $\Gamma^\Delta \vdash MN : A^\Delta$.

□

Using the previous results, we have proved that our typing is preserved under reductions. In other words, our type system satisfies the following core property:

Proposition 3.5.14 (Type Preservation) If $M \to_{W_P} M'$ and $\Gamma \vdash M : A$ then $\Gamma \vdash M' : A$.

Proof.- By structural induction on M. Since we know that $M \to_{W_P} M'$, there are 6 possibilities:

1. $M = K_P M_1 P^\sigma$ and $M' = M_1$ for some pattern P, CL_P-term M_1 and substitution σ.
2. $M = S_P M_1 M_2 P^\sigma$ and $M' = M_1 P^\sigma (M_2 P^\sigma)$ for some pattern P, CL_P-terms M_1 and M_2, and substitution σ.
3. $M = \Pi^1_{PQ}(P^\sigma Q^\sigma)$ and $M' = P^\sigma$ for some patterns P, Q and PQ, and substitution σ.
4. $M = \Pi^2_{PQ}(P^\sigma Q^\sigma)$ and $M' = Q^\sigma$ for some patterns P, Q and PQ, and substitution σ.
5. $M = M_1 M_2$, $M' = M_1' M_2$ and $M_1 \to_{W_P} M_1'$ for some CL_P-terms M_1, M_2 and M_1'.
6. $M = M_1 M_2$, $M' = M_1 M_2'$ and $M_2 \to_{W_P} M_2'$ for some CL_P-terms M_1, M_2 and M_2'.

We analyze each of these cases separately, assuming that $\Gamma \vdash M : A$, to prove that $\Gamma \vdash M' : A$.

1. $M = K_P M_1 P^\sigma$ and $M' = M_1$.
   By Inversion Lemma, there is a C $\Gamma \vdash K_P M_1 : C \to A$ and $\Gamma \vdash P^\sigma : C$. Using the Inversion Lemma again, we know there is a B s.t. $\Gamma \vdash K_P : B \to C \to A$ and $\Gamma \vdash M_1 : B$. A third application of the Inversion Lemma reveals that B must be A, and also that P must be of type C in some context Γ' (which, by Lemma 3.5.11, is always possible, as $\Gamma \vdash P^\sigma : C$). Then we have: $\Gamma \vdash K_P : A \to C \to A$, $\Gamma \vdash M_1 : A$, $\Gamma \vdash P^\sigma : C$ and $\Gamma' \vdash P : C$. Since $M' = M_1$, we have already proved that $\Gamma \vdash M' : A$.
2. $M = S_P M_1 M_2 P^\sigma$ and $M' = M_1 P^\sigma (M_2 P^\sigma)$.
   By Inversion Lemma, there exist D and P_1s.t. $\Gamma \vdash S_P M_1 M_2 : D \to A$ and $\Gamma \vdash P^\sigma : D$. Using the lemma again shows that there is a Cs.t. $\Gamma \vdash S_P M_1 : C \to D \to A$ and $\Gamma \vdash M_2 : C$. Using it once more, we get that $\Gamma \vdash S_P : B \to C \to D \to A$ for some B, and $\Gamma \vdash M_1 : B$. With a final application of the Inversion Lemma, we obtain that $B = D \to B' \to A$ and $C = D \to B'$ for some type B'. Also, $\exists \Gamma' (\Gamma' \vdash P : D)$, but this is true by Lemma 3.5.11, since P^σ matches P and $\Gamma \vdash P^\sigma : D$. Then we have, among other results:
   - $\Gamma \vdash S_P : (D \to B' \to A) \to (D \to B') \to D \to A$,
   - $\Gamma \vdash M_1 : D \to B' \to A$,
   - $\Gamma \vdash M_2 : D \to B'$,
   - $\Gamma \vdash P^\sigma : D$ and
   - $\Gamma' \vdash P : D$.

   Using these hypotheses, we will prove that $\Gamma \vdash M' : A$. That is, $\Gamma \vdash M_1 P^\sigma (M_2 P^\sigma) : A$.

$$\dfrac{\dfrac{\Gamma \vdash M_1 : D \to B' \to A \quad \Gamma \vdash P^\sigma : D}{\Gamma \vdash M_1 P^\sigma : B' \to A}\,(\to\text{-Elim}) \qquad \dfrac{\Gamma \vdash M_2 : D \to B' \quad \Gamma \vdash P^\sigma : D}{\Gamma \vdash M_2 P^\sigma : B'}\,(\to\text{-Elim})}{\Gamma \vdash M_1 P^\sigma (M_2 P^\sigma) : A}\,(\to\text{-Elim})$$

3. $M = \Pi^1_{PQ}(P^\sigma Q^\sigma)$ and $M' = P^\sigma$.

   By Inversion Lemma, there is a $B\Gamma \vdash \Pi^1_{PQ} : B \to A$ and $\Gamma \vdash P^\sigma Q^\sigma : B$. Using the Inversion Lemma on the right side of the application, we find that $\Gamma \vdash P^\sigma : C \to B$ and $\Gamma \vdash Q^\sigma : C$ for some C. Using the lemma again, now on the left side, we can see that $A = C \to B$ and there is a Γ' s.t. $\Gamma' \vdash P : C \to B$ and $\Gamma' \vdash Q : C$. As in the previous cases, the results about Γ' can also be derived from our other results, by using Lemma 3.5.11 twice with the types obtained for P^σ and Q^σ as the respective hypotheses. We can now conclude that $\Gamma \vdash P^\sigma : C \to B$; and since $M' = P^\sigma$ and $A = C \to B$, this means that $\Gamma \vdash M' : A$.

4. $M = \Pi^2_{PQ}(P^\sigma Q^\sigma)$ and $M' = Q^\sigma$.

   Here the results we can obtain about the types of P, Q, P^σ and Q^σ are the same as in the previous case. The only difference is that now the last use of the Inversion Lemma will reveal that $A = C$, which can also be used to type Q^σ in Γ. Thus $\Gamma \vdash \Pi^2_{PQ}(P^\sigma Q^\sigma) : C$ and $\Gamma \vdash Q^\sigma : C$, which means that $\Gamma \vdash M' : A$.

5. $M = M_1 M_2$, $M' = M_1' M_2$ and $M_1 \to_{W_P} M_1'$. By Inversion Lemma, there is a Cs.t. $\Gamma \vdash M_1 : C \to A$ and $\Gamma \vdash M_2 : C$. By the IH, $\Gamma \vdash M_1' : C \to A$. Therefore, we can derive $\Gamma \vdash M_1' M_2 : A$ by using $\to$-Elim.

6. $M = M_1 M_2$, $M' = M_1 M_2'$ and $M_2 \to_{W_P} M_2'$.

   By Inversion Lemma, there is a Cs.t. $\Gamma \vdash M_1 : C \to A$ and $\Gamma \vdash M_2 : C$. By the IH, $\Gamma \vdash M_2' : C$. We can now use $\to$-Elim to prove that $\Gamma \vdash M_1 M_2' : A$.

□

Remark 3.5.15 Notice that, whenever a pattern is involved in each case of the above proof, the information we obtain about the typability of this pattern from the Inversion Lemma turns out to be a consequence of other results we previously obtained (that is, for every pattern P, we reached the conclusion that $\Gamma' \vdash P : A$ in two different ways). This poses the question of whether the restrictions of the form $\Gamma' \vdash P : A$ on the rules are redundant. The answer is negative. In our proof of Type Preservation, we are always able to deduce $\Gamma' \vdash P : A$ by using Lemma 3.5.11 because we assume that the term M as a whole is typable, and thus know that P^σ can be given the type A in a context Γ. This is not the case if we want to type a combinator without passing it all its arguments. For example, $K_{S(SKK)(SKK)}$ is untypable, because its pattern $S(SKK)(SKK)$ cannot be typed in any context. If we removed the requirement for the pattern to be typable from the K-rule, the term $K_{S(SKK)(SKK)}$ would be typable, which is undesirable because $S(SKK)(SKK)$ has no typable instances and thus an application of the form $K_{S(SKK)(SKK)} M_1 \cdots M_n$ would never reduce (except, of course, inside each M_i).

We will now introduce some definitions and lemmas in order to prove the strong normalization of typable terms. Our proof is based on Sørensen and Urzyczyn's proof for CL [SU06], which uses a simplified version of Tait's *computability* method [Tai67].

Let SN denote the set of all strongly normalizable CL_P-terms (we will refer to the terms in this set as SN-terms). For each type B we define the standard notion of a set $[\![B]\!]$ of CL_P-terms *computable* in B:

$$
\begin{array}{rcl}
[\![\alpha]\!] & \triangleq & \mathrm{SN} \\
[\![(A \to C)]\!] & \triangleq & \{M \mid \forall N(N \in [\![A]\!] \supset MN \in [\![C]\!])\}
\end{array}
$$

The following four lemmas are necessary in order to prove the strong normalization result.

Lemma 3.5.16

(i) $[\![B]\!] \subseteq \mathrm{SN}$.

(ii) If $H_1, ..., H_k \in \mathrm{SN}$, then $xH_1...H_k \in [\![B]\!]$.

Proof.- By induction on B.

- If $B = \alpha$: the result is immediate by definition of $[\![B]\!]$.
- If $B = A \to C$: let $M \in [\![A \to C]\!]$, and x a variable. By IH *(ii)* on A, $x \in [\![A]\!]$. Therefore $Mx \in [\![C]\!]$ by definition of $[\![A \to C]\!]$. By IH *(i)* on C, we know that $Mx \in \mathrm{SN}$. This means that $M \in \mathrm{SN}$. Now we only need to prove that $xH_1...H_kN \in [\![C]\!]$ for every $N \in [\![A]\!]$. By IH *(i)*, $N \in \mathrm{SN}$. Then, by IH *(ii)*, $xH_1...H_kN \in [\![C]\!]$.

□

Lemma 3.5.17

1. If $LN(MN)H_1...H_k \in [\![B]\!]$, then $S_PLMNH_1...H_k \in [\![B]\!]$.
2. If $MH_1...H_k \in [\![B]\!]$, then $K_PMNH_1...H_k \in [\![B]\!]$.
3. If $(MN) \in \mathrm{SN}$ and $MH_1...H_k \in [\![B]\!]$, then $\Pi^1_{QR}(MN)H_1...H_k \in [\![B]\!]$.
4. If $(MN) \in \mathrm{SN}$ and $NH_1...H_k \in [\![B]\!]$, then $\Pi^2_{QR}(MN)H_1...H_k \in [\![B]\!]$.

Proof.- By induction on B, using Lemma 3.5.16.

- If $B = \alpha$:
  1. We need to prove that $S_PLMNH_1...H_k \in \mathrm{SN}$. Suppose there is an infinite reduction $T_0 \to T_1 \to \cdots$ where $T_0 = S_PLMNH_1...H_k$. First suppose that $T_i = S_PL^iM^iN^iH^i_1...H^i_k$ for all i, i.e. all the reductions are "internal". This cannot happen because
  $LN(MN)H_1...H_k \in [\![B]\!] \subseteq \mathrm{SN}$, and thus every proper subterm of T_0 is strongly normalizable (they are all subterms of $LN(MN)H_1...H_k$). It follows that a redex of the form $S_PL^iM^iN^i$ must eventually be reduced, and we have $T_j = L^jN^j(M^jN^j)H^j_1...H^j_k$ for some j. Additionally, $LN(MN)H_1...H_k \twoheadrightarrow T_j$, and we know that $LN(MN)H_1...H_k \in \mathrm{SN}$, so the reduction must terminate.
  2. the proof is similar to 1. Since $N \in \mathrm{SN}$ and $MH_1...H_k \in [\![B]\!] \subseteq \mathrm{SN}$, there is no infinite sequence of "internal" reductions starting from $K_PMNH_1...H_k$. The rest is analogous to the previous case.
  3. Analogous to 2, taking into account that $MN \in \mathrm{SN}$.

4. Analogous to 3.

- If $B = A \to C$:
  1. We need to prove that for every $T \in [\![A]\!]$: $S_P LMNH_1...H_k T \in [\![C]\!]$. We know by hypothesis that $LN(MN)H_1...H_k \in [\![A \to C]\!]$. Therefore, $LN(MN)H_1...H_k T \in [\![C]\!]$ and, by IH, $S_P LMNH_1...H_k T \in [\![C]\!]$.
  2, 3 and 4 are analogous.

□

Lemma 3.5.18 If M is not an application[6] and $H_1, ..., H_k$ are SN-terms: $\Pi^1_{QR} MH_1...H_k \in [\![B]\!]$ and $\Pi^2_{QR} MH_1...H_k \in [\![B]\!]$.

Proof.- By induction on B.

- If $B = \alpha$: all we need to prove here is that $\Pi^1_{QR} TH_1...H_k \in$ SN (and analogously for $\Pi^2_{QR} TH_1...H_k$). Since T is not an application, it cannot be an instance of the pattern QR. Thus, the subterm $\Pi^2_{QR} T$ does not form a redex, which means that only "internal" reductions are possible. And since $H_1, ..., H_k \in$ SN and T is a normal form (because it is not an application and only applications may reduce), then all the "internal" reductions must terminate.
- If $B = A \to C$: let $N \in [\![A]\!]$, then $N \in$ SN by Lemma 3.5.16(i), and we know by IH on C that $\Pi^1_{QR} TH_1...H_k N \in [\![C]\!]$ and $\Pi^2_{QR} TH_1...H_k N \in [\![C]\!]$. The property holds by definition of $[\![B]\!]$.

□

Lemma 3.5.19 If $\Gamma \vdash T : B$ then $T \in [\![B]\!]$.

Proof.- By structural induction on the term T, using the Inversion Lemma (IL).

- If T is a variable, then $T \in [\![B]\!]$ by Lemma 3.5.16(ii).
- If $T = K_P$, then by IL there exist A and Cs.t. $B = A \to C \to A$. Let $M \in [\![A]\!]$ and $N \in [\![C]\!]$, then by Lemma 3.5.16(i) $M \in$ SN and $N \in$ SN. Additionally, by Lemma 3.5.17(2), $K_P MN \in [\![A]\!]$, and thus $K_P \in [\![A \to C \to A]\!]$=$[\![B]\!]$.
- If $T = S_P$ the proof is analogous to the previous case, using three arguments which are computable in the respective types.
- If $T = \Pi^1_{QR}$, then by IL there exist A and Cs.t. $B = A \to C \to A$. So far we have the same result as with K_P, and we need to prove that, for every $M \in [\![A]\!]$ and $N \in [\![C]\!]$, $\Pi^1_{QR} MN \in [\![A]\!]$. If M is an application of the form $M_1 M_2$, this holds by Lemma 3.5.17(3), as both M and N are strongly normalizable by Lemma 3.5.16(i). Otherwise, it holds by Lemma 3.5.18.

[6] That is, M is either a variable, a combinator, or a constructor if we include constructors in the calculus (see Section 3.5.2).

- If $T = \Pi^2_{QR}$, then by IL there exist A and Cs.t. $B = A \to C$. We need to prove that $\Pi^2_{QR} M \in [\![C]\!]$ for every $M \in [\![A]\!]$. This holds either by Lemma 3.5.17(4) if M is an application, or by Lemma 3.5.18 in any other case.
- If $T = T_1 T_2$, then by IL there exists A s.t. $\Gamma \vdash T_1 : A \to B$ and $\Gamma \vdash T_2 : A$. By IH, we know that $T_1 \in [\![A \to B]\!]$ and $T_2 \in [\![A]\!]$. Then, by definition of $[\![A \to B]\!]$, $T_1 T_2 \in [\![B]\!]$.

□

Finally, we conclude with our main result of the section; it is an immediate consequence of Lemmas 3.5.16(i) and 3.5.19.

Proposition 3.5.20 (Strong Normalization) Every typable CL_P-term is SN.

Remark 3.5.21 It would be interesting to develop a (dependent) type system capable of detecting matching failure. For example, one where K_P had type $A \to B \to_P A$, where the type "$A \to_P B$" means "function with domain in A and range in B, where the argument must match the pattern P" and "$A \to B$" is the same as "$A \to_x B$". This approach would prevent matching failure if $\to$-Elim requires the matching to be successful for the application to be well-typed. However, the difficulty lies in typing applications of the form MN with M of type $A \to_P B$ and in which N reduces to an instance of P but is not one yet.

3.5.2 Modelling Data Structures

Constructors for modelling applicative data structures may be incorporated into CL_P. This is achieved by enriching CL_P with constants taken from some given set of constants $\mathcal{C}$. We illustrate with an example.

Example 3.5.22 If we are interested in supporting lists, we may assume constructors `nil` and `cons` belong to $\mathcal{C}$. A list with the elements M_1 and M_2 will be denoted by the term `cons` M_1 (`cons` M_2 `nil`). Having constructors as patterns allows us to define terms which can only be applied to structures built with a certain constructor. For example, the term $S_{\texttt{cons}\,x\,y}(K\Pi^2_{\texttt{cons}\ z})\Pi^1_{\texttt{cons}\,v\,w}$ will return the head of a non-empty list, and do nothing when applied to a term of any other form.

In general, a data structure takes the form $CM_1 \cdots M_n$, with $C \in \mathcal{C}$, and will be a normal form whenever $M_1, \cdots M_n$ are normal forms. The set of patterns can also be extended with constructors: if C is a constructor and $P_1, \cdots P_n$ are patterns ($n \geq 0$), then $CP_1 \cdots P_n$ is a pattern. As expected, $\mathsf{FV}(C) = \emptyset$ and $C^\sigma = C$ for all $C \in \mathcal{C}$ and for every substitution σ. When translating between λP and CL_P, constructors translate to themselves, and the results we have obtained in 3.3.1 and 3.3.3 still hold.

The corresponding extension of the type system consists in adding a new rule to determine the type of each constructor. For example, we can extend our calculus with natural numbers by introducing the primitive type `Nat`, and the constructors `0` and `succ`. The associated typing rules would be the following:

$$\frac{}{\Gamma \vdash \texttt{0}\colon \texttt{Nat}}\ (\textsc{Zero}) \qquad \frac{}{\Gamma \vdash \texttt{succ}\colon \texttt{Nat} \to \texttt{Nat}}\ (\textsc{Succ})$$

We can go one step further by allowing the use of parametric types, in order to achieve some measure of polymorphism. For example, we can define the type `List` A for any given type A, with $<>$ and `cons` as its constructors.

$$\frac{}{\Gamma \vdash <>\colon \texttt{List}\ A}\ (\textsc{Nil}) \qquad \frac{}{\Gamma \vdash \texttt{cons}\colon A \to \texttt{List}\ A \to \texttt{List}\ A}\ (\textsc{Cons})$$

This concept can be generalized to types with an arbitrary number of parameters (this is useful, for example, when defining tuples).

The properties mentioned in section 3.5.1 still hold when constructors are introduced. In the case of the Inversion Lemma, a new clause will be added for each new constructor. Since the addition of constructors does not introduce new reductions, Type Preservation and Strong Normalization are not affected by this extension (Lemma 3.5.16 must be extended with a new clause to deal with terms of the form $CH_1...H_k$, but these terms behave in the same way as those of the form $xH_1...H_k$, so the proof is analogous and, as a result, every constructor is computable in every type).

Remark 3.5.23 The CL_P-calculus may be extended with other constants besides constructors, namely *functional constants*. These could be incorporated as syntactic sugar to facilitate data manipulation (just like I and I_P were previously defined as SKK and S_PKK respectively). For instance, the function `pred` may be introduced with the purpose of obtaining the predecessor of a natural number, and the functions `head` and `tail` can be used to observe the elements in a list. These functions can be defined as follows:

$$\begin{array}{lcl} \texttt{pred} & \triangleq & \Pi^2_{\texttt{succ}\,x} \\ \texttt{head} & \triangleq & S_{\texttt{cons}\,x\,y}(K\Pi^2_{\texttt{cons}\,z})\Pi^1_{\texttt{cons}\,v\,w} \\ \texttt{tail} & \triangleq & \Pi^2_{\texttt{cons}\,x\,y} \end{array}$$

The definition of `head` may seem complicated at first sight, but note that, when translated to λP, the normal form of the resulting term is $\lambda \texttt{cons}\,x\,y.x$. Following the typing rules we have defined, it is easy to verify that $\Gamma \vdash \texttt{pred}\colon \texttt{Nat} \to \texttt{Nat}$, $\Gamma \vdash \texttt{head}\colon \texttt{List}\,A \to A$ and $\Gamma \vdash \texttt{tail}\colon \texttt{List}\,A \to \texttt{List}\,A$ for every context Γ and every type A.

3.6 Defining Extensions and Variants

This section considers modifications of CL_P in which either the set of terms or reduction rules are modified in some way or another. In order to aid the reader in following these modifications, we classify them into two kinds loosely specified as follows:

- A **variant** of a calculus is a new calculus which is obtained from the former by making minor modifications to the set of terms and/or the reduction rules (for example, by imposing restrictions to the terms which are considered well-formed, removing a reduction rule or replacing it by a slightly different one). The modifications must be 'minor' in the sense that the connection to the original calculus is still evident.

- An **extension** is a particular form of variant in which nothing is removed or replaced; new terms and reduction rules may be added, maintaining the old ones.

A summary of the variants and extensions follows

CL_P with matching-safe patterns	Variant	Section 3.6.1
CL_P with generalized pattern matching	Extension	Section 3.6.2
$CL_\star$= CL_P with explicit matching	Variant	Section 3.7
CL_P with multiple matching	Extension	Section 3.6.3
CL_P with structural polymorphism	Extension	Section 3.6.3

We will now see some examples and related properties.

3.6.1 Possible Restrictions to the Set of Patterns

The RPC^{++} restriction allows for a wide variety of patterns, with unlimited length and depth. It was defined in this way in order to allow the user as much freedom as possible, but in some cases this freedom may not be necessary, and simpler sets of patterns may be used. We will now show some possibilities.

Matching-safe patterns

While the RPC^{++} restriction presented in Section 3.2.1 is enough to guarantee confluence, it is possible to impose further restrictions in order to avoid dealing with unification while checking the syntax of patterns. This is done by eliminating patterns which contain matching failures within them. In a setting where matching failures are unwanted, terms of the form $K_K x S$ would be considered undesirable, and thus we would not want them to be used as patterns, since they would only be matched by terms of that same unwanted form.

In this variant, the syntax of patterns is restricted to:

$$P, Q, P_1, \cdots P_n ::= \quad x \mid K_P \mid S_P \mid K_P P_1 \mid S_P P_1 \mid S_P P_1 P_2 \mid \Pi^1_{PQ} \mid \Pi^2_{PQ} \mid C P_1 \cdots P_n$$

with $n \geq 0$, $P'P$ a pattern, and maintaining linearity. Here C represents any constructor.

Using this set of patterns results in a confluent calculus, as it is a subset of the patterns that satisfy RPC^{++}. It also removes the hassle of having to use unification to determine whether a pattern is well-formed. We will refer to the patterns in this set as *matching-safe patterns*.

Remark 3.6.1 Every λP-pattern which satisfies RPC^{+} translates into a matching-safe CL_P-pattern: since RPC^{+} restricts patterns to original λ-terms, the result of the translation of a λP-pattern satisfying RPC^{+} to CL_P is a CL-term (all its subscripts are variables) and, by Remark 3.3.6, it is also a CL_P-pattern. As such, they must fit the syntax presented in Section 3.2.1, and they may not contain combinators whose subscripts do not unify with one of their arguments (because variables unify with everything). Thus, the only option left is for that argument not to be present.

3.6.2 Parameterizing Pattern-Matching

So far we have presented a calculus which handles pattern-matching by means of substitutions, using the syntactic matching mechanisms of a TRS. In this section we explore a formulation of CL_P in which the pattern matching process is abstracted away and computed outside the formalism itself. The reduction rules of CL_P (cf. Definition 3.2.8) are replaced by the contextual closure of the following ones:

$$\begin{array}{lll} K_P MN & \to M & \Leftarrow \texttt{match}(P, N) \\ S_P M_1 M_2 N & \to M_1 N (M_2 N) & \Leftarrow \texttt{match}(P, N) \\ \Pi^1_{PQ}(MN) & \to M & \Leftarrow \texttt{match}(PQ, MN) \\ \Pi^2_{PQ}(MN) & \to N & \Leftarrow \texttt{match}(PQ, MN) \end{array}$$

The resulting system is a conditional TRS, where reduction depends on the matching predicate. The first argument of `match` must be a pattern, and the second, an arbitrary CL_P-term. The use of an external matching predicate will be referred to as **generalized pattern matching**. If we define $\texttt{match}(P, M) = (\exists\sigma$ substitution such that $M = P^\sigma)$, then the resulting reduction system will be $\to_{W_P}$.

A recursive formulation of the standard syntactic matching would be the following:

$$\begin{array}{lcll} \texttt{match}(P, P) & \triangleq & \texttt{True} & \\ \texttt{match}(x, M) & \triangleq & \texttt{True} & \text{if } x \in \mathcal{X} \\ \texttt{match}(P, M) & \triangleq & \texttt{False} & \text{if } P \text{ is a combinator or constructor and } M \neq P \\ \texttt{match}(PQ, MN) & \triangleq & \texttt{match}(P, M) \wedge \texttt{match}(Q, N) & \end{array}$$

Replacing this by a different matching predicate would yield a different reduction system. The matching predicate can be made as simple or as complex as desired, ranging from a simple syntactic verification to an elaborate algorithm. Confluence depends on the matching function. If, for example, we defined the matching function so that $\texttt{match}(P, M) \Leftrightarrow (P \in \mathcal{X} \vee M = PP)$, then the term $K_{KS}x(KS(KS))$ would have two different normal forms: x and $K_{KS}xS$. We will see that the new calculus will be confluent if the resulting **parallel reduction** is **coherent** (Def. 3.6.5).

Parallel Reduction and Coherence

Definition 3.6.2 Given a reduction $\to$, we define its **parallel reduction** $\rightrightarrows$ inductively as follows:

$$\frac{M \to N}{M \rightrightarrows N} \qquad \frac{}{M \rightrightarrows M} \qquad \frac{M \rightrightarrows M' \quad N \rightrightarrows N'}{MN \rightrightarrows M'N'}$$

The idea of the parallel reduction is to reduce all terms involved in an application simultaneously. Note that, unlike the simultaneous reduction in λP, the parallel reduction cannot contract nested redexes in one step. For example, $K(Kxy)z$ does not reduce to x in one step of $\rightrightarrows$.

Lemma 3.6.3 For every pattern P, term Q, and substitution σ: if $P^\sigma \rightrightarrows Q$ then there is a σ' s.t. $Q = P^{\sigma'}$, where $\rightrightarrows$ is the parallel reduction for $\rightarrow_{W_P}$.

Proof.- By structural induction on the pattern P.

- $P = x$ with $x \in \mathcal{X}$, every term (Q in particular) will be an instance of P, and we can take $\sigma' = \{x \leftarrow Q\}$.
- P is a combinator or a constructor: $P^\sigma = P$, and therefore $Q = P$ (because P is not an application, and thus cannot reduce to anything other than itself). In this case, any substitution will do. Take for example $\sigma' = \emptyset$.
- $P = P_1P_2$: since P_1P_2 is a pattern, it must comply with all the restrictions imposed by RPC^{++}. This means that P_1 and P_2 are CL_P-patterns, and P_1P_2 does not unify with the left-hand-side of a W_P-rule. Thus, $(P_1P_2)^\sigma$ is not a W_P-redex, and it may only reduce to Q if $Q = Q_1Q_2$ with $P_1^\sigma \rightrightarrows Q_1$ and $P_2^\sigma \rightrightarrows Q_2$ (we're taking into account the possibility that $Q_1 = P_1^\sigma$ and/or $Q_2 = P_2^\sigma$, in which case we still have $P_1^\sigma \rightrightarrows Q_1$ and $P_2^\sigma \rightrightarrows Q_2$ because $\rightrightarrows$ is reflexive). Then, by IH, $\exists \sigma'_1, \sigma'_2$s.t. $Q_1 = P_1^{\sigma'_1}$ and $Q_2 = P_2^{\sigma'_2}$. Since P is a pattern and all CL_P patterns are linear, then $\mathsf{FV}(P_1) \cap \mathsf{FV}(P_2) = \emptyset$. Take $\sigma' = \sigma'_1|_{\mathsf{FV}(P_1)} \cup \sigma'_2|_{\mathsf{FV}(P_2)}$.

□

Lemma 3.6.4 For every application pattern PQ and all substitutions σ, σ': if $(PQ)^\sigma \rightrightarrows (PQ)^{\sigma'}$, then $P^\sigma \rightrightarrows P^{\sigma'}$ and $Q^\sigma \rightrightarrows Q^{\sigma'}$.

Proof.- PQ is an application pattern, which means it must be of the form $TM_1 \cdots M_n$, where $n \geq 1$, $M_1, \cdots M_n$ are patterns and T is either a combinator or a constructor. A simple induction on the number of arguments can prove that $P^\sigma = TM_1^\sigma \cdots M_{n-1}^\sigma \rightrightarrows TM_1^{\sigma'} \cdots M_{n-1}^{\sigma'} = P^{\sigma'}$ and $Q^\sigma = M_n^\sigma \rightrightarrows M_n^{\sigma'} = Q^{\sigma'}$, using Lemma 3.6.3 and the IH when $n > 1$. □

Definition 3.6.5 A reduction $\rightarrow_R$ satisfies the **coherence** property if:

1. For every pattern P and all terms M, N: if $\mathtt{match}(P, M)$ and $M \rightarrow_R N$, then $\mathtt{match}(P, N)$.

2. For every application pattern PQ and all terms M, N, T: if $\mathtt{match}(PQ, MN)$ and $MN \rightarrow_R T$, then there are terms M', N' s.t. $T = M'N'$, $M \overset{=}{\rightarrow}_R M'$ and $N \overset{=}{\rightarrow}_R N'$.

Intuitively, the first condition means that a term which matches a pattern can only reduce to terms which match the same pattern; while the second condition means that, if an application matches an application pattern, then the result of any one-step reduction of this application can be reached by reducing (in zero or one steps) each side of the application separately.

Lemma 3.6.6 The parallel reduction associated with $\rightarrow_{W_P}$ is coherent:

1. For every pattern P, and all CL_P-terms M, N: if $\mathtt{match}(P, M)$ and $M \rightrightarrows N$, then $\mathtt{match}(P, N)$.

2. For every application pattern PQ and all CL_P-terms M, N, T: if $\texttt{match}(PQ, MN)$ and $MN \rightrightarrows T$, then there exist terms M', N's.t. $T = M'N'$, $M \overset{=}{\rightrightarrows} M'$ and $N \overset{=}{\rightrightarrows} N'$.

Proof.- Part 1 is a direct consequence of Lemma 3.6.3, as $\texttt{match}(P, M) = (\exists\sigma$ substitution s.t. $M = P^\sigma)$ (and analogously for N). Simply replace M by P^σ and N by $P^{\sigma'}$, with σ' the substitution obtained from Lemma 3.6.3.

Similarly, part 2 is a direct consequence of Lemma 3.6.4, replacing M by P^σ, N by Q^σ, M' by $P^{\sigma'}$, N' by $Q^{\sigma'}$, and $\overset{=}{\rightrightarrows}$ by $\rightrightarrows$ (since $\rightrightarrows$ is reflexive, it is equal to its reflexive closure). T will always be an application by Lemma 3.6.3, as only an application can match another application. □

Corollary 3.6.7 The RPC^{++} restriction ensures the coherence of the parallel reduction.

Proof.- This is a direct consequence of Lemma 3.6.6. □

We will say that a calculus is a **coherent variant** of the CL_P-calculus if its reduction rules fit the schema presented at the beginning of Section 3.6.2 and its parallel reduction is **coherent** (that is, it satisfies the coherence property). This concept was inspired by Van Oostrom's original RPC restriction (see Definition 3.1.4).

Confluence

We will now prove the confluence of any coherent variant of the CL_P-calculus (with or without constructors). We can no longer use the orthogonality result, as we have no guarantee that our variant can be formulated as an orthogonal TRS (or even an orthogonal Higher-order Rewriting System). Instead, we will use the parallel reduction technique. To this effect, we will use the following definition and lemma. A reduction $\rightarrow_R$ satisfies the **diamond** property (denoted as $\rightarrow_R \models \diamond$) if whenever $A \rightarrow_R B$ and $A \rightarrow_R C$, there is a D s.t. $B \rightarrow_R D$ and $C \rightarrow_R D$.

Lemma 3.6.8 Let $\rightarrow_1$ be a binary relation; if there exists $\rightarrow_2$ s.t. $\rightarrow_1 \subseteq \rightarrow_2 \subseteq \twoheadrightarrow_1$ and $\rightarrow_2 \models \diamond$, then $\rightarrow_1$ is confluent.

Proof.- See [BN98, BKdV03]. □

The following lemmas show that the parallel reduction satisfies the hypotheses of Lemma 3.6.8, using the reduction defined for our calculus ($\rightarrow$) as $\rightarrow_1$.

Lemma 3.6.9 $\rightarrow \subseteq \rightrightarrows \subseteq \twoheadrightarrow$

Proof.- The first inclusion is immediate from the first rule of the definition of $\rightrightarrows$.

The second inclusion derives from the fact that $\twoheadrightarrow$ is the reflexive transitive closure of our reduction $\rightarrow$. This means that:

1. $\rightarrow \subseteq \twoheadrightarrow$, which covers rule 1 of the definition of $\rightrightarrows$.
2. $\twoheadrightarrow$ is reflexive, which covers rule 2.
3. $\twoheadrightarrow$ is transitive. If $M \twoheadrightarrow M'$, $N \twoheadrightarrow N'$ and $MN \twoheadrightarrow M'N \twoheadrightarrow M'N'$, then $MN \twoheadrightarrow M'N'$. This covers rule 3 of the definition of $\rightrightarrows$. Therefore, $\rightrightarrows \subseteq \twoheadrightarrow$. □

It can be proved that $\rightrightarrows$ satisfies the diamond property, using the fact that $\rightrightarrows$ is coherent.

Lemma 3.6.10 If $\rightarrow$ is the reduction associated to a coherent variant of CL_P, then $\rightrightarrows \models \diamond$.

Proof.- In order to prove that $\rightrightarrows$ satisfies the diamond property, we will use the fact that $\rightrightarrows$ is coherent. We will prove that $M \rightrightarrows M_1 \wedge M \rightrightarrows M_2 \supset \exists M_3$ s.t. $M_1 \rightrightarrows M_3 \wedge M_2 \rightrightarrows M_3$ by induction on the reduction $M \rightrightarrows M_1$.

- $M \rightrightarrows M_1$ is $M \rightrightarrows M$: take $M_3 = M_2$.
- $M \rightrightarrows M_1$ is $M'M'' \rightrightarrows M_1'M_1''$ with $M' \rightrightarrows M_1'$ and $M'' \rightrightarrows M_1''$: we will need to analyze the reduction $M \rightrightarrows M_2$. Since the previous case covers reflexivity, there are 2 remaining possibilities:
  1. $M_2 = M_2'M_2''$ with $M' \rightrightarrows M_2'$ and $M'' \rightrightarrows M_2''$.
     By IH there exist M_3' and M_3'' such that $M_1' \rightrightarrows M_3'$, $M_2' \rightrightarrows M_3'$, $M_1'' \rightrightarrows M_3''$ and $M_2'' \rightrightarrows M_3''$.
     Take $M_3 = M_3'M_3''$. $M_1 \rightrightarrows M_3$ and $M_2 \rightrightarrows M_3$ by rule 3.
  2. M_2 is the result of a $\rightarrow$ reduction in the root of $M'M''$. We will analyze all the possibilities by cases according to the rule applied.
     - $M'M'' = K_PNM''$: $M' = K_PN$, $M_2 = N$, $K_PN \rightrightarrows M_1'$, $\texttt{match}(P, M'')$ and $M'' \rightrightarrows M_1''$. Here $M_1' = K_PN'$, with $N \rightrightarrows N'$. Take $M_3 = N'$. (Note that $K_PN'M_1'' \rightrightarrows N'$ because $M'' \rightrightarrows M_1''$ and then, by coherence, M_1'' matches P).

$$\begin{tikzcd}
 & K_PNM'' \arrow[dl, Rightarrow] \arrow[dr, Rightarrow] & \\
K_PN'M_1'' \arrow[dr, Rightarrow] & & N \arrow[dl, Rightarrow] \\
 & N' &
\end{tikzcd}$$

     - $M'M'' = S_PN_1N_2M''$: $M' = S_PN_1N_2$, $\texttt{match}(P, M'')$, $M_2 = N_1M''(N_2M'')$, $S_PN_1N_2 \rightrightarrows M_1'$ and $M'' \rightrightarrows M_1''$. In this case $M_1' = S_PN_1'N_2'$, with $S_PN_1 \rightrightarrows S_PN_1'$ and $N_2 \rightrightarrows N_2'$. Then $N_1 \rightrightarrows N_1'$ and $N_2 \rightrightarrows N_2'$.
       Since M_1'' matches P by coherence, $M_1 = M_1'M_1'' = S_PN_1'N_2'M_1'' \rightrightarrows N_1'M_1''(N_2'M_1'')$. And since $N_1 \rightrightarrows N_1'$, $N_2 \rightrightarrows N_2'$ and $M'' \rightrightarrows M_1''$, then $M_2 = N_1M''(N_2M'') \rightrightarrows N_1'M_1''(N_2'M_1'')$. Take $M_3 = N_1'M_1''(N_2'M_1'')$.

$$\begin{tikzcd}
 & S_PN_1N_2M'' \arrow[dl, Rightarrow] \arrow[dr, Rightarrow] & \\
S_PN_1'N_2'M_1'' \arrow[dr, Rightarrow] & & N_1M''(N_2M'') \arrow[dl, Rightarrow] \\
 & N_1'M_1''(N_2'M_1'') &
\end{tikzcd}$$

– $M'M'' = \Pi^1_{PQ}(P'Q')$: $M' = \Pi^1_{PQ}$, $M'' = P'Q'$, $\texttt{match}(PQ, P'Q')$, $M_2 = P'$, $\Pi^1_{PQ} \rightrightarrows M_1'$ and $P'Q' \rightrightarrows M_1''$.
Since Π^1_{PQ} is not an application, it can only reduce to itself, hence $M_1' = \Pi^1_{PQ}$; and since $\rightrightarrows$ is coherent, we can conclude that $P'Q' \rightrightarrows M_1'' \supset M_1'' = P''Q''$ for some terms P'' and Q'' such that $\texttt{match}(PQ, P''Q'')$, with $P' \rightrightarrows P''$ and $Q' \rightrightarrows Q''$.
Since $P''Q''$ matches PQ, then $M_1 = \Pi^1_{PQ}(P''Q'') \rightrightarrows P''$. Finally, since $M_2 = P'$ and $P' \rightrightarrows P''$, this means that $M_2 \rightrightarrows P''$. Take $M_3 = P''$.

$$\begin{tikzcd}
 & \Pi^1_{PQ}(P'Q') \arrow[dl, Rightarrow] \arrow[dr, Rightarrow] & \\
\Pi^1_{PQ}(P''Q'') \arrow[dr, Rightarrow] & & P' \arrow[dl, Rightarrow] \\
 & P'' &
\end{tikzcd}$$

– $M'M'' = \Pi^2_{PQ}(P'Q')$: this case is analogous to the previous one, using Q' and Q'' in place of P' and P''. Take $M_3 = Q''$.

$$\begin{tikzcd}
 & \Pi^2_{PQ}(P'Q') \arrow[dl, Rightarrow] \arrow[dr, Rightarrow] & \\
\Pi^2_{PQ}(P''Q'') \arrow[dr, Rightarrow] & & Q' \arrow[dl, Rightarrow] \\
 & Q'' &
\end{tikzcd}$$

- $M \rightrightarrows M_1$ is a $\rightarrow$-reduction in the root of M. Since at most one $\rightarrow$-rule can be used to reduce at the root of any given term (the left side of each rule has a different left-most combinator), M_2 must be either M (rule 2), $M_2'M_2''$ with $M = M'M''$, $M' \rightrightarrows M_2'$ and $M'' \rightrightarrows M_2''$ (rule 3), or M_1 (rule 1). The first 2 possibilities (rules 2 and 3) have already been covered above. As for the last case ($M_2 = M_1$), take $M_3 = M_1$, and $M_1 = M_2 \rightrightarrows M_3$ holds by reflexivity of $\rightrightarrows$.

□

As a result of Lemmas 3.6.8, 3.6.9 and 3.6.10, every coherent variant of CL_P is confluent.

Remark 3.6.11 Note that when using this generalized notion of pattern matching, patterns are no longer restricted to those satisfying RPC^{++}. Any coherent variant will be confluent.

3.6.3 Representing other pattern calculi

By introducing new terms and/or changing the set of allowed patterns we obtain different calculi, which can introduce new features to our calculus and represent different languages.

The λC-calculus. For example, we can translate λC into a variant of CL_P. Recall from the introduction that the λC-calculus is a variation of λP in which patterns are taken to be algebraic terms, and where multiple pattern matching is allowed by means of generalized abstractions of the form $\lambda P_1.M_1|\ldots|\lambda P_k.M_k$. The reduction relation is a generalization of that of λP:

$$(\lambda P_1.M_1|\ldots|\lambda P_k.M_k)P_i^\sigma \to M_i^\sigma$$

Confluence has been proved for linear constructor patterns, as well as for *rigid* multiple-patterns (i.e. patterns satisfying the *Rigid Pattern Condition* introduced in [vO90] for λP, and where the different P_i in $\lambda P_1.M_1|\ldots|\lambda P_k.M_k$ do not unify with each other). Since λC does not allow abstractions to be used as patterns, we may restrict our set of patterns to only variables and constructors applied to 0 or more arguments, which must themselves be patterns. In other words, the syntax for the patterns would be as follows:

$$P, P_1 \cdots P_n ::= x \mid \ CP_1 \cdots P_n$$

with $n \geq 0$. Note that this set of patterns satisfies RPC^{++}.

The syntax of CL_P is extended with *case* combinators in order to allow matching over multiple patterns, and even provide different responses depending on the pattern that has been matched. The case combinators have the form $\theta_{P_1,\ldots,P_n}$ with $n \geq 1$, where P_1,...,P_n are patterns which are not pairwise unifiable. A new rule schema is introduced:

$$\theta_{P_1,\ldots,P_n}x_1\ldots x_nP_i \to x_iP_i, \quad \text{if } 1 \leq i \leq n$$

Definition 3.6.12 The translation from λC to $CL_P + \theta$ (that is, CL_P extended with the *case* combinator and constructors) is as follows:

$$\begin{array}{rcl}
x_{CL} & \triangleq & x \\
C_{CL} & \triangleq & C \\
(MN)_{CL} & \triangleq & M_{CL}N_{CL} \\
(\lambda P.N)_{CL} & \triangleq & \lambda^* P_{CL}.N_{CL} \\
(\lambda P_1.M_1|\ldots|\lambda P_n.M_n)_{CL} & \triangleq & \theta_{(P_1)_{CL},\ldots,(P_n)_{CL}}(\lambda^*(P_1)_{CL}.(M_1)_{CL})\ldots(\lambda^*(P_n)_{CL}.(M_n)_{CL}), \\
 & & \text{if } n \geq 2
\end{array}$$

With λ^* defined as before (Definition 3.3.1).

Here is an example of how the translation works.

$$(\lambda <> . <> |\lambda \texttt{cons}\, x\, y.y)(\texttt{cons}\ x_1(\texttt{cons}\ x_2 <>))$$

would translate to:

$$\theta_{<>,\texttt{cons}\,x\,y}(K <>)(S_{\texttt{cons}\,x\,y}(K(SKK))\Pi^2_{\texttt{cons}\,x\,y})(\texttt{cons}\ x_1(\texttt{cons}\ x_2 <>))$$

which reduces as follows:

$$
\begin{array}{ll}
\theta_{<>,\mathtt{cons}\,x\,y}(K\ <>)(S_{\mathtt{cons}\,x\,y}(K(SKK))\Pi^2_{\mathtt{cons}\,x\,y})(\mathtt{cons}\ x_1(\mathtt{cons}\ x_2\ <>)) & \to_{W_P} \\
S_{\mathtt{cons}\,x\,y}(K(SKK))\Pi^2_{\mathtt{cons}\,x\,y}(\mathtt{cons}\ x_1(\mathtt{cons}\ x_2\ <>)) & \to_{W_P} \\
K(SKK)(\mathtt{cons}\ x_1(\mathtt{cons}\ x_2\ <>))(\Pi^2_{\mathtt{cons}\,x\,y}(\mathtt{cons}\ x_1(\mathtt{cons}\ x_2\ <>))) & \to_{W_P} \\
SKK(\Pi^2_{\mathtt{cons}\,x\,y}(\mathtt{cons}\ x_1(\mathtt{cons}\ x_2\ <>))) & \to_{W_P} \\
SKK(\mathtt{cons}\ x_2\ <>) & \twoheadrightarrow_{W_P} \\
\mathtt{cons}\ x_2\ <>\ . &
\end{array}
$$

The requirement for the patterns in the case not to be pairwise unifiable is needed – as in λC – to ensure that the argument will match at most one pattern.

The argument which will be matched against the patterns is placed immediately after the case combinator to facilitate potential implementations, as well as translations to other combinatory calculi which handle the matching in a sequential manner. Since the number of required arguments depends on the number of patterns used within the case combinator, skipping through a varying (and arbitrarily high) number of arguments in order to find the term that will be matched against each pattern would be both difficult and inefficient.

We now prove simulation for the general case (Corollary. 3.6.15).

Lemma 3.6.13 $(M^\sigma)_{CL} = (M_{CL})^{\sigma_{CL}}$.

Proof.- First note that Lemma 3.3.8 still holds, since none of the new cases for the translations introduce nor erase free variables. We prove this result by induction on the size of M, which is now a λC-term.

For the cases where M is a variable, an application or a single pattern abstraction, the proof is analogous to that of Lemma 3.3.9. If M is a constructor, the result holds trivially, since $(M^\sigma)_{CL} = M_{CL} = M = (M_{CL})^{\sigma_{CL}}$.

If $M = \lambda P_1.M_1|\ldots|\lambda P_k.M_k$, then $M^\sigma = (\lambda P_1.M_1)^\sigma|\ldots|(\lambda P_n.M_n)^\sigma$ and, by IH, $((\lambda P_i.M_i)^\sigma)_{CL} = ((\lambda P_i.M_i)_{CL})^{\sigma_{CL}}$ for $i \in \{1,\ldots,n\}$.

$$
\begin{array}{lcl}
(M^\sigma)_{CL} & = & \\
\theta_{(P_1)_{CL},\ldots,(P_n)_{CL}}((\lambda P_1.M_1)_{CL})^{\sigma_{CL}}\ldots((\lambda P_n.M_n)_{CL})^{\sigma_{CL}} & = & \\
\theta_{(P_1)_{CL},\ldots,(P_n)_{CL}}(\lambda^*(P_1)_{CL}.(M_1)_{CL})^{\sigma_{CL}}\ldots(\lambda^*(P_n)_{CL}.(M_n)_{CL})^{\sigma_{CL}} & = & (M_{CL})^{\sigma_{CL}}.
\end{array}
$$

Keep in mind that substitutions in λC, as in λP, do not affect the pattern of an abstraction, since its variables are bound. This means that it is safe to use $(P_1)_{CL},\ldots,(P_n)_{CL}$ as subindices for θ without applying the substitution to these patterns. □

Lemma 3.6.14 $\theta_{P_1,\ldots,P_n}(\lambda^*P_1.M_1)\ldots(\lambda^*P_n.M_n)P_i^\sigma \twoheadrightarrow_{W_P} M_i^\sigma$ for $1 \leq i \leq n$, if $\mathsf{dom}(\sigma) \subseteq \mathsf{FV}(P_i)$.

Proof.- First note that Lemma 3.3.14 still holds, since the definition of λ^* has not changed. $\theta_{P_1,\ldots,P_n}(\lambda^*P_1.M_1)\ldots(\lambda^*P_n.M_n)P_i^\sigma \to_{W_P} (\lambda^*P_n.M_n)P_i^\sigma$ by the new reduction rule. By Lemma 3.3.14, $(\lambda^*P_n.M_n)P_i^\sigma \twoheadrightarrow_{W_P} M_i^\sigma$. □

Corollary 3.6.15 $((\lambda P_1.M_1|\ldots|\lambda P_k.M_k)P_i^\sigma)_{CL} \twoheadrightarrow_{W_P} M_i^\sigma$ if $\mathsf{dom}(\sigma) \subseteq \mathsf{FV}(P_i)$.

Proof.- If $n \geq 2$, then by Lemma 3.6.13, $(P_i^\sigma)_{CL} = ((P_i)_{CL})^{\sigma_{CL}}$.

$$\begin{array}{ll} ((\lambda P_1.M_1|\ldots|\lambda P_k.M_k)P_i^\sigma)_{CL} & = \\ (\lambda P_1.M_1|\ldots|\lambda P_k.M_k)_{CL}((P_i)_{CL})^{\sigma_{CL}} & = \\ \theta_{(P_1)_{CL},\ldots,(P_n)_{CL}}(\lambda^*(P_1)_{CL}.(M_1)_{CL})\ldots(\lambda^*(P_n)_{CL}.(M_n)_{CL})((P_i)_{CL})^{\sigma_{CL}}. & \end{array}$$

The result follows by Lemma 3.6.14.

If $n = 1$, the result holds by Lemmas 3.6.13 and 3.3.14, as in Corollary 3.3.15. □

Generalized pattern matching can also be used to introduce structural polymorphism into our calculus. For example, we can now define a term that returns the first argument of a data structure defined by the application of a constructor to two terms (a list with a first element followed by another list, a tree made by combining two trees, etc.), and returns the data structure unchanged if it has fewer than two arguments (and becomes blocked if the argument has more than two arguments or is not a data structure). If we introduce a new pattern ♠ and extend the matching predicate so that ♠ is matched by every constructor (and only by constructors), then the result will be achieved by the term $\theta_{\spadesuit xy,\spadesuit z,\spadesuit}(S(K\Pi^2_{\spadesuit w})\Pi^1_{\spadesuit uv})II$. This technique can be applied with other structures and patterns, achieving an effect resembling path polymorphism (the difference is that we're using an ad-hoc matching predicate that is external to the calculus in order to produce this effect). For example, we can now define a term that returns the first argument of a data structure defined by the application of a constructor to two terms (a list with a first element followed by another list, a tree made by combining two trees, etc.), and returns the data structure unchanged if it has fewer than two arguments (and becomes blocked if the argument has more than two arguments or is not a data structure).

In order to define a pattern that is matched by data structures in general, the matching predicate can be extended so that every term of the form $CP_1...P_n$, with C a constructor and $n \geq 0$, matches the new pattern. Since terms of this form may only have internal reductions, this extension is coherent and thus confluence still holds. Data structures which are not of the form $CP_1...P_n$ do not present a problem, as they can be reduced to that form and then matched against the pattern.

The *SF*-calculus. Finally, there is another variant which can be used to capture other forms of polymorphism and partially simulate the *SF*-calculus. It consists of introducing two new special patterns (@ and ∘) in addition to the case combinator θ. However, we only need to use the θ combinator with these two patterns, so we can simply introduce one form of the case combinator, namely $\theta_{\circ,@}$. In this extension we also allow the use of the combinators $\Pi^1_@$ and $\Pi^2_@$, which will allow us to decompose a wide range of applications.

We extend the original matching algorithm so that the pattern @ is matched by all (instances of) application patterns. That is, every term of the forms K_PM, S_PM, $S_PM_1M_2$ and, eventually, all applied constructors. This way the @ pattern will be capable of dealing with applications, in spite of the presence of infinitely many constants. We are assuming the use of matching-safe patterns.

The pattern ∘ will be matched by all constants, namely K_P, S_P, Π^1_{PQ}, Π^2_{PQ}, $\Pi^1_@$, $\Pi^2_@$ and eventually all constructors without their arguments. Once more, a single pattern can 'absorb' the roles of infinitely many patterns.

Given this extension, we define:

$$\hat{\mathrm{F}} \triangleq S(KK)(S(S(KS)(S(K(S(SKK)))(S(KK)\Pi^1_{@})))(S(KK)\Pi^2_{@}))$$

This term, also a pattern, is the result of unfolding the expression $\lambda^* x.\lambda^* y.\lambda^* z.z(\Pi^1_{@}x)(\Pi^2_{@}x)$. When applied to an instance of an application pattern $(PQ)^\sigma$, it reduces in many steps to $K(S(SI(KP^\sigma))(KQ^\sigma))$. This way, a term of the form $\hat{\mathrm{F}}(PQ)^\sigma MN$ will reduce in many steps to $NP^\sigma Q^\sigma$.

This becomes useful when we use $\hat{\mathrm{F}}$ as an argument for the case combinator. The term $\theta_{\circ,@}(KK)\hat{\mathrm{F}}$ can partially simulate the behavior of the combinator F from the SF-calculus, if we consider all instances of application patterns as "*factorable forms*". The reduction rules of the SF-calculus are the following:

$$\begin{array}{lcll} SMNX & \to & MX(NX) & \\ FOMN & \to & M & \text{if } O \text{ is a constant.} \\ F(PQ)MN & \to & NPQ & \text{if } PQ \text{ is a factorable form.} \end{array}$$

Remark 3.6.16 If we define F as $\theta_{\circ,@}(KK)\hat{\mathrm{F}}$, we can simulate those same reductions in many steps. It is not, however, a full simulation of the SF-calculus, for the following reasons: first, while the SF-calculus has only 2 constants, we have an infinite number of them. Second, the term F is a constant in the SF-calculus, while in our extension it is a factorable form. This means that reducing the term $FFMN$ will yield different results in each calculus, and thus ours is not an exact simulation. It would be possible to patch the matching algorithm in order to make $\hat{\mathrm{F}}$ match the pattern $\circ$ and not @, but such a modification is not necessary to achieve the distinction between atoms and compounds: the term $\texttt{isComp} = \lambda x.Fx(KI)(K(KK))$ defined in [JGW11] can also be defined here as $\lambda^* x.Fx(KI)(K(KK))$ for our definition of F. Decomposition of "factorable forms" is already achieved in our system by projectors, thus we can already have a measure of structural polymorphism.

This extension is coherent for the same reason as the ♠ extension (terms which match the pattern @ can only have internal reductions, while terms that match the pattern $\circ$ cannot be reduced), thus confluence still holds (note that the rules for the θ combinators do not overlap with each other – since matching against @ and $\circ$ are mutually exclusive –, nor with any of the original rules).

3.7 Introducing Explicit Matching

While CL_P enjoys all the advantages of a TRS, such as eliminating the hassle of dealing with bound variables, its pattern-matching mechanism is implicit in the reductions: matching is determined by applying a substitution to the patterns, and any direct implementations of the calculus will have to address the pattern-matching problem at some point or another. In other words, they will have to implement a version of the matching algorithm shown in Section 3.6.2. While this is slightly less costly than computing substitutions, the aim of achieving a fine-grained analysis of reduction in λP would only be partially achieved.

The aim of this section is to formulate a combinatory logic for the λ-calculus with patterns in which the pattern matching process is reformulated from a fine-grained perspective. Under this view, CL_P can be seen as an intermediate language, which serves us to prove the main properties with respect to the host language, λP, and can then be "compiled" into an even lower-level variant. The proposed calculus eliminates not only the need for implicit pattern-matching, but also the need for α-conversion among patterns.

Designing such a system presents some difficulties. It would be tempting to simply modify the rules so that the patterns are passed to the combinators as arguments. This would lead to rules of the form:

$$\begin{array}{lll} \widehat{K}PMN & \to_{W_P} & M \qquad\qquad \text{if } \texttt{match}(P,N) \\ \widehat{S}PLMN & \to_{W_P} & LN(MN) \quad \text{if } \texttt{match}(P,N) \\ \dots & & \end{array}$$

However, this approach would not work. First note that the above rules cannot be directly translated to a TRS formulation, as any substitution applied to the left-hand-side of the rules would affect the pattern as much as the term that is meant to match it. We would then lose the advantage of having the pattern-matching problem be implicitly solved by the matching mechanism that is inherent to every TRS. The only way to use these rules would be by solving the pattern-matching problem outside the system, and having the reduction relation not be closed under substitution. Even then, we would no longer be able to rely on orthogonality in order to ensure confluence, and the rules would have to be fine-tuned to make sure that the system remains confluent (for example, by ensuring coherence, as seen in Section 3.6.2). This would lead to imposing restrictions to some of the arguments but not others, resulting in a system whose functionality would be the same as in the original formulation of CL_P, but whose description would be complex and more confusing. It would also cause terms with a different set of free variables to be functionally equivalent: for example, Kx would be the same as Ky.

Another possibility would be to retain the combinators S and K from CL, and incorporate the pattern-matching facility by means of a new combinator M_P, with $M_PNP^\sigma \to N$ as the only pattern-handling rule. Note the simplicity of returning one argument when the other argument matches some term. This amounts to having a term M_P for each pattern P. Actually, the K_P combinator does exactly this, thus we may simply use K_P instead of M_P. Let us explore the possibilities for expressing other terms using this combinator.

Naturally, the S combinator cannot be removed (otherwise we would trivially lose the possibility of term duplication upon reduction). All we can do is try to express our CL_P-combinators (K_P, S_P, Π^1_{PQ} and Π^2_{PQ}) in terms of K_P and S. The problem arises when we try to define the projectors: they cannot be expressed with the other combinators, since they break applications, which cannot be done by combining K_P and S_P. For instance, following the idea of the above translation for S_P, a definition like $\Pi^1_{PQ} = \lambda^\star x.K_{PQ}Px = K_{PQ}P$ (and an analogous rule for Π^2_{PQ}) may seem to do the job (if a match with PQ succeeds, then return P). However, this is not what is intended: a true projector should return not P but P^σ for every matching substitution σ. So this approach does not work. In CL_P, only the projection rules can return a term containing instances of proper subterms of the pattern, rather than instances of the pattern as a whole.

One could think that the solution to this problem is to add another primitive combinator to return the corresponding instance of the matched pattern, say K'_P, with its associated rule reading $K'_P N P^\sigma \to N^\sigma$, but this would be just another syntax for the β_P-rule of λP(it would effectively bind the variables in N to the variables in the pattern), which defeats the purpose of having a calculus based on combinators.

Therefore, the matching facility should be present for the projectors too. As a matter of fact we can still remove S_P from the syntax for minimality, as it is expressible as $S_P = \lambda^\star x.\lambda^\star y.\lambda^\star z.K_P(xz(yz))z = S(S(KS)(S(K(S(KS)))(S(K(S(K(S(KK_P)))))S)))(K(K(SKK)))$. Of course, the lengths of derivations involving S_P would increase and normal forms would not be preserved[7], but the whole formulation would still be confluent and abstractions would still be simulated. However, there does not seem to be much to gain from doing this.

In this section we present a possible solution which, at the cost of introducing new terms and longer reductions, eliminates the need for external matching algorithms and α-conversions. As a result, it can be used as a base language for compiling expressions written in CL_P.

3.7.1 The $CL_\star$-calculus

The set of $CL_\star$-terms is described by the syntax:

$$M_1, M_2, P \ ::= \ \star \ | \ x \ | \ K \ | \ S \ | \ \mathsf{K}\langle P\rangle \ | \ \mathsf{S}\langle P\rangle \ | \ \mathsf{m}\langle P\rangle \ | \ \Pi^1 \ | \ \Pi^2 \ | \ M_1 M_2 \ | \ M_1 \sim M_2$$

where x ranges over a countably infinite set of variables $\mathcal{X}$. K, S, $\mathsf{K}\langle\bullet\rangle$, $\mathsf{S}\langle\bullet\rangle$, Π^1, Π^2 and $\mathsf{m}\langle\bullet\rangle$ are **combinators**, and the constant $\star$ works as a universal pattern which can be matched by any term (it plays the role that was assigned to variable patterns in CL_P). Note that we no longer have an infinite number of combinators. Instead, we have a finite set of combinators, some of which take a pattern as their first – obligatory – argument. We will see their behaviour in detail when we have presented the reduction rules. Intuitively, the combinator $\mathsf{m}\langle\bullet\rangle$ will be used as a matcher, while $\mathsf{S}\langle\bullet\rangle$ and $\mathsf{K}\langle\bullet\rangle$ will act as translations of the CL_P combinators restricted to matching-safe patterns, passing their arguments to $\mathsf{m}\langle\bullet\rangle$ for validation before triggering a reduction.

For example, $\mathsf{K}\langle P\rangle MN$ will reduce to $\mathsf{m}\langle P\rangle NKMN$. Then the subterm $\mathsf{m}\langle P\rangle N$ will take care of matching N against P and, if successful, will return SKK. Then the reduction will proceed from $SKKKMN$ and will eventually reach (the normal form of) M. Otherwise, reduction is blocked.

The binary operator "$\bullet \sim \bullet$" is used to aid in the pattern-matching process and determine whether two patterns are equal. This is needed in order to determine whether a combinator with a pattern matches another. In CL_P, combinators are constants and, as such, can only be matched by themselves (that is, the same combinator decorated with the same subindex, modulo α-conversion). In order to translate that into $CL_\star$, we require a mechanism to decide whether two patterns are the same. That is the role of the $\bullet \sim \bullet$ operator[8]. This operator will be treated as left-associative.

[7] For example, the term $S_K xyz$ is a normal form in CL_P, but it would be reducible with this new definition of S_P.

[8] We do not need the ability to compare arbitrary terms, only terms that result from the translation of matching-safe CL_P-patterns.

For example, $\Pi^1 \sim \Pi^1$, $\mathsf{K}\langle\mathsf{S}\langle\star\rangle\rangle \sim \mathsf{K}\langle\mathsf{S}\langle\star\rangle\rangle$ and $\mathsf{S}\langle\star\rangle \sim \mathsf{S}\langle\star\rangle$ will all succeed and eventually reduce to SKK. On the other hand, $\Pi^1 \sim \Pi^2$, $\mathsf{K}\langle\mathsf{S}\langle\star\rangle\rangle \sim \mathsf{K}\langle\star\rangle$ and $\mathsf{S}\langle\star\rangle \sim \mathsf{S}\langle\mathsf{S}\langle\star\rangle\rangle$ will all fail and reduction will become blocked.

The calculus is formulated over a first-order signature, so free variables are defined as usual. While only certain terms can be obtained from the translation of CL_P-patterns, we do not need to restrict the syntax of $CL_\star$ to distinguish the set of patterns from that of arbitrary terms, as we shall see.

We next address $W_\star$-reduction. For that we introduce a rewriting system based on the above combinators, which will simulate the version of CL_P restricted to matching-safe patterns in the sense of combinatorial completeness (and hence λP, as shown by Remark 3.6.1), preserving the pattern matching of the respective calculi.

Definition 3.7.1 $W_\star$-reduction ($\rightarrow_{W_\star}$) is given by the following rules:

$$
\begin{array}{lcl}
\text{Classical rules:} & & \\
\quad Kxy & \rightarrow & x \\
\quad Sxyz & \rightarrow & xz(yz) \\
\text{Matching introduction:} & & \\
\quad \mathsf{K}\langle p\rangle xy & \rightarrow & \mathsf{m}\langle p\rangle yKxy \\
\quad \mathsf{S}\langle p\rangle xyz & \rightarrow & \mathsf{m}\langle p\rangle zSxyz \\
\text{Projection:} & & \\
\quad \Pi^1(\mathsf{K}\langle x\rangle y) & \rightarrow & \mathsf{K}\langle x\rangle \\
\quad \Pi^2(\mathsf{K}\langle x\rangle y) & \rightarrow & y \\
\quad \Pi^1(\mathsf{S}\langle x\rangle y) & \rightarrow & \mathsf{S}\langle x\rangle \\
\quad \Pi^2(\mathsf{S}\langle x\rangle y) & \rightarrow & y \\
\quad \Pi^1(\mathsf{S}\langle x\rangle yz) & \rightarrow & \mathsf{S}\langle x\rangle y \\
\quad \Pi^2(\mathsf{S}\langle x\rangle yz) & \rightarrow & z \\
\text{Matching validation:} & & \\
\quad \mathsf{m}\langle\star\rangle x & \rightarrow & I \\
\quad \mathsf{m}\langle\Pi^1\rangle\Pi^1 & \rightarrow & I \\
\quad \mathsf{m}\langle\Pi^2\rangle\Pi^2 & \rightarrow & I \\
\quad \mathsf{m}\langle\mathsf{K}\langle p\rangle\rangle\mathsf{K}\langle x\rangle & \rightarrow & p\sim x \\
\quad \mathsf{m}\langle\mathsf{K}\langle p\rangle q\rangle(\mathsf{K}\langle x\rangle y) & \rightarrow & p\sim x(\mathsf{m}\langle q\rangle y) \\
\quad \mathsf{m}\langle\mathsf{S}\langle p\rangle\rangle\mathsf{S}\langle x\rangle & \rightarrow & p\sim x \\
\quad \mathsf{m}\langle\mathsf{S}\langle p\rangle q\rangle(\mathsf{S}\langle x\rangle y) & \rightarrow & p\sim x(\mathsf{m}\langle q\rangle y) \\
\quad \mathsf{m}\langle\mathsf{S}\langle p\rangle qr\rangle(\mathsf{S}\langle x\rangle yz) & \rightarrow & p\sim x(\mathsf{m}\langle q\rangle y)(\mathsf{m}\langle r\rangle z) \\
\text{Pattern equality:} & & \\
\quad \star\sim\star & \rightarrow & I \\
\quad \Pi^1\sim\Pi^1 & \rightarrow & I \\
\quad \Pi^2\sim\Pi^2 & \rightarrow & I \\
\quad \mathsf{K}\langle p\rangle\sim\mathsf{K}\langle x\rangle & \rightarrow & p\sim x \\
\quad \mathsf{K}\langle p\rangle q\sim(\mathsf{K}\langle x\rangle y) & \rightarrow & p\sim x(q\sim y) \\
\quad \mathsf{S}\langle p\rangle\sim\mathsf{S}\langle x\rangle & \rightarrow & p\sim x \\
\quad \mathsf{S}\langle p\rangle q\sim(\mathsf{S}\langle x\rangle y) & \rightarrow & p\sim x(q\sim y) \\
\quad \mathsf{S}\langle p\rangle qr\sim(\mathsf{S}\langle x\rangle yz) & \rightarrow & p\sim x(q\sim y)(r\sim z)
\end{array}
$$

We use I as a macro to abbreviate SKK. Also, we use variables p, q, r to indicate that they are meant to be used as patterns, although they could be instantiated with any term. Let us comment on some of the rules.

- First, note that there are similarities as well as differences between the rules for S and K and those for $\mathsf{S}\langle\bullet\rangle$ and $\mathsf{K}\langle\bullet\rangle$. The former behave in the same way as in CL, while the latter pass the last argument to the matcher in order to test it against the pattern.

- Matching is handled explicitly by the $\mathsf{m}\langle\bullet\rangle$ combinator, using $\bullet\sim\bullet$ to determine whether the subsequent levels of patterns are equal. This is done by reducing recursively until the matching either succeeds or fails. In the case of success, the identity is returned in order

to allow reduction to proceed normally. Otherwise, the reduction is blocked (internal reductions may still occur, but the one for which the matching was required will not). Normal forms containing occurrences of $\mathsf{m}\langle\bullet\rangle$ or $\bullet\sim\bullet$ are indicators of matching failure.

- As a result, the $\mathsf{K}\langle\bullet\rangle$ and $\mathsf{S}\langle\bullet\rangle$ combinators effectively block reduction until the matching is verified. They will behave like their CL counterparts if and only if the matching succeeds.

- Since the aim of $CL_\star$ is to compile expressions originally written in λP or CL_P, we are only interested in matching terms with patterns which may result from translating such expressions. Ill-formed patterns will never form a redex when passed as arguments to $\mathsf{m}\langle\bullet\rangle$, which is why it is not necessary to define a syntax for the patterns. The $\mathsf{m}\langle\bullet\rangle$ combinator will decide whether the pattern is valid or not.

- Similarly, we do not need to equip the projectors with guards to verify pattern-matching. Every λP or CL_P-term – including the CL_P projectors – can be translated to $CL_\star$ in a way that the matching is tested by the $\mathsf{K}\langle\bullet\rangle$ and $\mathsf{S}\langle\bullet\rangle$ combinators, as we shall see.

Remark 3.7.2 One may be tempted to define rules like $\mathsf{m}\langle pq\rangle(xy) \to \mathsf{m}\langle p\rangle x(\mathsf{m}\langle q\rangle y)$, but this would break confluence, since the critical pair generated by the subterms pq and xy match with the left-hand sides of most reduction rules. Therefore it becomes necessary to add several instances of this schema for the corresponding pattern combinations, and then have the matching act recursively for the sub-patterns. For a similar reason, the projectors Π^1 and Π^2 cannot be applied to arbitrary applications, but only to applications which do not form redexes.

Proposition 3.7.3 $\to_{W_\star}$ is confluent.

Proof.- Orthogonality holds: each rule is left-linear and there are no critical pairs. □

Therefore, as we had anticipated, it is not necessary to restrict the set of patterns in order to achieve confluence. A pattern could be any term, even non-linear, with active variables, or even with redexes. One of the main refinements with respect to the previous approach is that, instead of handling sub-indices, the main combinators $\mathsf{K}\langle\bullet\rangle$ and $\mathsf{S}\langle\bullet\rangle$ are unary symbols. Patterns are not rigid anymore, in the sense that an instance of a given pattern may reduce to a term that is no longer an instance. Naturally, since the calculus is confluent, this can only happen if the pattern itself can be reduced, and thus matching can be recovered by reducing the pattern.

The rules which deal with matching validation, pattern equality and projection (rather than simulating S_P and K_P) will not create the possibility of infinite derivations.

Proposition 3.7.4 (Strong Normalization of Explicit Matching) The TRS obtained by removing the first 4 reduction rules is SN.

Proof.- We use the following weight function:

$$
\begin{array}{lcl}
\| \star \| = \|K\| = \|S\| = \|\Pi^1\| = \|\Pi^2\| & \triangleq & 1 \\
\|MN\| & \triangleq & \|M\| + \|N\| + 1 \\
\| \mathsf{m}\langle M\rangle \| & \triangleq & 3 + \|M\| \\
\|M \sim N\| & \triangleq & \|M\| + \|N\| + 4 \\
\|\mathsf{K}\langle M\rangle\| = \|\mathsf{S}\langle M\rangle\| & \triangleq & 4 + \|M\|
\end{array}
$$

It is easy to see that, for every reduction of the form $M \to N$ within the sub-TRS, $\|M\| > \|N\|$. $\square$

3.7.2 Translation from CL_P to $CL_\star$

We now will show that $CL_\star$ can code terms from CL_P (and therefore also from λP). In order to prove that the $W_\star$ reduction represents an abstraction mechanism, we will define translations between the two systems, and then present a simulation proof for a subset of CL_P which includes the image of the translation from λP.

Definition 3.7.5 First or all, we translate patterns from CL_P to $CL_\star$ via the transformation $[\bullet]$, defined as follows:

$$
\begin{array}{lcl}
[x] & \triangleq & \star \\
[K_P] & \triangleq & \mathsf{K}\langle [P] \rangle \\
[S_P] & \triangleq & \mathsf{S}\langle [P] \rangle \\
[MN] & \triangleq & [M][N] \\
[\Pi^1_{PQ}] & \triangleq & \mathsf{S}\langle [PQ] \rangle (\mathsf{K}\langle \star \rangle \Pi^1)(\mathsf{S}\langle \star \rangle \mathsf{K}\langle \star \rangle \star) \\
[\Pi^2_{PQ}] & \triangleq & \mathsf{S}\langle [PQ] \rangle (\mathsf{K}\langle \star \rangle \Pi^2)(\mathsf{S}\langle \star \rangle \mathsf{K}\langle \star \rangle \star)
\end{array}
$$

We are now ready to define the translation $\_^\star$ from CL_P-terms to $CL_\star$-terms:

$$
\begin{array}{lcl}
x^\star & \triangleq & x \\
(K_P)^\star & \triangleq & [K_P] \\
(S_P)^\star & \triangleq & [S_P] \\
(\Pi^1_{PQ})^\star & \triangleq & [\Pi^1_{PQ}] \\
(\Pi^2_{PQ})^\star & \triangleq & [\Pi^2_{PQ}] \\
(MN)^\star & \triangleq & M^\star N^\star
\end{array}
$$

We say that a term is **blocked** if it has a subterm of the form $\mathsf{m}\langle M\rangle N$ which may never be reduced in the root.

Remark 3.7.6 For all matching-safe CL_P-patterns P, Q: $[P] = [Q] \Leftrightarrow P =_\alpha Q$ (where $=$ denotes syntactic equality in $CL_\star$).

3.7.3 Relationship between $\rightarrow_{W_P}$ and $\rightarrow_{W_\star}$

In order to prove that abstractions in λP are also simulated by $\rightarrow_{W_\star}$, we will show simulation of the $\rightarrow_{W_P}$-reduction. Such a simulation does not hold for arbitrary CL_P-terms: for instance, the reduction $K_{K_S x K} y(K_S z K) \rightarrow_{W_P} y$ cannot be simulated in $CL_\star$. However, it does hold if we restrict the set of terms to those with matching-safe patterns (as defined in Section 3.6.1), and that is enough to simulate the λP abstraction. To this effect, we will use the following lemmas (see Appendix B.1 for their proofs), in which we assume that all CL_P-patterns are matching-safe.

Lemma 3.7.7 $[P] \sim [P] \twoheadrightarrow_{W_\star} I$

Lemma 3.7.8

1. $\mathsf{m}\langle [P] \rangle [P] \twoheadrightarrow_{W_\star} I$.
2. $\mathsf{m}\langle [P] \rangle (P^\sigma)^\star \twoheadrightarrow_{W_\star} I$.

Lemma 3.7.9

1. $\Pi^1((PQ)^\sigma)^\star \rightarrow_{W_\star} (P^\sigma)^\star$
2. $\Pi^2((PQ)^\sigma)^\star \rightarrow_{W_\star} (Q^\sigma)^\star$

We can now prove that $CL_\star$ simulates the reduction of CL_P restricted to terms with matching-safe patterns.

Proposition 3.7.10 (Simulation) If $M \rightarrow_{W_P} N$ then $M^\star \twoheadrightarrow_{W_\star} N^\star$

Corollary 3.7.11 $((\lambda P.M)P^\sigma)_{CL\star} \twoheadrightarrow_{W_\star} (M^\sigma)_{CL\star}$ whenever $(\lambda P.M)P^\sigma$ satisfies RPC^+.

Proof.- This is a consequence of Corollary 3.3.15 for CL_P, in conjunction with Proposition 3.7.10 and Lemma 3.7.8. Since P^σ matches P, Lemma 3.7.8 guarantees that the translation from λP to $CL_\star$ will behave as expected even with the optimizations for rules 3 and 4. □

3.7.4 Notes on Multiple Matching

While there are many possible uses for multiple matching, the definition of a calculus which combines the explicit control over the matching mechanism of $CL_\star$ with the multiple-matching facilities of a *case* construct is not simple. In Section 3.6.3 we defined a family of combinators of the form $\theta_{P_1,...,P_n}$ which allow the matching of multiple patterns in CL_P. However, these combinators may have complex structures, and their behaviour is determined by the implicit matching of a term against an arbitrarily large set of patterns (the rule schema for $\theta_{P_1,...,P_n} x_1 ... x_n P_i$ expands into n different rules for every instantiation of the patterns $P_1, ..., P_n$). For these reasons, this approach would not work on an extension of $CL_\star$, as the advantages of explicit matching would be lost and, moreover, we would no longer have a TRS with a finite set of rules. Thus, it would seem that the best alternative would be to test the matching against each pattern in a step-by-step manner.

It could be tempting to define a combinator that dealt with multiple matching, while keeping matching mechanism explicit. The matching would be performed step by step, one pattern at a

time. However, in the context of a TRS with explicit matching, we would lack a definite means to identify matching failure.

If we were to define a calculus in the spirit of $CL_\star$ with multiple matching capabilities, we would require a mechanism to detect whether a term matches a given pattern or not, and rules to determine the path to follow depending on the result. The latter could be solved by changing the matching validation rules so that matching a term against a pattern reduces to K in case of success, and KI in case of failure. For example, we could have a rule of the form $\mathsf{m}\langle\mathsf{K}\langle p\rangle q\rangle(\mathsf{K}\langle x\rangle y) \to p{\sim}x(\mathsf{m}\langle q\rangle yK(KI))(KI)$. This way, matching a term of the form $\mathsf{K}\langle M\rangle N$ against the pattern $\mathsf{K}\langle P\rangle Q$ would eventually reduce to K if $M =_{Wp} P$ and N matched Q, and KI otherwise. The problem lies, however, in detecting whether or not there is a match. Or, more precisely, on deciding when a certain term does *not* match a pattern.

Recall the $CL_P{+}\theta$-calculus from Section 3.6.3. First of all, note that some $CL_P{+}\theta$-reductions over open terms cannot be simulated by a calculus with explicit matching in the style of $CL_\star$. For instance, $\theta_{SKS,SxK}KS(SyK)$ is a well-formed $CL_P + \theta$-term, as the SKS and SxK are non-unifiable $CL_P + \theta$-patterns. Moreover, this term reduces in one step to $S(SyK)$, since SyK matches SxK (and, naturally, it does not match SKS). However, when translating this to a calculus which handles the matching in a step by step manner, we are forced to set an order in which the components of the patterns are analyzed. Assuming a left-to-right analysis, we would first need to contrast the translated form of SyK with the translation of SKS which, in turn, would require us to determine whether the variable y matches the translation of K. There is no way to determine this a priori by means of a TRS: since y is a variable, there will always be a substitution which causes it to match any given combinator, as well as a substitution that prevents the matching. Thus, we cannot define a rule to decide what to do when a variable needs to be matched against a non-variable pattern. The $CL_P + \theta$-calculus does not present this problem, since the matching is done in one single step, and no substitution can cause SyK to match SKS regardless of the term assigned to y (in other words, both terms are not unifiable). Analyzing the components of the terms and patterns in a different order would not make a difference: there will always be an ordering of the subindices of the θ combinator that will cause the variable in the argument to be matched against a non-variable term.

While the above problem only affects open terms, restricting our analysis to ground terms would not be enough to achieve simulation. Another problem arises when the argument of a θ combinator is not a normal form: if the argument does not match a certain pattern, there is no general way to foresee whether it will eventually reduce to a term which does. In $CL_P + \theta$, a reduction can take place whenever the argument of the θ combinator matches one of its subindices. This way, for example, the term $\theta_{SKK,SyS}\ K\ S(S(SII(SII))S)$ will reduce to $S(S(SII(SII))S)$, since the term $S(SII(SII))S$ matches the pattern SyS. However, if we attempt a step-by-step matching process, we will first need to test whether the term $S(SII(SII))S$ – which has no normal form – matches the pattern SKK. We cannot extend our TRS with a rule that can predict whether a reducible term will eventually match a given pattern; nor can we declare in advance that it does not match, since it could reduce to a term which does.

In sum, multiple matching and explicit matching are not expected to be combined within the context of a TRS.

3.8 Comparison between CL_P, $CL_\star$ and other Pattern Calculi

Motivations for using combinatory logic instead of a calculus with binders becomes now more evident. Specifically, the $CL_\star$-calculus has the following advantages when compared to CL_P:

- There is only a finite number of rules.
- Any $CL_\star$-term can be a pattern, however matching will only succeed for matching-safe patterns, and the patterns to be translated from λP or CL_P are still restricted.
- There is no need to match terms against rule schemas. Matching is now explicit: a matching operator is part of the syntax and is driven by appropriate reduction rules without the need for an external matching algorithm (other than the basic one needed to match terms with the left-hand-side of the rules, which is required for every TRS, including CL). A failed matching provides information on the cause of the failure. For example, reaching a normal form with a sub-term of the form $\mathsf{m}\langle p\rangle M$ will not only denote that the matching process has not succeeded (i.e. it has given a matching error), but it will also show us the precise conditions that failed.
- Reduction can also occur inside patterns, wherever necessary.
- α-conversion or equality up to renaming of variables is no longer needed. Having eliminated subscripts, there is no need to rename variables in them, and now term equality is strict.

On the other hand, CL_P has a number of advantages over $CL_\star$:

- Fewer reduction steps are required to reach a normal form.
- Multiple matching can be handled with the appropriate extension (see Section 3.6.3, and compare with the discussion in Section 3.7.4).
- Constructors can be introduced without the need to incorporate additional reduction rules (unlike $CL_\star$, which would need additional matching and equality rules to handle the new patterns).

There are several other calculi which can handle patterns in different ways. We will now briefly compare CL_P with some of these calculi.

The Pure Pattern Calculus ([Jay06, JK09]) is more complex in its formulation, requiring a concept of matchable forms with their own specific syntax. The matching mechanism is also handled by an external algorithm, defined as a sequence of clauses with complex conditions and a specific order of evaluations. The concept of variable is also less intuitive, as each variable x has a counterpart $\hat{x}$ and an associated constructor.

The logical frameworks defined in [Hon06] rely on external algorithms in order to determine the matching process. It generalizes the settings of some of the previous works. It contemplates the possibility of detecting whether the term to be reduced is open or ground.

While the need to match a given term against a pattern is made explicit in both Rho-calculus ([CK01]) and the ([WH08]), matching itself is resolved by an external algorithm; while

in CL_P the matching is handled by the TRS itself. Generalizations or variations to the matching process may be introduced by either adding or modifying the reduction rules, and the system will remain confluent as long as orthogonality is preserved. [WH08] also proves strong normalization of typable terms.

In all the above cases the matching, once formulated, is instantaneous. And the concept of bound variable is needed in all of the previous rewriting systems mentioned above.

The SF-calculus, mentioned in the introduction, is a combinatory calculus which can look into arguments much like in all known pattern calculi. However, instead of using patterns, it defines a concept of factorable form, which can be decomposed by a conditional combinator F. This combinator has the ability to distinguish between atoms and compound terms, allowing it to define a number of functions based on the structure of the arguments. On the other hand, CL_P is a TRS, with atomic combinators and no conditional rules, and the presence of projectors makes the decomposition of arguments in our calculus more intuitive. The SF-calculus could be mapped into a TRS by replacing the F rules by an infinite number of rules (one for each constant and one for each application pattern which can be matched by factorable forms, the latter being an infinite set). Additionally, [JGW11] does not present a mapping between the SF-calculus and a higher-order pattern calculus; nor does it present a simple type system for the calculus, since simple types have been shown to have insufficient expressive power to type the F and E combinators. Instead, a system with polymorphic types is introduced, with the drawback of type assignment being undecidable. [BR14] introduces a type system with a decidable type-assignment for a reduced version of the calculus named the $SFBY$-calculus, which emulates the E combinator via a term constructed with several hundred operators.

Chapter 4

Propositional Hypothetical Logic of Proofs

We propose a Natural Deduction presentation for the propositional Logic of Proofs and study proof normalisation. Section 4.1 introduces LP. It recalls the formulas, axioms, inference schemes and fundamental metatheoretical results. Section 4.2 introduces the Hypothetical Logic of Proofs or HLP. We then establish the precise correspondence between LP and HLP in Section 4.3. A term assignment for HLP is presented in Section 4.4. This is followed by a study of the fundamental properties of reduction for that term assignment, namely strong normalisation (Section 4.5) and confluence (Section 4.6). Section 4.7 considers some possible extensions and variants, including additional permutative rules, natural numbers and alternative inference rules with a more computationally-oriented flavour.

4.1 The Logic of Proofs

Recall from the Introduction that LP is a logic in which the usual propositional connectives are augmented by a new one: given a proof polynomial s and a proposition A, build $[\![s]\!]A$. The intended reading is: "*s is a proof of A*".

Definition 4.1.1 (Proof polynomials and Formulas) The set of proof polynomials and formulas of LP is defined as follows:

$$\begin{array}{rcl} s, t & ::= & x^A \mid c \mid s \cdot t \mid !s \mid s + t \\ A, B & ::= & P \mid \bot \mid A \supset B \mid [\![s]\!]A \end{array}$$

Proofs are represented by **proof polynomials**, which are constructed from **proof variables** and **proof constants** by means of functional symbols for elementary computable operations on proofs, binary "$\cdot$", "$+$", and unary "!". The "$\cdot$" is the traditional application operator. "$+$" stands for proof concatenation (or union of proofs). The "!" operator is known as **proof checker**, and may be understood as a means for checking the validity of a proof. Proof variables are used as hypotheses and range over a countably infinite set. The formulas of LP are built

by Boolean connectives from propositional atoms and those of the form $[\![s]\!]A$, where s is a proof polynomial and A is a formula. The operations of LP are specified by the following schemas (where A, B are arbitrary formulas and s, t are arbitrary proof polynomials).

Remark 4.1.2 One minor difference of our presentation of proof polynomials w.r.t. that of [Art95, Art01] is that proof variables are assumed to be decorated with a proposition. This entails no loss of generality since proof variables in LP are proposition-free. The benefit, however, is that when reasoning under a context of assumptions Γ, variables will serve to tag propositions and this will help in our translation from LP to HLP.

Definition 4.1.3 We recall from the Introduction the axiom and inference rules of LP:

A0.	Axioms of classical propositional logic in the language of LP
A1.	$[\![s]\!]A \supset A$
A2.	$[\![s]\!](A \supset B) \supset ([\![t]\!]A \supset [\![s \cdot t]\!]B)$
A3.	$[\![s]\!]A \supset [\![!s]\!][\![s]\!]A$
A4.	$[\![s]\!]A \supset [\![s+t]\!]A$
A5.	$[\![t]\!]A \supset [\![s+t]\!]A$
MP.	$\vdash A \supset B$ and $\vdash A \Rightarrow \vdash B$
Nec.	A an axiom $\mathbf{A0} - \mathbf{A5} \Rightarrow \vdash [\![c]\!]A$

For **A1** one reads:"*if s is a proof of A, then A holds*". For **A2** one reads: "*if s is a proof of $A \supset B$ and t is a proof of A, then $s \cdot t$ is a proof of B*". Thus "$\cdot$" represents composition of proofs. For **A3** one reads: "*if s is a proof of A, then $!s$ is a proof of the sentence* 's is a proof of A' ". Thus $!s$ is seen as a computation that verifies $[\![s]\!]A$. For plus one reads: "*if s is a proof of A, then so is $s+t$, regardless of the form of t*". We use letters π, π' for derivations in LP.

A **constant specification** ($\mathcal{CS}$) is a finite set of formulas $[\![c_1]\!]A_1, \ldots, [\![c_n]\!]A_n$ such that c_i is a constant, and A_i an axiom **A0**-**A5**. Each derivation in LP naturally generates the $\mathcal{CS}$ consisting of all formulas introduced in this derivation by the necessitation rule. A constant specification $\mathcal{CS}$ is **injective** if for each constant c there is at most one formula $[\![c]\!]A \in \mathcal{CS}$ (each constant denotes a proof of at most one axiom). We often distinguish in our notation the axiom to which a constant refers by putting the name of the axiom as superindex and the instances of its metavariables as its subindices. For example, we write $c^{\mathbf{A1}}_{x^P,P}$ for the constant corresponding to the instance $[\![x^P]\!]P \supset P$ of axiom **A1** (and similarly for the other axioms).

We often use deduction under the assumption of a **context** of hypotheses. A context Γ in LP is a set of hypotheses of proof variables s.t. if $x^{A_1} \in \Gamma$ and $x^{A_1} \in \Gamma$, then $A_1 = A_2$. The same formula may appear in a context more than once with different names (i.e. different variables). We occasionally write Γ, A for Γ, x^A with x a fresh name, and $A \in \Gamma$ means that there is some hypothesis x^A in Γ. We will use a, b, c, v, w, x, y, z, α as names for hypotheses in LP-contexts. Let $\Gamma = \{x_1^{A_1}, \ldots, x_n^{A_n}\}$ be a context and $\vec{u} = u_1, \ldots, u_n$ a list of proof polynomials, we define $[\![\vec{u}]\!]\Gamma$ as $\{x_1'^{[\![u_1]\!]A_1}, \ldots, x_n'^{[\![u_n]\!]A_n}\}$ with $x_1', \ldots, x_n'$ fresh names. Note that the variables $x_i^{A_i}$ and $x_i'^{[\![u_i]\!]A_i}$ have different types. Also, we write $\rhd_{\mathsf{LP}} \Gamma \vdash A$ to indicate that A is provable in LP under hypotheses Γ. We often say that a judgement $\Gamma \vdash A$ is *derivable* in LP meaning $\rhd_{\mathsf{LP}} \Gamma \vdash A$.

4.1.1 Metatheoretical Results

We will now establish some results about LP which will be used later (Section 4.3). The first is internalization, which requires the preliminary notion of *associated proof polynomial*.

Definition 4.1.4 (Associated proof polynomials) Let $[\![\vec{u}]\!]\Gamma \vdash D$ be a derivable judgement in LP, then r is an associated proof polynomial of D in $[\![\vec{u}]\!]\Gamma$ if one of the following conditions holds:

- D is of the form $A \supset B \supset A$, $(A \supset B \supset C) \supset (A \supset B) \supset A \supset C$ or $\neg\neg A \supset A$ and $r = c^{\mathbf{A0}}_{A,B}$, $c^{\mathbf{A0}}_{A,B,C}$ or $c^{\mathbf{A0}}_{A}$ respectively.
- D is of the form $[\![t]\!]A \supset A$ and $r = c^{\mathbf{A1}}_{t,A}$.
- D is of the form $[\![s]\!](A \supset B) \supset ([\![t]\!]A \supset [\![s \cdot t]\!]B)$ and $r = c^{\mathbf{A2}}_{s,t,A,B}$.
- D is of the form $[\![t]\!]A \supset [\![!t]\!][\![t]\!]A$ and $r = c^{\mathbf{A3}}_{t,A}$.
- D is of the form $[\![s]\!]A \supset [\![s+t]\!]A$ and $r = c^{\mathbf{A4}}_{s,t,A}$.
- D is of the form $[\![t]\!]A \supset [\![s+t]\!]A$ and $r = c^{\mathbf{A5}}_{s,t,A}$.
- D is of the form $[\![s]\!]A$ and $r = !s$.
- There is some A s.t. $[\![\vec{u}]\!]\Gamma \vdash A \supset D$ and $[\![\vec{u}]\!]\Gamma \vdash A$ are both derivable, and $r = s \cdot t$ where s and t are associated proof polynomials of $A \supset D$ and A in $[\![\vec{u}]\!]\Gamma$ respectively.

Lemma 4.1.5 (Internalization) If $\rhd_{\mathsf{LP}} [\![\vec{u}]\!]\Gamma \vdash D$, then:

1. There exists at least one associated proof polynomial of D in $[\![\vec{u}]\!]\Gamma$.
2. If r is an associated proof polynomial of D in $[\![\vec{u}]\!]\Gamma$, then $\rhd_{\mathsf{LP}} [\![\vec{u}]\!]\Gamma \vdash [\![r]\!]D$.

Proof.- Part 2 follows directly from the axioms and inference rules of LP. We prove part 1 by induction on the derivation of $[\![\vec{u}]\!]\Gamma \vdash D$, finding an associated proof polynomial r for each case.

- If there is some hypothesis $x^{[\![u]\!]A} \in [\![\vec{u}]\!]\Gamma$ and the derivation is obtained by using this hypothesis, then $D = [\![u]\!]A$ and $r = !u$ (note that $[\![\vec{u}]\!]\Gamma$ can only contain hypotheses of the form $x^{[\![u]\!]A}$ for some u and A).
- If the derivation is an instance of an axiom **Ai** with $i \in \{0, \cdots, 5\}$, then $r = c^{\mathbf{Ai}}_{\cdots}$ (with the corresponding arguments for D).
- If the derivation is obtained by using **Nec**, then D is of the form $[\![c^{\mathbf{Ai}}_{\cdots}]\!]A$ and $r = !c^{\mathbf{Ai}}_{\cdots}$.
- If the derivation is obtained by using **MP** from $[\![\vec{u}]\!]\Gamma \vdash A \supset D$ and $[\![\vec{u}]\!]\Gamma \vdash A$, then by IH there exist s and t associated proof polynomials of $A \supset D$ and A respectively. Take $r = s \cdot t$.

$\square$

Lemma 4.1.6 (Stripping) Suppose π is an LP-derivation of $\Gamma, x^{[\![y^A]\!]A} \vdash B$ and $y^A \notin \Gamma$. Then there is a derivation of $\Gamma, y^A \vdash B'$, where B' results from B, by replacing all occurrences of $[\![t]\!]A$ by A for every proof polynomial t containing y^A (including constants for instances of axioms containing y^A).

Proof.- By induction on π.

- If $B = [\![y^A]\!]A$ and π is obtained by using the hypothesis $x^{[\![y^A]\!]A}$, then $B' = A$ and π' is the derivation of $\Gamma, y^A \vdash A$ obtained by using the hypothesis y^A.
- If π is obtained by using a hypothesis $z^B \in \Gamma$, then there is a derivation of $\Gamma \vdash B$ which uses neither $x^{[\![y^A]\!]A}$ nor y^A. We obtain π' from this derivation by Weakening, and $B' = B$.
- If π is obtained by using an axiom **A0-A5**, there are three possibilities:
  - B has no proof polynomial containing y^A: then $B' = B$, and the derivation of $\Gamma \vdash B$ can be obtained by Weakening of the axiom.
  - B has one or more proof polynomials containing y^A, but B' is still an instance of the same axiom. Since axioms can be derived in any context, then $\rhd_{\mathsf{LP}} \Gamma, y^A \vdash B'$.
  - B has at least one proof polynomial containing y^A in a way that B' is no longer an instance of the same axiom as B: in this case, B' is an instance of one of the following schemes:
    1. $A \supset A$ from axioms **A1**, **A3**, **A4**, **A5**.
    2. $[\![s]\!]A \supset A$ from axioms **A4**, **A5**.
    3. $(A \supset B) \supset ([\![t]\!]A \supset B)$, 4. $[\![s]\!](A \supset B) \supset (A \supset B)$ or 5. $(A \supset B) \supset (A \supset B)$ from axiom **A2**.
    All these can be derived in LP in any context.
- If π is obtained by applying **MP**:

$$\dfrac{\dfrac{\cdots}{\Gamma, x^{[\![y^A]\!]A} \vdash C \supset B} \qquad \dfrac{\cdots}{\Gamma, x^{[\![y^A]\!]A} \vdash C}}{\Gamma, x^{[\![y^A]\!]A} \vdash B}\, MP$$

  By induction hypothesis, we have derivations of $\Gamma, y^A \vdash C' \supset B'$ and $\Gamma, y^A \vdash C'$. Therefore, by **MP**, we obtain a derivation of $\Gamma, y^A \vdash B'$.
- If π is obtained by applying **Nec**, then B is of the form $[\![c]\!]D$ with $c \in \mathcal{C}$ and D an instance of an axiom. If $y^A \in D$, then $B' = D'$ where $\Gamma, y^A \vdash D'$ is derivable by IH. Otherwise, $B' = B$.

□

Lemma 4.1.7 (λ-Abstraction) If $[\![\vec{u}]\!]\Gamma, y^{[\![x^A]\!]A} \vdash [\![s(\vec{u}, x^A)]\!]B$ and $x^A \notin \Gamma, B$, then there exists $t_\lambda^{A \supset B}([\![\vec{u}]\!]\Gamma)$ such that:

1. $[\![\vec{u}]\!]\Gamma \vdash [\![t_\lambda^{A \supset B}([\![\vec{u}]\!]\Gamma)]\!](A \supset B)$; and
2. $t_\lambda^{A \supset B}([\![\vec{u}]\!]\Gamma)$ is an associated proof polynomial of $A \supset B$ in $[\![\vec{u}]\!]\Gamma$.

Proof.- Note that w.l.o.g. we may assume that $x^A \in s(\vec{u}, x^A)$. Indeed, if this were not the case, then we could add it as follows:

$$\begin{array}{ll}
(a) & [\![\vec{u}]\!]\Gamma, y^{[\![x^A]\!]A} \vdash [\![c^{\mathbf{A0}}_{B,A}]\!](B \supset A \supset B) \\
(b) & [\![\vec{u}]\!]\Gamma, y^{[\![x^A]\!]A} \vdash [\![s(\vec{u}, x^A)]\!]B \\
(c) & [\![\vec{u}]\!]\Gamma, y^{[\![x^A]\!]A} \vdash [\![c^{\mathbf{A0}}_{B,A} \cdot s(\vec{u}, x^A)]\!](A \supset B) \\
(d) & [\![\vec{u}]\!]\Gamma, y^{[\![x^A]\!]A} \vdash [\![x^A]\!]A \\
(e) & [\![\vec{u}]\!]\Gamma, y^{[\![x^A]\!]A} \vdash [\![c^{\mathbf{A0}}_{B,A} \cdot s(\vec{u}, x^A) \cdot x^A]\!]B
\end{array}$$

We reason as follows:

$$\begin{array}{ll}
[\![\vec{u}]\!]\Gamma, y^{[\![x^A]\!]A} \vdash [\![s(\vec{u}, x^A)]\!]B & \text{(Hypothesis)} \\
[\![\vec{u}]\!]\Gamma, x^A \vdash B & \text{(Stripping and } x \in s(\vec{u}, x)\text{)} \\
[\![\vec{u}]\!]\Gamma \vdash A \supset B & \text{(Deduction for } \mathsf{LP}\text{)} \\
[\![\vec{u}]\!]\Gamma \vdash [\![t^{A \supset B}_{\lambda}([\![\vec{u}]\!]\Gamma)]\!](A \supset B) & \text{(Internalization for } \mathsf{LP}\text{)}
\end{array}$$

□

Corollary 4.1.8 (μ-Abstraction) If $[\![\vec{u}]\!]\Gamma, y^{[\![\alpha^{\neg A}]\!]\neg A} \vdash [\![s(\vec{u}, \alpha^{\neg A})]\!]\bot$ and $\alpha^{\neg A} \notin \Gamma$, let $t^A_{\mu}([\![\vec{u}]\!]\Gamma) = c^{\mathbf{A0}}_A \cdot t^{\neg\neg A}_{\lambda}([\![\vec{u}]\!]\Gamma)$, then $[\![\vec{u}]\!]\Gamma \vdash [\![t^A_{\mu}([\![\vec{u}]\!]\Gamma)]\!]A$ (where $c^{\mathbf{A0}}_A$ is a constant corresponding to the classical propositional logic axiom $\neg\neg A \supset A$).

Proof.- We reason as follows:

$$\begin{array}{ll}
[\![\vec{u}]\!]\Gamma \vdash [\![t^{\neg\neg A}_{\lambda}([\![\vec{u}]\!]\Gamma)]\!](\neg\neg A) & (\lambda\text{-Abstraction}) \\
[\![\vec{u}]\!]\Gamma \vdash [\![c^{\mathbf{A0}}_A]\!](\neg\neg A \supset A) & (\mathbf{A0} \text{ and } \mathbf{Nec}) \\
[\![\vec{u}]\!]\Gamma \vdash [\![c^{\mathbf{A0}}_A \cdot t^{\neg\neg A}_{\lambda}([\![\vec{u}]\!]\Gamma)]\!]A & (\mathbf{A2}\text{, and } \mathbf{MP} \text{ twice})
\end{array}$$

□

Lemma 4.1.9 If B is an instance of an LP-axiom, then $B\{x^A \leftarrow s\}$ is also an instance of the same axiom.

Proof.- By definition of the substitution and the axioms. □

Lemma 4.1.10 (Substitution) If $\Gamma \vdash [\![s]\!]A$ and $\Gamma, y^{[\![x^A]\!]A} \vdash B$ are derivable and $x^A \notin \Gamma$, then $\Gamma \vdash B\{x^A \leftarrow s\}$.

Proof.- By induction on the derivation of $\Gamma, y^{[\![x^A]\!]_{\Xi'}A} \vdash B$.

- If there is some hypothesis $z^B \in \Gamma$, $z^B \neq y^{[\![x^A]\!]_{\Xi'}A}$, and the derivation is obtained by using this hypothesis, then, by variable convention, $x^A \notin \mathsf{FV}(B)$ and $y^{[\![x^A]\!]_{\Xi'}A} \notin \mathsf{FV}(B)$. Therefore $B\{x^A \leftarrow s\} = B$, and $\Gamma \vdash B$ is derivable by Strengthening.
- If $B = [\![x^A]\!]_{\Xi'}A$ and the derivation is obtained by using the hypothesis $y^{[\![x^A]\!]_{\Xi'}A}$, then $B\{x^A \leftarrow s\} = [\![s]\!]A$.

- If the derivation is an instance of an axiom, then $B\{x^A \leftarrow s\}$ is also an instance of the same axiom by Lemma 4.1.9.
- If the derivation is obtained by using **MP** from $\Gamma \vdash C \supset B$ and $\Gamma \vdash C$, then by IH $\Gamma \vdash C\{x^A \leftarrow s\} \supset B\{x^A \leftarrow s\}$ and $\Gamma \vdash C\{x^A \leftarrow s\}$ are both derivable. We obtain the result by **MP**.
- If the derivation is obtained by using **Nec**, then B is of the form $[\![c]\!]C$ with C an instance of an axiom. Since $C\{x^A \leftarrow s\}$ is also an instance of the same axiom (by Lemma 4.1.9), then $\Gamma \vdash [\![c]\!]C\{x^A \leftarrow s\}$ is derivable by **Nec**.

□

4.1.2 Models for the Logic of Proofs

This section briefly revisits two abstract (i.e. non-arithmetic) models of LP, namely Mkrtychev and Kripke models, with the aim of providing a firmer grip on the logic itself. This thesis does not deal with models of LP.

Mkrtychev models. The first abstract model for LP was introduced by Mkrtychev [Mkr97]. Given a constant specification $\mathcal{CS}$ with one constant for each LP-axiom, a **Mkrtychev model** for LP corresponding to $\mathcal{CS}$ is a pair $\langle *, \Vdash \rangle$, where $*$ is a function – called a **witness function** – which maps a proof polynomial t into a set of formulas which accept t as their witness, and $\Vdash$ is a truth assignment for formulas. The witness function $*$ has the following properties:

- If $[\![c]\!]A \in \mathcal{CS}$ then $A \in *(c)$.
- If $A \supset B \in *(s)$ and $A \in *(t)$ then $B \in *(s \cdot t)$.
- If $A \in *(s)$ then $[\![s]\!]A \in *(!s)$.
- $*(s) \cup *(t) \subseteq *(s + t)$.

The truth assignment $\Vdash$ is defined by arbitrarily assigning truth values to propositional variables, and distributing it inductively by means of the Boolean laws and the following condition:

$$\Vdash [\![t]\!]A \;\Leftrightarrow\; A \in *(t) \text{ and } \Vdash A$$

Soundness and completeness of LP with respect to Mkrtychev models were established in [Mkr97]. Despite their simplicity, they have been used as a tool for proving various properties of LP. One is decidability [Mkr97]. See [AB05] for other applications including bounds on satisfiability and the disjunction property.

Kripke models. Kripke semantics was developed by Fitting [Fit05a]. Given a *frame* $\langle \mathcal{G}, \mathcal{R} \rangle$, where $\mathcal{G}$ is a non-empty set of states or possible worlds, and $\mathcal{R}$ is a binary accessibility relation on $\mathcal{G}$, a possible evidence function ε is a mapping from states and proof polynomials to sets of formulas. "$A \in \varepsilon(\Gamma, t)$" can be read as "$t$ serves as possible evidence for A in the world Γ".

An evidence function must obey certain conditions which respect the intended meanings of the operations on proof polynomials (all except monotonicity have their roots in [Mkr97]).

- Application: if $A \supset B \in \varepsilon(\Gamma, s)$ and $A \in \varepsilon(\Gamma, t)$, then $B \in \varepsilon(\Gamma, s \cdot t)$.
- Monotonicity: if $\Gamma \mathcal{R} \Gamma'$, then $\varepsilon(\Gamma, t) \subseteq \varepsilon(\Gamma', t)$.
- Proof Checker: if $A \in \varepsilon(\Gamma, t)$, then $[\![t]\!]A \in \varepsilon(\Gamma, !t)$.
- Sum: $\varepsilon(\Gamma, s) \cup \varepsilon(\Gamma, t) \subseteq \varepsilon(\Gamma, s + t)$.

As usual in Kripke semantics, truth of atomic formulas at possible worlds is specied arbitrarily.

A structure $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, \mathcal{E}, \mathcal{V} \rangle$ is a **weak LP-model** if $\langle \mathcal{G}, \mathcal{R} \rangle$ is a frame with $\mathcal{R}$ reflexive and transitive, $\mathcal{E}$ is an evidence function on $\langle \mathcal{G}, \mathcal{R} \rangle$, and $\mathcal{V}$ is a mapping from propositional variables to subsets of $\mathcal{G}$. Given a weak model $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, \mathcal{E}, \mathcal{V} \rangle$, a forcing relation is defined by the following rules. For each $\Gamma \in \mathcal{G}$:

- $\Gamma \Vdash P$ for a propositional variable P if $\Gamma \in \mathcal{V}(P)$.
- $\Gamma \nVdash \bot$.
- $\Gamma \Vdash A \supset B$ if and only if $\Gamma \nVdash A$ or $\Gamma \Vdash B$.
- $\Gamma \Vdash [\![t]\!]A$ if and only if $A \in \varepsilon(\Gamma, t)$ and, for every $\Gamma' \in \mathcal{G}$ s.t. $\Gamma \mathcal{R} \Gamma'$, $\Gamma \Vdash A$.

We say A is *true* at world Γ if $\Gamma \Vdash A$, and otherwise A is *false* at Γ.

$\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, \mathcal{E}, \mathcal{V} \rangle$ is a **strong LP-model** if it is a weak LP-model, and whenever $\Gamma' \Vdash A$ for every $\Gamma' \in \mathcal{G}$ such that $\Gamma \mathcal{R} \Gamma'$, then there is proof polynomial t such that $\Gamma \Vdash [\![t]\!]A$.

A **signed formula** is $T\ A$ or $F\ A$, where A is a formula of LP. Intuitively these can be read: A is true, respectively false, in some particular context. Given a constant specification $\mathcal{CS}$, a set S of signed formulas is weakly (resp. strongly) $\mathcal{CS}$ – **LP satisfiable** if there is a weak (resp. strong) LP-model $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, \mathcal{E}, \mathcal{V} \rangle$ that meets $\mathcal{CS}$, and a possible world $\Gamma \in \mathcal{G}$ at which all T-signed members of S are true and all F-signed members are false; that is, $\mathcal{M}, \Gamma \Vdash A$ for every formula A s.t. $T\ A \in S$, and $\mathcal{M}, \Gamma \nVdash A$ for every formula A s.t. $F\ A \in S$.

A formula A is weakly (resp. strongly) $\mathcal{CS}$-**LP valid** if $T\ A$ is satisfied by every weak (resp. strong) LP-model.

Fitting proved soundness and completeness of LP with respect to both weak LP-models and strong LP-models:

Theorem (Soundness). If A is derivable in LP with constant specification $\mathcal{CS}$, then A is weakly and strongly $\mathcal{CS}$-LP valid.

Theorem (Completeness). If $\mathcal{CS}$ is a constant specification, every formula that is weakly or strongly $\mathcal{CS}$-LP valid is derivable in LP.

4.2 Hypothetical Logic of Proofs

A **formula** in HLP is the same as in LP, except that $r, s, \ldots$ range over proof witnesses rather than proof polynomials.

Definition 4.2.1 (Proof witness) A **proof witness** is an expression generated by the following grammar:

$$
\begin{array}{rcll}
r, s, t & ::= & x^A & \text{truth hypothesis} \\
 & | & v^A & \text{validity hypothesis} \\
 & | & \lambda x^A.s & \text{abstraction} \\
 & | & s \cdot t & \text{application} \\
 & | & !s & \text{bang} \\
 & | & t\langle v^A{:=}r, s\rangle & \text{unbox} \\
 & | & s + t & \text{plus} \\
 & | & [\alpha^A]s & \text{name application} \\
 & | & \mu\alpha^A.s & \text{name abstraction}
\end{array}
$$

In $\lambda x^A.s$, the scope of the bound variable x^A is s; in $t\langle v^A{:=}r, s\rangle$ the scope of the bound variable v^A is t and in $\mu\alpha^A.s$ the scope of the bound variable α^A is s. Also, $!s$ binds all free occurrences of truth and falsehood variables in s; likewise $t\langle v^A{:=}r, s\rangle$ binds all free occurrences of truth and falsehood variables in r. Abstractions, applications, name abstractions and name applications are similar to their $\lambda\mu$ analogues (see Section 2.5.4), with the difference that here we are using them as proof witnesses rather than terms. The "!" and "+" operators are similar to those of LP. Regarding proof witnesses of the form $t\langle v^A{:=}r, s\rangle$ they can be read as "replace all free occurrences of v^A by r in the formula witnessed by t, with s bearing witness to the truth of $[\![r]\!]A$".

We assume the following **variable convention**: all bound variable names are different from each other, and different from all free variables. We also assume that application "$\cdot$" and sum "+" are left-associative, and implication "$\supset$" is right-associative. We use "$\neg A$" as an abbreviation for "$A \supset \bot$". The operators "!", "$\neg$" and "$[\![\ ]\!]$" have precedence over "$\cdot$", "+" and "$\supset$", which in turn have precedence over "λ", "μ" and "[ ]". For example, $[\alpha^{([\![r]\!]A)\supset((\neg B)\supset C)}]((!s) + t)$ may be written $[\alpha^{[\![r]\!]A\supset\neg B\supset C}]!s + t$.

Definition 4.2.2 (Contexts and judgements for HLP) A **truth context** (Γ) is a set of truth hypotheses $\{x_1^{A_1}, \ldots, x_n^{A_n}\}$; a **validity context** (Θ) is a set of validity variables $\{v_1^{A_1}, \ldots, v_m^{A_m}\}$; a **falsehood context** (Δ) is a set of falsehood variables $\{\alpha_1^{A_1}, \ldots, \alpha_k^{A_k}\}$. We write $\cdot$ for the empty context. A **judgement** is an expression of the form:

$$\Theta; \Gamma; \Delta \vdash A \,|\, s$$

It will often be convenient to abbreviate the expression $\Theta; \Gamma; \Delta$ in order to improve readability. We will use $\mathcal{H}$ for this purpose and refer to $\mathcal{H}$ as a *composite context*. So the above judgement will also be written:

$$\mathcal{H} \vdash A \,|\, s \tag{4.1}$$

Also we write:

$$
\begin{array}{l}
\mathcal{H}, x^A \text{ for } \Theta; \Gamma, x^A; \Delta \\
\mathcal{H}, \alpha^A \text{ for } \Theta; \Gamma; \Delta, \alpha^A \\
\mathcal{H}, v^A \text{ for } \Theta, v^A; \Gamma; \Delta
\end{array}
$$

Finally, we write $\rhd_{\mathsf{HLP}} \mathcal{H} \vdash A \,|\, s$, to indicate that the judgement $\mathcal{H} \vdash A \,|\, s$ is derivable in HLP.

The set of **free variables of validity, truth and falsehood** in a formula A are denoted $\mathsf{FVT}(A)$, $\mathsf{FVV}(A)$ and $\mathsf{FVF}(A)$, resp. The definition of $\mathsf{FVT}(A)$ is as follows ($\mathsf{FVV}(A)$ and $\mathsf{FVF}(A)$ are similar and hence omitted), where $\mathsf{FVT}(A, B)$ abbreviates $\mathsf{FVT}(A) \cup \mathsf{FVT}(B)$:

$$\begin{array}{rcl}
\mathsf{FVT}(P) & \triangleq & \emptyset \\
\mathsf{FVT}(\bot) & \triangleq & \emptyset \\
\mathsf{FVT}(A \supset B) & \triangleq & \mathsf{FVT}(A, B) \\
\mathsf{FVT}([\![t]\!]A) & \triangleq & \mathsf{FVT}(t) \cup \mathsf{FVT}(A)
\end{array}$$

The set of free variables of validity, truth and falsehood in a proof witness s, denoted $\mathsf{FVT}(s)$, $\mathsf{FVV}(s)$ and $\mathsf{FVF}(s)$, resp., are defined as follows:

$$\begin{array}{rcl@{\qquad}rcl}
\mathsf{FVT}(x^A) & \triangleq & \{x^A\} & \mathsf{FVV}(x^A) & \triangleq & \emptyset \\
\mathsf{FVT}(v^A) & \triangleq & \emptyset & \mathsf{FVV}(v^A) & \triangleq & \{v^A\} \\
\mathsf{FVT}(\lambda x^A.s) & \triangleq & \mathsf{FVT}(s) \backslash \{x^A\} & \mathsf{FVV}(\lambda x^A.s) & \triangleq & \mathsf{FVV}(s) \\
\mathsf{FVT}(s \cdot t) & \triangleq & \mathsf{FVT}(s, t) & \mathsf{FVV}(s \cdot t) & \triangleq & \mathsf{FVV}(s, t) \\
\mathsf{FVT}(!s) & \triangleq & \emptyset & \mathsf{FVV}(!s) & \triangleq & \mathsf{FVV}(s) \\
\mathsf{FVT}(t\langle v^A{:=}r, s\rangle) & \triangleq & \mathsf{FVT}(t, s) & \mathsf{FVV}(t\langle v^A{:=}r, s\rangle) & \triangleq & (\mathsf{FVV}(t) \backslash \{v^A\}) \cup \mathsf{FVV}(r, s) \\
\mathsf{FVT}(s + t) & \triangleq & \mathsf{FVT}(s, t) & \mathsf{FVV}(s + t) & \triangleq & \mathsf{FVV}(s, t) \\
\mathsf{FVT}([\alpha^A]s^A) & \triangleq & \mathsf{FVT}(s^A) & \mathsf{FVV}([\alpha^A]s^A) & \triangleq & \mathsf{FVV}(s^A) \\
\mathsf{FVT}(\mu\alpha^A.s) & \triangleq & \mathsf{FVT}(s^A) & \mathsf{FVV}(\mu\alpha^A.s) & \triangleq & \mathsf{FVV}(s^A)
\end{array}$$

$$\begin{array}{rcl}
\mathsf{FVF}(x^A) & \triangleq & \emptyset \\
\mathsf{FVF}(v^A) & \triangleq & \emptyset \\
\mathsf{FVF}(\lambda x^A.s) & \triangleq & \mathsf{FVF}(s) \\
\mathsf{FVF}(s \cdot t) & \triangleq & \mathsf{FVF}(s, t) \\
\mathsf{FVF}(!s) & \triangleq & \emptyset \\
\mathsf{FVF}(t\langle v^A{:=}r, s\rangle) & \triangleq & \mathsf{FVF}(t, s) \\
\mathsf{FVF}(s + t) & \triangleq & \mathsf{FVF}(s, t) \\
\mathsf{FVF}([\alpha^A]s^A) & \triangleq & \mathsf{FVF}(s^A) \cup \{\alpha^A\} \\
\mathsf{FVF}(\mu\alpha^A.s) & \triangleq & \mathsf{FVF}(s^A) \backslash \{\alpha^A\}
\end{array}$$

Remark 4.2.3 In a judgement $\Theta; \Gamma; \Delta \vdash A \,|\, s$ we shall assume the following **freshness condition**: all v_i with $i \in 1..m$, x_i with $i \in 1..n$ and α_i with $i \in 1..k$ are assumed distinct and moreover *fresh* (i.e. that they do not occur in the A_i, B_i and C_i). More precisely, for every pair of formulas A, B such that $x^A \in \Gamma$, $v^A \in \Theta$ or $\alpha^A \in \Delta$:

- if $y^B \in \Gamma$, then $y^B \notin \mathsf{FVT}(A)$;
- if $w^B \in \Theta$, then $w^B \notin \mathsf{FVV}(A)$; and
- if $\beta^B \in \Delta$, then $\beta^B \notin \mathsf{FVF}(A)$.

$$\frac{}{\mathcal{H}, x^A \vdash A \,|\, x^A}\ \mathsf{Var}$$

$$\frac{\mathcal{H}, x^A \vdash B \,|\, s}{\mathcal{H} \vdash A \supset B \,|\, \lambda x^A.s}\ \supset\mathsf{I} \qquad \frac{\mathcal{H} \vdash A \supset B \,|\, s \quad \mathcal{H} \vdash A \,|\, t}{\mathcal{H} \vdash B \,|\, s \cdot t}\ \supset\mathsf{E}$$

$$\frac{}{\mathcal{H}, v^A \vdash A \,|\, v^A}\ \mathsf{VarM}$$

$$\frac{\Theta; \cdot; \cdot \vdash B \,|\, s \quad \Theta; \cdot; \cdot \vdash s \equiv t : B}{\Theta; \Gamma; \Delta \vdash [\![t]\!]B \,|\, !t}\ \Box\mathsf{I} \qquad \frac{\mathcal{H} \vdash [\![r]\!]A \,|\, s \quad \mathcal{H}, v^A \vdash C \,|\, t}{\mathcal{H} \vdash C\{v^A \leftarrow r\} \,|\, t\langle v^A := r, s\rangle}\ \Box\mathsf{E}$$

$$\frac{\mathcal{H} \vdash A \,|\, s}{\mathcal{H} \vdash A \,|\, s + t}\ \mathsf{PlusL} \qquad \frac{\mathcal{H} \vdash A \,|\, t}{\mathcal{H} \vdash A \,|\, s + t}\ \mathsf{PlusR}$$

$$\frac{\mathcal{H}, \alpha^A \vdash A \,|\, s}{\mathcal{H}, \alpha^A \vdash \bot \,|\, [\alpha^A]s}\ \mathsf{Name} \qquad \frac{\mathcal{H}, \alpha^A \vdash \bot \,|\, s}{\mathcal{H} \vdash A \,|\, \mu\alpha^A.s}\ \mathsf{NAbs}$$

Figure 4.1: Axiom and inference schemes of HLP.

Inference schemes. The axiom and inference schemes of HLP are depicted in Fig. 4.1. We write $\rhd_{\mathsf{HLP}}\, \Theta; \Gamma; \Delta \vdash A \,|\, s$ to indicate that the judgement $\Theta; \Gamma; \Delta \vdash A \,|\, s$ is derivable using these schemes. An informal explanation of these schemes follows.

The axiom scheme Var states that the judgement $\Theta; \Gamma, x^A; \Delta \vdash A \,|\, x^A$ is evident in itself. Indeed, if we assume that x^A is a witness that proposition A is true, then we immediately conclude that A is true with proof witness x^A.

The introduction scheme for the $[\![t]\!]$ modality internalises meta-level evidence into the object logic. It states that if s is unconditional evidence that A is true, then A is in fact valid with proof witness s, or more generally, any proof witness t equivalent to s (in a sense to be made precise shortly, cf. Sec. 4.2.1). Evidence for the truth of $[\![t]\!]A$ is constructed from the (verified) evidence that A is unconditionally true by prefixing it with a bang constructor.

Remark 4.2.4 As discussed in [AB07], the following naive introduction scheme for $\Box$ would yield a system in which Type Preservation fails (see Section 4.2.1 for further details):

$$\frac{\Theta; \cdot; \cdot \vdash B \,|\, s}{\Theta; \Gamma; \Delta \vdash [\![s]\!]B \,|\, !s}\ \Box\mathsf{I}'$$

The $\Box$E scheme allows the discharging of validity hypotheses. In order to discharge the validity hypothesis v^A, a proof of the validity of A is required. In this system, this requires proving that $[\![r]\!]A$ is true with proof witness s, for some proof witnesses r and s. Note that r is a witness that A is unconditionally true (i.e. valid) whereas s is witness to the truth of $[\![r]\!]A$. The former is then substituted in the place of all free occurrences of v in the proposition C.

This construction is recorded with proof witness $t\langle v^A{:=}r, s\rangle$ in the conclusion, meaning that s is proof that r can be used in place of v^A in t. This has the practical effect of allowing us to take the witness r out of the box from $[\![r]\!]A$. The expression $C\{v^A{\leftarrow}\ r\}$ denotes the substitution of v^A by r in C. This and other forms of substitution will be explained in detail in Sec. 4.4.1. A final remark on $\Box\mathsf{E}$, its witness includes s since this is required for the proof that derivable HLP formulas are also derivable in LP (Sec. 4.3) and also for Type Preservation (see validity variable substitution and its use in the reduction rule γ in Def. 4.4.11).

Regarding the schemes for plus we comment on PlusL, the case of PlusR being similar. Informally, the proof witness $s+t$ testifies that either s or t is witness to the truth of A *without* supplying details on which of the two. Note that t is any proof witness whatsoever. Indeed, it may even contain variables not included in Θ, Γ nor Δ (see also Example 4.2.7). The reason is that we seek to preserve the *theorems* of LP in HLP, in particular $[\![s]\!]A \supset [\![s+t]\!]A$, which places no restriction on t. For further discussion see Remark 4.3.3.

Finally, the schemes Name and NAbs were introduced by Parigot in the $\lambda\mu$-calculus, as discussed in Section 2.5.3. The intuition behind them is that hypotheses in Δ must be considered negated. Under this reading, the scheme Name states that from $\neg A$ and A we may deduce $\bot$. Likewise, the scheme NAbs is the classical negation rule stating that if we arrive at a contradiction from the hypothesis $\neg A$, then we may discharge this hypothesis and obtain A.

Some sample derivations follow.

Example 4.2.5 We prove $\cdot;\cdot;\cdot \vdash [\![s]\!]A \supset [\![!s]\!][\![s]\!]A \mid \lambda x^{[\![s]\!]A}.!!v^A\langle v^A{:=}s, x^{[\![s]\!]A}\rangle$.

$$
\dfrac{\dfrac{\dfrac{}{\cdot; x^{[\![s]\!]A}; \cdot \vdash [\![s]\!]A \mid x^{[\![s]\!]A}}\,\mathsf{Var} \qquad \dfrac{\dfrac{\dfrac{}{v^A;\cdot;\cdot \vdash A \mid v^A}\,\mathsf{VarM}}{v^A;\cdot;\cdot \vdash [\![v^A]\!]A \mid !v^A}\,\Box\mathsf{I}}{v^A; x^{[\![s]\!]A};\cdot \vdash [\![!v^A]\!][\![v^A]\!]A \mid !!v^A}\,\Box\mathsf{I}}{\cdot; x^{[\![s]\!]A};\cdot \vdash [\![!s]\!][\![s]\!]A \mid !!v^A\langle v^A{:=}s, x^{[\![s]\!]A}\rangle}\,\Box\mathsf{E}}{\cdot;\cdot;\cdot \vdash [\![s]\!]A \supset [\![!s]\!][\![s]\!]A \mid \lambda x^{[\![s]\!]A}.!!v^A\langle v^A{:=}s, x^{[\![s]\!]A}\rangle}\,\supset\mathsf{I}
$$

Example 4.2.6 The following judgement is easily seen to be derivable: $\cdot;\cdot;\cdot \vdash [\![x^A \cdot x^A]\!]B \supset [\![x^A{\cdot}x^A]\!]B \mid \lambda y^{[\![x^A\cdot x^A]\!]B}.y$. Note that the proof witness $x^A{\cdot}x^A$ does not denote any valid derivation in HLP. In fact, for any proof witness s, $\rhd_{\mathsf{HLP}} \cdot;\cdot;\cdot \vdash [\![s]\!]B \supset [\![s]\!]B \mid \lambda y^{[\![s]\!]B}.y$.

The following example illustrates the "+" operator, for which it is convenient to extend HLP with conjunction and disjunction (Fig. 4.2).

Example 4.2.7 We seek to prove the formula $[\![s]\!]A{\vee}[\![t]\!]B \supset [\![\mathsf{inl}(s)+\mathsf{inr}(t)]\!](A{\vee}B)$. For that, let $\Theta_1 \triangleq v^A$, $\Theta_2 \triangleq u^B$, $\Gamma \triangleq z^{[\![s]\!]A\vee[\![t]\!]B}$ $\Gamma_1 \triangleq \Gamma, x^{[\![s]\!]A}$ and $\Gamma_2 \triangleq \Gamma, y^{[\![t]\!]B}$ in the derivations $\pi_{1,2}$:

$$\frac{\mathcal{H} \vdash A \,|\, s \quad \mathcal{H} \vdash B \,|\, t}{\mathcal{H} \vdash A \wedge B \,|\, \mathsf{pair}(s,t)}\ \wedge_{\mathsf{I}}$$

$$\frac{\mathcal{H} \vdash A \wedge B \,|\, s}{\mathcal{H} \vdash A \,|\, \mathsf{fst}(s)}\ \wedge_{\mathsf{E1}} \quad \frac{\mathcal{H} \vdash A \wedge B \,|\, s}{\mathcal{H} \vdash B \,|\, \mathsf{snd}(s)}\ \wedge_{\mathsf{E2}}$$

$$\frac{\mathcal{H} \vdash A \,|\, s}{\mathcal{H} \vdash A \vee B \,|\, \mathsf{inl}(s)}\ \vee_{\mathsf{I1}} \quad \frac{\mathcal{H} \vdash B \,|\, s}{\mathcal{H} \vdash A \vee B \,|\, \mathsf{inr}(s)}\ \vee_{\mathsf{I2}}$$

$$\frac{\mathcal{H} \vdash A \vee B \,|\, r \quad \mathcal{H}, x^A \vdash C \,|\, s \quad \mathcal{H}, y^B \vdash C \,|\, t}{\mathcal{H} \vdash C \,|\, \mathsf{case}\, r\, x^A.s\, y^B.t}\ \vee_{\mathsf{E}}$$

Figure 4.2: Inference schemes for conjunction and disjunction

$$\frac{\cdot;\Gamma_1;\cdot \vdash [\![s]\!]A \,|\, x^{[\![s]\!]A} \qquad \dfrac{\dfrac{\dfrac{\Theta_1;\cdot;\cdot \vdash A \,|\, v^A}{\Theta_1;\cdot;\cdot \vdash A \vee B \,|\, \mathsf{inl}(v^A)}\ \vee_{\mathsf{I1}}}{\Theta_1;\cdot;\cdot \vdash A \vee B \,|\, \mathsf{inl}(v^A) + \mathsf{inr}(t)}\ \mathsf{PlusL}}{\Theta_1;\Gamma_1;\cdot \vdash [\![\mathsf{inl}(v^A) + \mathsf{inr}(t)]\!](A \vee B) \,|\, !(\mathsf{inl}(v^A) + \mathsf{inr}(t))}\ \Box\mathsf{I}}{\cdot;\Gamma_1;\cdot \vdash [\![\mathsf{inl}(s) + \mathsf{inr}(t)]\!](A \vee B) \,|\, !(\mathsf{inl}(v^A) + \mathsf{inr}(t))\langle v^A := s, x^{[\![s]\!]A}\rangle}\ \Box\mathsf{E}$$

$$\frac{\cdot;\Gamma_2;\cdot \vdash [\![t]\!]B \,|\, y^{[\![s]\!]B} \qquad \dfrac{\dfrac{\dfrac{\Theta_2;\cdot;\cdot \vdash B \,|\, u^B}{\Theta_2;\cdot;\cdot \vdash A \vee B \,|\, \mathsf{inr}(u^B)}\ \vee_{\mathsf{I2}}}{\Theta_2;\cdot;\cdot \vdash A \vee B \,|\, \mathsf{inl}(s) + \mathsf{inr}(u^B)}\ \mathsf{PlusR}}{\Theta_2;\Gamma_2;\cdot \vdash [\![\mathsf{inl}(s) + \mathsf{inr}(u^B)]\!](A \vee B) \,|\, !(\mathsf{inl}(s) + \mathsf{inr}(u^B))}\ \Box\mathsf{I}}{\cdot;\Gamma_2;\cdot \vdash [\![\mathsf{inl}(s) + \mathsf{inr}(t)]\!](A \vee B) \,|\, !(\mathsf{inl}(s) + \mathsf{inr}(u^B))\langle u^B := t, y^{[\![s]\!]B}\rangle}\ \Box\mathsf{E}$$

Finally, we have:

$$\frac{\cdot;\Gamma;\cdot \vdash [\![s]\!]A \vee [\![t]\!]B \,|\, z^{[\![s]\!]A \vee [\![t]\!]B} \qquad \pi_1 \qquad \pi_2}{\cdot;\Gamma;\cdot \vdash [\![\mathsf{inl}(s) + \mathsf{inr}(t)]\!](A \vee B) \,|\, \mathsf{case}\, z^{[\![s]\!]A \vee [\![t]\!]B}\, x^A.r_1\, y^B.r_2}\ \vee_{\mathsf{E}}$$

Note that the use of PlusL in π_1 and PlusR in π_2 is required in order to concatenate the two alternative proofs of $A \vee B$ into a unique proof, and allow the application of $\vee_{\mathsf{E}}$ in π_3.

Remark 4.2.8 One may wonder whether, for the implicational fragment, the "+" may be dispensed with while still maintaining realization of all $\mathsf{S4}$ theorems. This is the case if non-injective constant specification sets and non-normal realizations are allowed (see [Kuz09] and also [AB05, Sec.11.2]).

Artemov's proof of the Realization Theorem [Art95, Art01] is constructive, and uses cut-free Gentzen-style derivations. The "+" operator is used in the cases where $\Box$ is introduced on the

right side of a sequent. Kuznets showed that the proof polynomial used for the realization of that $\Box$ can be replaced by one that is "+"-free, by making use of the Lifting Lemma:

Lemma 4.2.9 If $\rhd_{\mathsf{LP}} G_1, \ldots, G_n, [\![s_1]\!]H_1, \ldots, [\![s_k]\!]H_k \vdash F$, then it is possible to construct a "+"-free proof polynomial $t(x_1, \ldots, x_n, y_1, \ldots, y_k)$ such that:
$\rhd_{\mathsf{LP}} [\![x_1]\!]G_1, \ldots, [\![x_n]\!]G_n, [\![s_1]\!]H_1, \ldots, [\![s_k]\!]H_k \vdash [\![t(x_1, \ldots, x_n, s_1, \ldots, s_k)]\!]F$.

However, the "+"-free realization thus obtained is not *normal* according to Artemov's definition of *normality*, because all its negative occurrences of "$\Box$" are realized by the same proof variable rather than many distinct ones. It has been shown that there are theorems of modal logic $\mathsf{S4}$ which do not have normal "+"–free realizations in LP.

We end the section with some basic metatheoretical results.

Lemma 4.2.10 (Weakening, Strengthening) If $\rhd_{\mathsf{HLP}} \Theta; \Gamma; \Delta \vdash A \,|\, s$, then:

1. $\rhd_{\mathsf{HLP}} \Theta \cup \Theta'; \Gamma \cup \Gamma'; \Delta \cup \Delta' \vdash A \,|\, s$; and
2. $\rhd_{\mathsf{HLP}} \Theta \cap \mathsf{FVV}(s); \Gamma \cap \mathsf{FVT}(s); \Delta \cap \mathsf{FVF}(s) \vdash A \,|\, s$.

Proof.- By induction on the derivation. 1 is straightforward. For 2, in the $\Box\mathsf{I}$ case we can either assume that $s = t$ (which, as we shall see, can be assumed safely by Remark 4.2.17) and use the induction hypothesis, or prove this result and Strengthening of proof witness equivalence (Lemma 4.2.16) by mutual induction. $\Box$

The following result, which is proved by induction on the derivation of $\Gamma \vdash \Delta; A$, entails that all classical tautologies are derivable in HLP:

Proposition 4.2.11 If $\Gamma \vdash \Delta; A$ is derivable in the $\lambda\mu$-calculus, then $\rhd_{\mathsf{HLP}} \cdot; \Gamma; \Delta \vdash A \,|\, s$, for some proof witness s.

Corollary 4.2.12 If A is a classical tautology, then $\rhd_{\mathsf{HLP}} \cdot; \cdot; \cdot \vdash A \,|\, s$ for some proof witness s.

4.2.1 Proof Witness Equivalence

The $\Box\mathsf{I}$ inference scheme we have presented (Fig. 4.1) resorts to a notion of proof equivalence in order to derive the corresponding formula. It would be tempting to use a simpler approach:

$$\frac{\Theta; \cdot; \cdot \vdash B \,|\, s}{\Theta; \Gamma; \Delta \vdash [\![s]\!]B \,|\, !s}\ \Box\mathsf{I}'$$

as suggested in Remark 4.2.4. However, if we were to replace $\Box\mathsf{I}$ with $\Box\mathsf{I}$', the proof normalization process would yield invalid proofs [AB07]. For instance:

$$\dfrac{\dfrac{\dfrac{\dfrac{\pi_1}{\Theta; x^A; \cdot \vdash B \mid s}}{\Theta; \cdot; \cdot \vdash A \supset B \mid \lambda x^A.s}\supset\mathsf{I} \qquad \dfrac{\pi_2}{\Theta; \cdot; \cdot \vdash A \mid t}}{\Theta; \cdot; \cdot \vdash B \mid \Theta; \cdot; \cdot \vdash B \mid (\lambda x^A.s) \cdot t}\supset\mathsf{E}}{\Theta; \Gamma; \Delta \vdash [\![(\lambda x^A.s) \cdot t]\!]B \mid !((\lambda x^A.s) \cdot t)}\Box\mathsf{I}' \quad \rightarrow \quad \dfrac{\dfrac{\pi_3}{\Theta; \cdot; \cdot \vdash B \mid \Theta; \cdot; \cdot \vdash B \mid s\{x^A \leftarrow t\}}}{\Theta; \Gamma; \Delta \vdash [\![(\lambda x^A.s) \cdot t]\!]B \mid !((\lambda x^A.s) \cdot t)}\Box\mathsf{I}'$$

by resorting to an appropriate substitution principle for truth variables. The derivation which results from the reduction in the above example is invalid, since the witnesses $(\lambda x^A.s) \cdot t$ and $s\{x^A \leftarrow t\}$ are not the same. This shows that a naïve approach to Type Preservation is doomed to fail. The current formulation of $\Box\mathsf{I}$ allows us to regain this property. It relies on **proof witness equivalence judgements**, which take the form:

$$\Theta; \Gamma; \Delta \vdash s \equiv t : A$$

stating that s and t are equivalent witnesses for A under the hypotheses from the contexts Θ, Γ, Δ. Following 4.1, we abbreviate proof witness equivalence judgements as follows:

$$\mathcal{H} \vdash s \equiv t : A$$

The meaning for such judgements is given by the schemes depicted in Fig. 4.3, Fig. 4.4, and Fig. 4.5. The principal inference schemes (Fig. 4.3 and Fig. 4.4) establish the equivalence of witnesses induced by a single step of proof normalization at the root. Normalization steps performed elsewhere or multiple times are encoded using the inference schemes from Fig. 4.5. These ensure that proof witness equivalence is indeed an equivalence, and is compatible with all operators except "!". The reason why the equivalence is not compatible with "!" is that, whenever two witnesses s and t are equivalent, they must both be witnesses of (different) proofs of the same formula (see Lemma 4.2.15). However, $\Box\mathsf{I}$ states that a witness of the form $!s$ can only be the witness of a proof of $[\![s]\!]A$ for some A. Since $[\![s]\!]A$ and $[\![t]\!]A$ are not the same formula (for $s \neq t$), then $!s$ and $!t$ cannot be equivalent, even if s and t are. However, as we will see later, using an equivalence relation which is not compatible with the "!" operator is not an obstacle for the confluence and termination of proof normalization.

The following examples make use of proof witness equivalence.

Example 4.2.13 A proof of $w^A; \cdot; \cdot \vdash [\![w^A]\!]A \mid !w^A$ follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{w^A; x^A; \cdot \vdash A \mid x^A}\mathsf{Var}}{w^A; \cdot; \cdot \vdash A \supset A \mid \lambda x^A.x^A}\supset\mathsf{I} \qquad \dfrac{}{w^A; \cdot; \cdot \vdash A \mid w^A}\mathsf{VarM}}{w^A; \cdot; \cdot \vdash A \mid (\lambda x^A.x^A) \cdot w^A}\supset\mathsf{E} \qquad \dfrac{\dfrac{}{w^A; x^A; \cdot \vdash A \mid x^A}\mathsf{Var} \quad \dfrac{}{w^A; \cdot; \cdot \vdash A \mid w^A}\mathsf{VarM}}{w^A; \cdot; \cdot \vdash (\lambda x^A.x^A) \cdot w^A \equiv w^A : A}\mathsf{Eq}\text{-}\beta}{w^A; \cdot; \cdot \vdash [\![w^A]\!]A \mid !w^A}\Box\mathsf{I}$$

Example 4.2.14 Another proof of $w^A; \cdot; \cdot \vdash [\![w^A]\!]A \mid !w^A$. While both this and the previous example prove the same judgement, once the term assignment is introduced, each of these derivations will be assigned a different term.

$$\frac{\mathcal{H}, x^A \vdash B \,|\, s \quad \mathcal{H} \vdash A \,|\, t}{\mathcal{H} \vdash (\lambda x^A.s) \cdot t \equiv s\{x^A \leftarrow t\} : B}\ \mathsf{Eq}\text{-}\beta$$

$$\frac{\Theta; \cdot; \cdot \vdash A \,|\, s \quad \Theta, v^A; \Gamma; \Delta \vdash C \,|\, t}{\Theta; \Gamma; \Delta \vdash t\langle v^A{:=}s, !s\rangle \equiv t\{v^A \leftarrow s\} : C\{v^A \leftarrow s\}}\ \mathsf{Eq}\text{-}\gamma$$

$$\frac{\mathcal{H} \vdash A \supset B \,|\, r \quad \mathcal{H} \vdash A \,|\, t}{\mathcal{H} \vdash (r + s) \cdot t \equiv (r \cdot t) + s : B}\ \mathsf{Eq}\text{-}\psi_L$$

$$\frac{\mathcal{H} \vdash A \supset B \,|\, s \quad \mathcal{H} \vdash A \,|\, t}{\mathcal{H} \vdash (r + s) \cdot t \equiv r + (s \cdot t) : B}\ \mathsf{Eq}\text{-}\psi_R$$

$$\frac{\mathcal{H} \vdash [\![r]\!]A \,|\, s \quad \mathcal{H}, v^A \vdash C \,|\, u}{\mathcal{H} \vdash u\langle v^A{:=}r, (s + t)\rangle \equiv u\langle v^A{:=}r, s\rangle + t : C\{v^A \leftarrow r\}}\ \mathsf{Eq}\text{-}\phi_L$$

$$\frac{\mathcal{H} \vdash [\![r]\!]A \,|\, t \quad \mathcal{H}, v^A \vdash C \,|\, u}{\mathcal{H} \vdash u\langle v^A{:=}r, (s + t)\rangle \equiv s + u\langle v^A{:=}r, t\rangle : C\{v^A \leftarrow r\}}\ \mathsf{Eq}\text{-}\phi_R$$

Figure 4.3: Schemes defining the meaning of the proof witness judgement (1/2)

$$\frac{\dfrac{}{w^A; \cdot; \cdot \vdash A \,|\, w^A}\ \mathsf{VarM} \qquad \dfrac{\dfrac{}{w^A; \cdot; \cdot \vdash A \,|\, w^A}\ \mathsf{VarM}}{w^A; \cdot; \cdot \vdash w^A \equiv w^A : A}\ \mathsf{Eq\text{-}Refl}}{w^A; \cdot; \cdot \vdash [\![w^A]\!]A \,|\, !w^A}\ \Box\mathsf{I}$$

Some basic metatheoretical results follow.

Lemma 4.2.15 If $\mathcal{H} \vdash s \equiv t : B$, both $\mathcal{H} \vdash B \,|\, s$ and $\mathcal{H} \vdash B \,|\, t$ are derivable.

Proof.- The proof is developed in Section 4.4. □

Lemma 4.2.16 (Weakening and Strengthening) Suppose $\rhd_{\mathsf{HLP}} \Theta; \Gamma; \Delta \vdash s \equiv t : A$. Then:

1. $\rhd_{\mathsf{HLP}} \Theta \cup \Theta'; \Gamma \cup \Gamma'; \Delta \cup \Delta' \vdash s \equiv t : A$; and
2. $\rhd_{\mathsf{HLP}} \Theta \cap \mathsf{FVV}(s) \cap \mathsf{FVV}(t); \Gamma \cap \mathsf{FVT}(s) \cap \mathsf{FVT}(t); \Delta \cap \mathsf{FVF}(s) \cap \mathsf{FVF}(t) \vdash s \equiv t : A.\mathsf{FVT}(s) \subseteq \Gamma$ and $\mathsf{FVF}(s) \subseteq \Delta$.

Proof.- By induction on the derivation of $\Theta; \Gamma; \Delta \vdash s \equiv t : A$. □

The equational theory is **consistent** in the sense that there exist $\mathcal{H}, A, s$ and t s.t.

- $\mathcal{H} \vdash A \,|\, s$ is derivable.

$$\frac{\mathcal{H}, \alpha^A, \beta^A \vdash \bot \,|\, s}{\mathcal{H}, \beta^A \vdash [\beta^A]\mu\alpha^A.s \equiv s\{\alpha^A \leftarrow \beta^A\} : \bot}\ \mathsf{Eq}\text{-}\mu$$

$$\frac{\mathcal{H}, \alpha^{A \supset B} \vdash \bot \,|\, s \quad \mathcal{H} \vdash A \,|\, t}{\mathcal{H} \vdash (\mu\alpha^{A \supset B}.s) \cdot t \equiv \mu\beta^B.s(\!|\, [\alpha^{A \supset B}](\bullet) \leftarrow [\beta^B](\bullet)t \,|\!) : B}\ \mathsf{Eq}\text{-}\zeta$$

$$\frac{\mathcal{H} \vdash A \,|\, s \quad \alpha^A \notin \mathsf{FVF}(s)}{\mathcal{H} \vdash \mu\alpha^A.[\alpha^A]s \equiv s : A}\ \mathsf{Eq}\text{-}\theta$$

$$\frac{\mathcal{H}, \alpha^A \vdash A \,|\, s}{\mathcal{H}, \alpha^A \vdash [\alpha^A]s + t \equiv ([\alpha^A]s) + t : \bot}\ \mathsf{Eq}\text{-}\chi_L \qquad \frac{\mathcal{H}, \alpha^A \vdash A \,|\, t}{\mathcal{H}, \alpha^A \vdash [\alpha^A]s + t \equiv s + [\alpha^A]t : \bot}\ \mathsf{Eq}\text{-}\chi_R$$

$$\frac{\mathcal{H}, \alpha^A \vdash \bot \,|\, s}{\mathcal{H} \vdash \mu\alpha^A.(s + t) \equiv (\mu\alpha^A.s) + t : A}\ \mathsf{Eq}\text{-}\iota_L \qquad \frac{\mathcal{H}, \alpha^A \vdash \bot \,|\, t}{\mathcal{H} \vdash \mu\alpha^A.(s + t) \equiv s + \mu\alpha^A.t : A}\ \mathsf{Eq}\text{-}\iota_R$$

Figure 4.4: Schemes defining the meaning of the proof witness judgement (2/2)

- $\mathcal{H} \vdash A \,|\, t$ is derivable.
- The judgement $\mathcal{H} \vdash s \equiv t : A$ is *not* derivable.

Consistency is proved in Section 4.6 (Corollary 4.6.3).

Remark 4.2.17 If $\rhd_{\mathsf{HLP}} \Theta; \Gamma; \Delta \vdash A \,|\, s$, then there is a derivation of $\Theta; \Gamma; \Delta \vdash A \,|\, s$ which does not make use of the equivalence rules. That is, there is a derivation which uses $\Box\mathsf{I}'$ instead of $\Box\mathsf{I}$. This follows from Lemma 4.2.15, whose proof does not introduce applications of $\Box\mathsf{I}$. However, this derivation may not be in normal form.

4.3 Relating LP and HLP

Definition 4.3.1 (Translation from LP to HLP) The translation $\underline{\bullet}$ from LP to HLP is defined below. For propositions it traverses the term structure:

$$\begin{array}{rcl} \underline{P} & \triangleq & P \\ \underline{\bot} & \triangleq & \bot \\ \underline{A \supset B} & \triangleq & \underline{A} \supset \underline{B} \\ \underline{[\![s]\!]A} & \triangleq & [\![\underline{s}]\!]\underline{A} \end{array}$$

For proof polynomials it is defined as follows:

$$\frac{\mathcal{H} \vdash A \,|\, s}{\mathcal{H} \vdash\ s \equiv s \colon A}\ \textsf{Eq-Refl} \qquad \frac{\mathcal{H} \vdash\ s \equiv t \colon A}{\mathcal{H} \vdash\ t \equiv s \colon A}\ \textsf{Eq-Symm}$$

$$\frac{\mathcal{H} \vdash\ r \equiv s \colon A \quad \mathcal{H} \vdash\ s \equiv t \colon A}{\mathcal{H} \vdash\ r \equiv t \colon A}\ \textsf{Eq-Trans}$$

$$\frac{\mathcal{H}, x^A \vdash\ s \equiv t \colon B}{\mathcal{H} \vdash\ \lambda x^A.s \equiv \lambda x^A.t \colon A \supset B}\ \textsf{Eq-}\lambda$$

$$\frac{\mathcal{H} \vdash\ s \equiv s' \colon A \supset B \quad \mathcal{H} \vdash\ t \equiv t' \colon A}{\mathcal{H} \vdash\ s \cdot t \equiv s' \cdot t' \colon B}\ \textsf{Eq-}\cdot$$

$$\frac{\mathcal{H} \vdash\ s \equiv s' \colon [\![r]\!]A \quad \mathcal{H}, v^A \vdash\ t \equiv t' \colon C}{\mathcal{H} \vdash\ (t\langle v^A{:=}r, s\rangle) \equiv t'\langle v^A{:=}r, s'\rangle \colon C\{v^A \leftarrow r\}}\ \textsf{Eq-}\langle\rangle$$

$$\frac{\mathcal{H} \vdash\ r \equiv s \colon A}{\mathcal{H} \vdash\ r + t \equiv s + t \colon A}\ \textsf{Eq-}+_{\mathrm{L}} \qquad \frac{\mathcal{H} \vdash\ r \equiv s \colon A}{\mathcal{H} \vdash\ t + r \equiv t + s \colon A}\ \textsf{Eq-}+_{\mathrm{R}}$$

$$\frac{\mathcal{H}, \alpha^A \vdash\ s \equiv t \colon A}{\mathcal{H}, \alpha^A \vdash\ [\alpha^A]s \equiv [\alpha^A]t \colon \bot}\ \textsf{Eq-}[\alpha] \qquad \frac{\mathcal{H}, \alpha^A \vdash\ s \equiv t \colon \bot}{\mathcal{H} \vdash\ \mu\alpha^A.s \equiv \mu\alpha^A.t \colon A}\ \textsf{Eq-}\mu\alpha$$

Figure 4.5: Equivalence and compatibility schemes

$$\begin{array}{rcl}
\underline{x^A} & \triangleq & x^{\underline{A}} \\
\underline{s \cdot t} & \triangleq & \underline{s} \cdot \underline{t} \\
\underline{!s} & \triangleq & !\underline{s} \\
\underline{s + t} & \triangleq & \underline{s} + \underline{t} \\
\underline{c^{\mathbf{A0}}_{A,B}} & \triangleq & (\lambda x^{\underline{A}}.\lambda y^{\underline{B}}.x) \\
\underline{c^{\mathbf{A0}}_{A,B,C}} & \triangleq & (\lambda x^{\underline{A \supset B \supset C}}.\lambda y^{\underline{A \supset B}}.\lambda z^{\underline{A}}.x \cdot z \cdot (y \cdot z)) \\
\underline{c^{\mathbf{A0}}_{A}} & \triangleq & (\lambda y^{\underline{\neg\neg A}}.\mu\alpha^{\underline{A}}.y \cdot \lambda x^{\underline{A}}.[\alpha^{\underline{A}}]x) \\
\underline{c^{\mathbf{A1}}_{t,A}} & \triangleq & \lambda x^{\underline{[\![t]\!]A}}.v^{\underline{A}}\langle v^{\underline{A}}{:=}\underline{t}, x\rangle \\
\underline{c^{\mathbf{A2}}_{s,t,A,B}} & \triangleq & \lambda x^{\underline{[\![s]\!]A \supset B}}.\lambda y^{\underline{[\![t]\!]A}}.!(w \cdot v)\langle w^{\underline{A \supset B}}{:=}\underline{s}, x\rangle\langle v^{\underline{A}}{:=}\underline{t}, y\rangle \\
\underline{c^{\mathbf{A3}}_{s,A}} & \triangleq & \lambda x^{\underline{[\![s]\!]A}}.!!v^{\underline{A}}\langle v^{\underline{A}}{:=}\underline{s}, x^{\underline{[\![s]\!]A}}\rangle \\
\underline{c^{\mathbf{A4}}_{s,t,A}} & \triangleq & \lambda x^{\underline{[\![s]\!]A}}.!(v^{\underline{A}} + \underline{t})\langle v^{\underline{A}}{:=}\underline{s}, x\rangle \\
\underline{c^{\mathbf{A5}}_{s,t,A}} & \triangleq & \lambda x^{\underline{[\![t]\!]A}}.!(\underline{s} + v^{\underline{A}})\langle v^{\underline{A}}{:=}\underline{t}, x\rangle \\
\underline{\Gamma} & \triangleq & \{x^{\underline{A}} \mid x^A \in \Gamma\}
\end{array}$$

Proposition 4.3.2 If $\rhd_{\mathsf{LP}} \Gamma \vdash A$, then $\rhd_{\mathsf{HLP}} \cdot; \underline{\Gamma}; \cdot \vdash \underline{A} \,|\, s$ for some proof witness s.

Proof.- By induction on the derivation of $\Gamma \vdash A$. If $x^A \in \Gamma$, the result is immediate by Var, and $s = x^A$. For the cases where A is an axiom **A0-A5**, we will show that $\underline{A}$ can be derived in the empty context. By *weakening*, this means it can also be derived in any context $\underline{\Gamma}$.

A0. Note that all classical tautologies are derivable in HLP (Cor. 4.2.12), in particular the axioms of classical propositional logic:

- $\cdot; \Gamma; \cdot \vdash A \supset B \supset A \mid \lambda x^A.\lambda y^B.x^A$
- $\cdot; \Gamma; \cdot \vdash (A \supset B \supset C) \supset (A \supset B) \supset A \supset C \mid \lambda x^{A \supset B \supset C}.\lambda y^{A \supset B}.\lambda z^A.x^{A \supset B \supset C} \cdot z^A \cdot (y^{A \supset C} \cdot z^A)$
- $\cdot; \Gamma; \cdot \vdash \neg\neg A \supset A \mid \lambda y^{\neg\neg A}.\mu\alpha^A.y^{\neg\neg A} \cdot \lambda x^A.[\alpha^A]x^A$

A1. $\Gamma \vdash [\![t]\!]A \supset A$.

$$\dfrac{\dfrac{\dfrac{}{\cdot; x^{[\![t]\!]A}; \cdot \vdash [\![t]\!]A \mid x^{[\![t]\!]A}}\,\mathsf{Var} \qquad \dfrac{}{v^A; x^{[\![t]\!]A}; \cdot \vdash A \mid v^A}\,\mathsf{VarM}}{\cdot; x^{[\![t]\!]A}; \cdot \vdash A \mid v^A\langle v^A := t, x^{[\![t]\!]A}\rangle}\,\Box\mathsf{E}}{\cdot; \cdot; \cdot \vdash [\![t]\!]A \supset A \mid \lambda x^{[\![t]\!]A}.v^A\langle v^A := t, x^{[\![t]\!]A}\rangle}\,\supset\mathsf{I}$$

A2. $\Gamma \vdash [\![s]\!](A \supset B) \supset [\![t]\!]A \supset [\![s \cdot t]\!]B$. Let $\Gamma_1 = \{x^{[\![s]\!](A \supset B)}, y^{[\![t]\!]A}\}$, $\Theta_1 = \{w^{A \supset B}, v^A\}$. We split the proof in two parts:

Part 1:

$$\dfrac{\dfrac{}{v^A; \Gamma_1; \cdot \vdash [\![s]\!](A \supset B) \mid x^{[\![s]\!](A \supset B)}}\,\mathsf{Var} \qquad \dfrac{\dfrac{\dfrac{}{\Theta_1; \cdot; \cdot \vdash A \supset B \mid w^{A \supset B}}\,\mathsf{VarM} \quad \dfrac{}{\Theta_1; \cdot; \cdot \vdash A \mid v^A}\,\mathsf{VarM}}{\Theta_1; \cdot; \cdot \vdash B \mid w^{A \supset B} \cdot v^A}\,\supset\mathsf{E}}{\Theta_1; \Gamma_1; \cdot \vdash [\![w^{A \supset B} \cdot v^A]\!]B \mid !(w^{A \supset B} \cdot v^A)}\,\Box\mathsf{I}}{v^A; \Gamma_1; \cdot \vdash [\![s \cdot v^A]\!]B \mid !(w^{A \supset B} \cdot v^A)\langle w^{A \supset B} := s, x^{[\![s]\!](A \supset B)}\rangle}\,\Box\mathsf{E}$$

Part 2:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\cdot; \Gamma_1; \cdot \vdash [\![t]\!]A \mid y^{[\![t]\!]A}}\,\mathsf{Var} \qquad \dfrac{\cdots}{\begin{array}{l} v^A; \Gamma_1; \cdot \vdash [\![s \cdot v^A]\!]B \mid \\ !(w^{A \supset B} \cdot v^A)\langle w^{A \supset B} := s, x^{[\![s]\!](A \supset B)}\rangle \end{array}}\,\text{Part 1}}{\cdot; \Gamma_1; \cdot \vdash [\![s \cdot t]\!]B \mid !(w^{A \supset B} \cdot v^A)\langle w^{A \supset B} := s, x^{[\![s]\!](A \supset B)}\rangle\langle v^A := t, y^{[\![t]\!]A}\rangle}\,\Box\mathsf{E}}{\cdot; x^{[\![s]\!](A \supset B)}; \cdot \vdash [\![s \cdot t]\!]B \mid \lambda y^{[\![t]\!]A}.!(w^{A \supset B} \cdot v^A)\langle w^{A \supset B} := s, x^{[\![s]\!](A \supset B)}\rangle\langle v^A := t, y^{[\![t]\!]A}\rangle}\,\supset\mathsf{I}}{\begin{array}{l} \cdot; \cdot; \cdot \vdash [\![s]\!](A \supset B) \supset ([\![t]\!]A \supset [\![s \cdot t]\!]B) \mid \\ \lambda x^{[\![s]\!](A \supset B)}.\lambda y^{[\![t]\!]A}.!(w^{A \supset B} \cdot v^A)\langle w^{A \supset B} := s, x^{[\![s]\!](A \supset B)}\rangle\langle v^A := t, y^{[\![t]\!]A}\rangle \end{array}}\,\supset\mathsf{I}$$

A3. $\Gamma' \vdash [\![t']\!]A' \supset [\![!t']\!][\![t']\!]A'$

$$\dfrac{\dfrac{\dfrac{}{\cdot; x^{[\![s]\!]A}; \cdot \vdash [\![s]\!]A \mid x^{[\![s]\!]A}}\,\mathsf{Var} \qquad \dfrac{\dfrac{\dfrac{}{v^A; \cdot; \cdot \vdash A \mid v^A}\,\mathsf{VarM}}{v^A; \cdot; \cdot \vdash [\![v^A]\!]A \mid !v^A}\,\Box\mathsf{I}}{v^A; x^{[\![s]\!]A}; \cdot \vdash [\![!v^A]\!][\![v^A]\!]A \mid !!v^A}\,\Box\mathsf{I}}{\cdot; x^{[\![s]\!]A}; \cdot \vdash [\![!s]\!][\![s]\!]A \mid !!v^A\langle v^A := s, x^{[\![s]\!]A}\rangle}\,\Box\mathsf{E}}{\cdot; \cdot; \cdot \vdash [\![s]\!]A \supset [\![!s]\!][\![s]\!]A \mid \lambda x^{[\![s]\!]A}.!!v^A\langle v^A := s, x^{[\![s]\!]A}\rangle}\,\supset\mathsf{I}$$

A4. $\Gamma \vdash [\![s]\!]A \supset [\![s+t]\!]A$.

$$\dfrac{\dfrac{\dfrac{}{\cdot; x^{[\![s]\!]A}; \cdot \vdash [\![s]\!]A \mid x^{[\![s]\!]A}}\,\mathsf{Var} \qquad \dfrac{\dfrac{\dfrac{}{v^A; \cdot; \cdot \vdash A \mid v^A}\,\mathsf{VarM}}{v^A; \cdot; \cdot \vdash A \mid v^A + t}\,\mathsf{PlusL}}{v^A; x^{[\![s]\!]A}; \cdot \vdash [\![v^A + t]\!]A \mid !(v^A + t)}\,\Box\mathsf{I}}{\cdot; x^{[\![s]\!]A}; \cdot \vdash [\![s+t]\!]A \mid !(v^A + t)\langle v^A := s, x^{[\![s]\!]A}\rangle}\,\Box\mathsf{E}}{\cdot; \cdot; \cdot \vdash [\![s]\!]A \supset [\![s+t]\!]A \mid \lambda x^{[\![s]\!]A}.!(v^A + t)\langle v^A := s, x^{[\![s]\!]A}\rangle}\,\supset\mathsf{I}$$

A5. $\Gamma \vdash [\![t]\!]A \supset [\![s+t]\!]A$

$$\dfrac{\dfrac{\dfrac{}{\cdot; x^{[\![t]\!]A}; \cdot \vdash [\![t]\!]A \mid x^{[\![t]\!]A}}\,\mathsf{Var} \qquad \dfrac{\dfrac{\dfrac{}{v^A; \cdot; \cdot \vdash A \mid v^A}\,\mathsf{VarM}}{v^A; \cdot; \cdot \vdash A \mid s + v^A}\,\mathsf{PlusR}}{v^A; x^{[\![t]\!]A}; \cdot \vdash [\![s + v^A]\!]A \mid !(s + v)}\,\Box\mathsf{I}}{\cdot; x^{[\![t]\!]A}; \cdot \vdash [\![s+t]\!]A \mid !(s + v^A)\langle v^A := t, x^{[\![t]\!]A}\rangle}\,\Box\mathsf{E}}{\cdot; \cdot; \cdot \vdash [\![t]\!]A \supset [\![s+t]\!]A \mid \lambda x^{[\![t]\!]A}.!(s + v^A)\langle v^A := t, x^{[\![t]\!]A}\rangle}\,\supset\mathsf{I}$$

MP. $\Gamma \vdash D$ was derived from $\Gamma \vdash C \supset D$ and $\Gamma \vdash C$. Let $\Gamma' = \underline{\Gamma}$, $A = \underline{C}$), $B = \underline{D}$. By induction hypothesis, we know that $\cdot; \Gamma; \cdot \vdash A \supset B \mid s$ and $\cdot; \Gamma; \cdot \vdash A \mid t$. Thus:

$$\dfrac{\cdot; \Gamma'; \cdot \vdash A \supset B \mid s \quad \cdot; \Gamma'; \cdot \vdash A \mid t}{\cdot; \Gamma'; \cdot \vdash B \mid s \cdot t}\,\supset\mathsf{E}$$

Nec. In this case, it is easy to verify that for each instance of each axiom scheme $\mathbf{A}_i(\cdots)$ (the dots indicate the formulas parameters instantiating the axiom scheme) with $i \in 0..5$, the judgement $\cdot; \cdot; \cdot \vdash \underline{\mathbf{A}_i(\cdots)} \mid \underline{c^{\mathbf{Ai}}_{\cdots}}$ can be derived. Therefore,

$$\dfrac{\cdot; \cdot; \cdot \vdash \underline{\mathbf{A}_i(\cdots)} \mid \underline{c^{\mathbf{Ai}}_{\cdots}}}{\cdot; \cdot; \cdot \vdash [\![\underline{c^{\mathbf{Ai}}_{\cdots}}]\!]\underline{\mathbf{A}_i(\cdots)} \mid !\underline{c^{\mathbf{Ai}}_{\cdots}}}\,\Box\mathsf{I}$$

□

Remark 4.3.3 The following alternative inference scheme for PlusL (and similarly for PlusR):

$$\dfrac{\Theta; \Gamma; \Delta \vdash A \mid s \quad \mathsf{FVV}(t) \subseteq \Theta \quad \mathsf{FVT}(s) \subseteq \Gamma \quad \mathsf{FVF}(t) \subseteq \Delta}{\Theta; \Gamma; \Delta \vdash A \mid s + t}\,\mathsf{PlusL}$$

does not allow the proof of Proposition 4.3.2 to go through in the case of axiom **A4**, since no restriction is a priori placed on t in that axiom, and $\Box$I requires that there be no truth or falsehood dependencies.

While it is not clear whether it can prove *all* LP-theorems, an option which may still be worth exploring is to resort to truth dependent modalities [NPP08]. The modality $\Box A$ is replaced by $[\Gamma]A$ which informally may be read as $\Box(\Gamma \supset A)$. That is, validity of A is dependent on the truth of the hypothesis in Γ. A sample of three inference schemes of the resulting *Contextual Modal Logic* [NPP08] are:

$$\frac{\Theta;\Gamma_1 \vdash A}{\Theta;\Gamma_2 \vdash [\Gamma_1]A}\,\Box\mathsf{I} \qquad \frac{\Theta;\Gamma_1 \vdash [\Gamma_2]A \quad \Theta, v :: A[\Gamma_2];\Gamma_1 \vdash C}{\Theta;\Gamma_1 \vdash C}\,\Box\mathsf{E}^v$$

$$\frac{\Theta_1, v :: A[\Gamma_1], \Theta_2;\Gamma_2 \vdash \sigma \Leftarrow \Gamma_1}{\Theta_1, v :: A[\Gamma_1], \Theta_2;\Gamma_2 \vdash \mathsf{clo}(v,\sigma) \Rightarrow [\sigma]^a_{\Gamma_1}(A)}\,\mathsf{mvar}$$

We will now address the reverse direction, namely that all theorems of HLP are theorems of LP. More precisely, we seek to prove the following result for a suitable translation from proof witnesses and formulas in HLP to proof polynomials and formulas in LP.

Corollary 4.3.4 If $\rhd_{\mathsf{HLP}} \cdot;\cdot;\cdot \vdash A \,|\, s$, then both $\rhd_{\mathsf{LP}} \cdot \vdash [\![s^\star]\!]A^\star$ and $\rhd_{\mathsf{LP}} \cdot \vdash A^\star$.

A proof witness s is **inhabited** if for some Θ, Γ, Δ and A, the judgement $\Theta;\Gamma;\Delta \vdash A \,|\, s$ is derivable.

Definition 4.3.5 (Translation from HLP to LP) It turns out the translation is unproblematic for formulas and hypotheses:

$$\begin{array}{rcl@{\qquad}rcl}
P^\star & \triangleq & P & \cdot^\star & \triangleq & \cdot \\
\bot^\star & \triangleq & \bot & (\Theta, v^A)^\star & \triangleq & \Theta^\star, [\![v^{A^\star}]\!]A^\star \\
(A \supset B)^\star & \triangleq & A^\star \supset B^\star & (\Gamma, x^A)^\star & \triangleq & \Gamma^\star, [\![x^{A^\star}]\!]A^\star \\
([\![s]\!]A)^\star & \triangleq & [\![s^\star]\!]A^\star & (\Delta, \alpha^A)^\star & \triangleq & \Delta^\star, [\![\alpha^{\neg A^\star}]\!]\neg A^\star
\end{array}$$

However, special care must be taken when defining the translation on proof witnesses. The following cases are straightforward:

$$\begin{array}{rcl@{\qquad}rcl}
(x^A)^\star & \triangleq & x^{A^\star} & (s \cdot t)^\star & \triangleq & s^\star \cdot t^\star \\
(v^A)^\star & \triangleq & v^{A^\star} & ([\alpha^A]s)^\star & \triangleq & \alpha^{\neg A^\star} \cdot s^\star \\
(s{+}t)^\star & \triangleq & s^\star{+}t^\star & (!s)^\star & \triangleq & !(s^\star) \\
 & & & (t\langle v^A{:=}r, s\rangle)^\star & \triangleq & t^\star\{v^{A^\star} \leftarrow r^\star\}
\end{array}$$

The more delicate cases are: $(\lambda x^A.s)^\star$ and $(\mu\alpha^A.s)^\star$. We briefly explain the case of the abstraction since that of name abstraction is similar. Bear in mind that the translation of an HLP-judgement $\Theta;\Gamma;\Delta \vdash A \,|\, s$ is defined as:

$$(\Theta;\Gamma;\Delta \vdash A \,|\, s)^\star \triangleq \Theta^\star, \Gamma^\star, \Delta^\star \vdash [\![s^\star]\!]A^\star$$

Suppose that the last scheme applied in the derivation of a judgement $\Theta;\Gamma;\Delta \vdash C \,|\, s$ is:

$$\frac{\Theta;\Gamma, x^A;\Delta \vdash B \,|\, s}{\Theta;\Gamma;\Delta \vdash A \supset B \,|\, \lambda x^A.s}\,\supset\mathsf{I}$$

The IH will yield derivability in LP of:

$$\Theta^\star \cup \Gamma^\star, [\![x^{A^\star}]\!]A^\star \cup \Delta^\star \vdash [\![s^\star]\!]B^\star \tag{4.2}$$

However, we are after derivability of $\Theta^\star \cup \Gamma^\star \cup \Delta^\star \vdash [\![t]\!](A^\star \supset B^\star)$, for an appropriate proof polynomial t. Building a derivation of this judgement requires three steps:

1. We first need to "drop" the outermost modalities of $[\![x^{A^\star}]\!]A^\star$ and $[\![s^\star]\!]B^\star$ from (4.2). This is achieved via the Stripping Lemma (Lem. 4.1.6).

2. This allows us then to resort to the standard Deduction Theorem to deduce $A^\star \supset B^\star$.

3. Finally, we resort to the reflective capabilities of LP in order to deduce the appropriate proof polynomial t. This is achieved via the Internalization Lemma (Lem. 4.1.5).

These three steps conform the content of the Abstraction Lemma (Lemma 4.1.7). Note that t is thus a function of the original LP derivation of (4.2). Since there may be multiple LP derivations of an LP judgement we shall assume in our proof of Corollary 4.3.4 that all derivable occurrences of (4.2) use the same derivation.

Returning to our translation described in Definition 4.3.5, we now address the defining clauses for $(\lambda x^A.s)^\star$ and $(\mu\alpha^A.s)^\star$. Let $c^{\mathbf{A1}}$ be the proof constant denoting any instance of **A1**.

$$
\begin{array}{lcl}
(\lambda x^A.s)^\star & \triangleq & \text{any } t_\lambda^{A^\star \supset B^\star}(\Theta^\star \cup \Gamma^\star \cup \Delta^\star) \text{ for any } \Theta,\ \Gamma,\ \Delta,\ B \text{ s.t.} \\
 & & \Theta; \Gamma; \Delta \vdash A \supset B \,|\, \lambda x^A.s \text{ is derivable, if } \lambda x^A.s \text{ is inhabited.} \\
(\lambda x^A.s)^\star & \triangleq & c^{\mathbf{A1}} \cdot c^{\mathbf{A1}}, \text{ otherwise.} \\
(\mu\alpha^A.s)^\star & \triangleq & \text{any } t_\mu^{A^\star}(\Theta^\star \cup \Gamma^\star \cup \Delta^\star) \text{ for any } \Theta,\ \Gamma,\ \Delta \text{ s.t.} \\
 & & \Theta; \Gamma; \Delta \vdash A \,|\, \mu\alpha^A.s \text{ is derivable, if } \mu\alpha^A.s \text{ is inhabited.} \\
(\mu\alpha^A.s)^\star & \triangleq & c^{\mathbf{A1}} \cdot c^{\mathbf{A1}}, \text{ otherwise.}
\end{array}
$$

In our use of this translation (Prop. 4.3.6) the conditions of the first clause and third clauses shall be met when dealing with modalities that are introduced using $\Box$I; the other cases are used when uninhabited proof witnesses occur inside boxes that are not introduced. In the former cases, note that there may be more than one possible proof polynomial (for instance, $\lambda x^A.(y^B + z^B)$ and $[\![\lambda x^A.(y^B + z^B)]\!]A \supset B$). Any of these may be resorted to since they are all associated proof polynomials of the same formula under the same context, provided that the proof witnesses to be translated can verify some formula (i.e. they are the s in some derivable judgment $\Theta; \Gamma; \Delta \vdash A \,|\, s$). Otherwise, if the proof witness is not inhabited (for example $\lambda x^A.x^A \cdot x^A$), then its content is unimportant and any translation will yield the same results. For each inhabited proof witness, we shall assume that we use one and the same proof of the corresponding judgement.

Proposition 4.3.6 If $\Theta; \Gamma; \Delta \vdash D \,|\, s$ is derivable in HLP without resorting to proof witness equivalence, the LP-judgement $\Theta^\star \cup \Gamma^\star \cup \Delta^\star \vdash [\![s^\star]\!]D^\star$ is derivable in LP.

Proof.- Since $\Theta; \Gamma; \Delta \vdash D \,|\, s$ is derivable in HLP, let π be a derivation of $\Theta; \Gamma; \Delta \vdash D \,|\, s$ in HLP. We prove by induction on π that $\rhd_{\mathsf{LP}} \Theta^\star \cup \Gamma^\star \cup \Delta^\star \vdash [\![s^\star]\!]D^\star$.

1. Case Var (VarM is similar and hence omitted): $\Theta; \Gamma; \Delta \vdash D \,|\, s$ is of the form $\Theta; \Gamma', x^A; \Delta \vdash A \,|\, x^A$. Trivially we have $\rhd_{\mathsf{LP}} [\![x^{A^\star}]\!]A^\star \vdash [\![x^{A^\star}]\!]A^\star$ and hence $\rhd_{\mathsf{LP}} \Theta^\star \cup \Gamma^\star \cup \Delta^\star \vdash [\![x^{A^\star}]\!]A^\star$ by Weakening.

2. Case $\supset$I: the derivation ends in:

$$\frac{\Theta; \Gamma, x^A; \Delta \vdash B \mid s}{\Theta; \Gamma; \Delta \vdash A \supset B \mid \lambda x^A.s} \supset \mathsf{I}$$

we will use the contexts $\Theta^\star$, $\Gamma^\star$ and $\Delta^\star$ for the translation of $\lambda x^A.s$. We want to see that $\rhd_{\mathsf{LP}} \Theta^\star \cup \Gamma^\star \cup \Delta^\star \vdash [\![t_\lambda^{A^\star \supset B^\star}(\Theta^\star \cup \Gamma^\star \cup \Delta^\star)]\!](A^\star \supset B^\star)$. (We know $[\![t_\lambda^{A^\star \supset B^\star}(\Theta^\star \cup \Gamma^\star \cup \Delta^\star)]\!](A^\star \supset B^\star)$ is a correct translation of $[\![s]\!]D$, since $\Theta; \Gamma; \Delta \vdash A \supset B \mid \lambda x^A.s$ is derivable by hypothesis). By the IH $\rhd_{\mathsf{LP}} \Theta^\star \cup \Gamma^\star, [\![x^{A^\star}]\!]A^\star \cup \Delta^\star \vdash [\![s^\star]\!]B^\star$. Therefore, from the λ-Abstraction Lemma (4.1.7), we obtain $t_\lambda^{A^\star \supset B^\star}(\Theta^\star \cup \Gamma^\star \cup \Delta^\star)$ associated proof polynomial of $A^\star \supset B^\star$ in $\Theta^\star \cup \Gamma^\star \cup \Delta^\star$. This proof polynomial moreover verifies $\rhd_{\mathsf{LP}} \Theta^\star \cup \Gamma^\star \cup \Delta^\star \vdash [\![t_\lambda^{A^\star \supset B^\star}(\Theta^\star \cup \Gamma^\star \cup \Delta^\star)]\!](A^\star \supset B^\star)$. Note that, by Internalization, *any* associated proof polynomial of $A^\star \supset B^\star$ in $\Theta^\star \cup \Gamma^\star \cup \Delta^\star$ can be used in place of $t_\lambda^{A^\star \supset B^\star}(\Theta^\star \cup \Gamma^\star \cup \Delta^\star)$. This means that, whenever $\supset \mathsf{I}$ is used within a derivation, we may choose whatever associated proof polynomial works best in order to translate the derivation as a whole.

3. Case $\supset \mathsf{E}$: the derivation ends in:

$$\frac{\Theta; \Gamma; \Delta \vdash A \supset B \mid s \quad \Theta; \Gamma; \Delta \vdash A \mid t}{\Theta; \Gamma; \Delta \vdash B \mid s \cdot t} \supset \mathsf{E}$$

By the IH both of the following judgements are derivable in LP:

(a) $\Theta^\star \cup \Gamma^\star \cup \Delta^\star \vdash [\![s^\star]\!](A \supset B)^\star$ and

(b) $\Theta^\star \cup \Gamma^\star \cup \Delta^\star \vdash [\![t^\star]\!]A^\star$

From these, using **A2** and **MP** twice, we derive $\Theta^\star \cup \Gamma^\star \cup \Delta^\star \vdash [\![s^\star \cdot t^\star]\!]B^\star$ in LP.

Note that we can always choose the same $A^\star$ as the translation of A on both sides regardless of how each premise was derived.

4. Case $\Box\mathsf{I}$: the derivation ends in:

$$\frac{\Theta; \cdot; \cdot \vdash B \mid s}{\Theta; \Gamma; \Delta \vdash [\![s]\!]B \mid !s} \Box\mathsf{I}$$

We reason as follows:

(a)	$\Theta^\star \vdash [\![s^\star]\!]B^\star$	(IH)
(b)	$\Theta^\star \vdash [\![s^\star]\!]B^\star \supset [\![!s^\star]\!][\![s^\star]\!]B^\star$	(**A3**)
(c)	$\Theta^\star \vdash [\![!s^\star]\!][\![s^\star]\!]B^\star$	(**MP** from (b) and (a))
(d)	$\Theta^\star \cup \Gamma^\star \cup \Delta^\star \vdash [\![!s^\star]\!][\![s^\star]\!]B^\star$	(Weakening)

5. Case $\Box\mathsf{E}$: the derivation ends in:

$$\frac{\Theta; \Gamma; \Delta \vdash [\![r]\!]A \mid s \quad \Theta, v^A; \Gamma; \Delta \vdash C \mid t}{\Theta; \Gamma; \Delta \vdash C\{v^A \leftarrow r\} \mid t\langle v^A := r, s\rangle} \Box\mathsf{E}$$

By the IH both of the following judgements are derivable in LP:

(a) $\Theta^\star \cup \Gamma^\star \cup \Delta^\star \vdash [\![s^\star]\!][\![r^\star]\!]A^\star$ and

(b) $\Theta^\star, [\![v^{A^\star}]\!]A^\star \cup \Gamma^\star \cup \Delta^\star \vdash [\![t^\star]\!]C^\star$

We now reason as follows:

(1)	$\Theta^\star \cup \Gamma^\star \cup \Delta^\star \vdash [\![s^\star]\!][\![r^\star]\!]A^\star \supset [\![r^\star]\!]A^\star$	(**A1**)
(2)	$\Theta^\star \cup \Gamma^\star \cup \Delta^\star \vdash [\![r^\star]\!]A^\star$	((a) and **MP**)
(3)	$\Theta^\star \cup \Gamma^\star \cup \Delta^\star \vdash [\![t^\star\{v^{A^\star} \leftarrow r^\star\}]\!]C^\star\{v^{A^\star} \leftarrow r^\star\}$	(Lem. 4.1.10)

6. Case Name: the derivation ends in:

$$\frac{\Theta; \Gamma; \Delta, \alpha^A \vdash A \,|\, s}{\Theta; \Gamma; \Delta, \alpha^A \vdash \bot \,|\, [\alpha^A]s}\ \mathsf{Name}$$

By IH, $\Theta^\star \cup \Gamma^\star \cup \Delta^\star, x^{[\![\alpha^{\neg A^\star}]\!]\neg A^\star} \vdash [\![s^\star]\!]A^\star$ is derivable in LP.
We reason as follows:

(1)	$\Theta^\star \cup \Gamma^\star \cup \Delta^\star, x^{[\![\alpha^{\neg A^\star}]\!]\neg A^\star} \vdash [\![\alpha^{\neg A^\star}]\!]\neg A^\star$	(hypothesis $x^{[\![\alpha^{\neg A^\star}]\!]\neg A^\star}$)
(2)	$\Theta^\star \cup \Gamma^\star \cup \Delta^\star, x^{[\![\alpha^{\neg A^\star}]\!]\neg A^\star} \vdash [\![s^\star]\!]A^\star$	(IH)
(3)	$\Theta^\star \cup \Gamma^\star \cup \Delta^\star, x^{[\![\alpha^{\neg A^\star}]\!]\neg A^\star} \vdash [\![\alpha^{\neg A^\star} \cdot s^\star]\!]\bot$	(**A2** and **MP** twice)

7. Case NAbs: the derivation ends in:

$$\frac{\Theta; \Gamma; \Delta, \alpha^A \vdash \bot \,|\, s}{\Theta; \Gamma; \Delta \vdash A \,|\, \mu\alpha^A.s}\ \mathsf{NAbs}$$

By IH $\rhd_{\mathsf{LP}} \Theta^\star \cup \Gamma^\star \cup \Delta^\star, [\![\alpha^{\neg A^\star}]\!]A^\star \vdash [\![s^\star]\!]\bot$. From the μ-Abstraction Corollary (4.1.8), $\rhd_{\mathsf{LP}} \Theta^\star \cup \Gamma^\star \cup \Delta^\star \vdash [\![t_\mu^{A^\star}(\Theta^\star \cup \Gamma^\star \cup \Delta^\star)]\!]A^\star$, where the proof polynomial $t_\mu^{A^\star}(\Theta^\star \cup \Gamma^\star \cup \Delta^\star) = c_{A^\star}^{\mathbf{A0}} \cdot t_\lambda^{\neg\neg A}(\Theta^\star \cup \Gamma^\star \cup \Delta^\star)$. Again, by Internalization, if this rule is applied within a larger derivation, we can choose $t_\lambda^{\neg\neg A}(\Theta^\star \cup \Gamma^\star \cup \Delta^\star)$ freely among all possible associated proof polynomials of $\neg\neg A$ in $\Theta^\star \cup \Gamma^\star \cup \Delta^\star$.

8. Case PlusL: the derivation ends in:

$$\frac{\Theta; \Gamma; \Delta \vdash A \,|\, s}{\Theta; \Gamma; \Delta \vdash A \,|\, s + t}\ \mathsf{PlusL}$$

By the induction hypothesis, $\rhd_{\mathsf{LP}} \Theta^\star \cup \Gamma^\star \cup \Delta^\star \vdash [\![s^\star]\!]A^\star$. Thus, by **A4** and **MP**, also $\rhd_{\mathsf{LP}} \Theta^\star \cup \Gamma^\star \cup \Delta^\star \vdash [\![s^\star + t^\star]\!]A^\star$.

9. Case PlusR: this case is analogous to the previous one.

□

Corollary 4.3.7 If $\rhd_{\mathsf{HLP}} \cdot; \cdot; \cdot \vdash A \,|\, s$, then both $\cdot \vdash [\![s^\star]\!]A^\star$ and $\cdot \vdash A^\star$ are derivable in LP.

Proof.- The former holds by Proposition 4.3.6, and the latter is obtained by using axiom **A1**. □

Corollary 4.3.8 HLP is consistent.

Proof.- The result follows from the consistency of LP. Since $\bot^\star = \bot$, if $\cdot; \cdot; \cdot \vdash \bot \,|\, s$ were derivable in HLP for some witness s, then $\cdot \vdash \bot$ would be derivable in LP. □

4.4 Term Assignment

While proof witnesses contain enough information to ensure that a formula is derivable, they do not contain enough information to reason about the derivation itself. For instance, the proof witness $v^A + w^A$ ensures that A is true if we assume both v^A and w^A, but it does not tell us which hypothesis was used in order to derive it. In fact, there are two possible derivations, one using v^A and PlusL, and the other using w^A and PlusR. Similarly, $!v^A$ can be used to verify that $[\![v^A]\!]A$ is true assuming v^A as a validity hypothesis, but this may have been derived in an infinite number of ways, using $\Box\mathsf{I}$ with any witness which is equivalent to v^A (for example, v^A itself, $(\lambda x^A.x^A) \cdot v^A$, $\mu\alpha^A.[\alpha^A]v^A$, etc.). In order to reason about the derivations and obtain results related to the metatheory, it is convenient to encode proofs as *terms* containing more information, enabling us to reason about proofs and formulas by means of their representation as terms and types.

Definition 4.4.1 (Terms of λ^{LP}) The set of terms for λ^{LP} is defined as follows:

$$
\begin{array}{rcl}
M, N & ::= & x^A \\
& | & v^A \\
& | & (\lambda x^A.M^B)^{A \supset B} \\
& | & (M^{A \supset B} N^A)^B \\
& | & (!M^A)^{[\![s]\!]A} \\
& | & (N^B \langle v^A{:=}r, M^{[\![r]\!]A} \rangle)^{B\{v^A \leftarrow r\}} \\
& | & ([\alpha^A]M^A)^{\bot} \\
& | & (\mu\alpha^A.M^{\bot})^A \\
& | & (M^A +_{\mathrm{L}} s)^A \mid (s +_{\mathrm{R}} N^B)^B
\end{array}
$$

Returning to our examples, the term $(v^A +_{\mathrm{L}} w^A)^A$ encodes a proof of A using $\mathsf{T\text{-}PlusL}$ and v^A, and not the alternative (which would be encoded by $(v^A +_{\mathrm{R}} w^A)^A$). And the term $(!((\lambda x^A.x^A)^{A \supset A} v^A)^A)^{[\![v^A]\!]A}$ encodes a proof of $[\![v^A]\!]A$ which uses $(\lambda x^A.x^A) \cdot v^A$ as a witness for the premises, and not v^A, $\mu\alpha^A.[\alpha^A]v^A$ nor any other equivalent witness. Similarly, the terms $(!v^A)^{[\![v^A]\!]A}$ and $(!v^A)^{[\![(\lambda x^A.x^A)\cdot v^A]\!]A}$ encode different derivations, which are used to prove different formulas. Hence type annotations over a ! are important.

Some information is still left out, since our terms do not encode the equivalence rules used to derive the second premise of $\mathsf{T}\text{-}\Box\mathsf{I}$, nor the contexts used in the derivations (we may have assumed additional hypotheses which were never used). However, these terms provide us with enough information to reason about the proof normalisation process and other properties of the metatheory.

Free variables of validity, truth and falsehood over terms are defined analogously to those for proof witnesses. Again, decorations may be omitted where it is safe.

Definition 4.4.2 (Typing rules for λ^{LP}) A **typing judgement** for λ^{LP} is an expression of the form $\Theta; \Gamma; \Delta \vdash M^A \mid s$. The typing rules for λ^{LP} specify the subset of typable typing judgement. These rules are presented in Fig. 4.6. We write $\rhd_{\lambda^{\mathsf{LP}}} \Theta; \Gamma; \Delta \vdash M^A \mid s$, when the judgement

$$\frac{}{\mathcal{H}, x^A \vdash x^A \,|\, x^A}\ \mathsf{T\text{-}Var} \qquad \frac{}{\mathcal{H}, v^A \vdash v^A \,|\, v^A}\ \mathsf{T\text{-}VarM}$$

$$\frac{\mathcal{H}, x^A \vdash M^B \,|\, s}{\mathcal{H} \vdash (\lambda x^A.M)^{A \supset B} \,|\, \lambda x^A.s}\ \mathsf{T\text{-}\supset I} \qquad \frac{\Theta; \cdot; \cdot \vdash M^B \,|\, s \quad \Theta; \cdot; \cdot \vdash s \equiv t : B}{\Theta; \Gamma; \Delta \vdash (!M^B)^{[\![t]\!]B} \,|\, !t}\ \mathsf{T\text{-}\Box I}$$

$$\frac{\mathcal{H} \vdash M^{A \supset B} \,|\, s \quad \mathcal{H} \vdash N^A \,|\, t}{\mathcal{H} \vdash (MN)^B \,|\, s \cdot t}\ \mathsf{T\text{-}\supset E}$$

$$\frac{\mathcal{H} \vdash M^{[\![r]\!]A} \,|\, s \quad \mathcal{H}, v^A \vdash N^C \,|\, t}{\mathcal{H} \vdash (N\langle v^A{:=}r, M\rangle)^{C\{v^A \leftarrow r\}} \,|\, t\langle v^A{:=}r, s\rangle}\ \mathsf{T\text{-}\Box E}$$

$$\frac{\mathcal{H} \vdash M^A \,|\, s}{\mathcal{H} \vdash (M +_{\mathrm{L}} t)^A \,|\, s + t}\ \mathsf{T\text{-}PlusL} \qquad \frac{\mathcal{H} \vdash N^B \,|\, t}{\mathcal{H} \vdash (s +_{\mathrm{R}} N)^B \,|\, s + t}\ \mathsf{T\text{-}PlusR}$$

$$\frac{\mathcal{H}, \alpha^A \vdash M^A \,|\, s}{\mathcal{H}, \alpha^A \vdash ([\alpha^A]M)^{\perp} \,|\, [\alpha^A]s}\ \mathsf{T\text{-}Name} \qquad \frac{\mathcal{H}, \alpha^A \vdash M^{\perp} \,|\, s}{\mathcal{H} \vdash (\mu\alpha^A.M)^A \,|\, \mu\alpha^A.s}\ \mathsf{T\text{-}NAbs}$$

Figure 4.6: Typing rules for λ^{LP}

$\Theta; \Gamma; \Delta \vdash M^A \,|\, s$ is derivable using the rules in Fig. 4.6. Similar to what we did with HLP, we write $\mathcal{H} \vdash M^A \,|\, s$ to abbreviate $\Theta; \Gamma; \Delta \vdash M^A \,|\, s$, only here $\mathcal{H}$ is a composite *typing* context.

Note that there is a correspondence between these rules and the deduction rules for HLP (removing all terms from the typing rules results in the original inference schemes). Additionally, these typing rules are syntactically driven, as the last symbol used to construct the term in the succedent of each rule is different to the others. Finally, note that the term in the succedent of each rule contains (as subterms) all the terms used in the premises of that same rule. As a result, every well-typed term encodes a derivation in HLP (modulo the equivalence rules, which are not encoded), and every HLP-derivation can be encoded by a term.

Definition 4.4.3 (Proof witness associated to a term) In order to define the reduction rules, we will first define the **associated witness** $\mathsf{w}(M^A)$ of a term M^A:

$$
\begin{array}{lcl}
\mathsf{w}(x^A) & \triangleq & x^A \\
\mathsf{w}(v^A) & \triangleq & v^A \\
\mathsf{w}((\lambda x^A.M^B)^{A\supset B}) & \triangleq & \lambda x^A.\,\mathsf{w}(M^B) \\
\mathsf{w}((M^{A\supset B}N^A)^B) & \triangleq & \mathsf{w}(M^{A\supset B})\cdot\mathsf{w}(N^A) \\
\mathsf{w}((!M^A)^{[\![s]\!]A}) & \triangleq & !s \\
\mathsf{w}((N^B\langle v^A{:=}r, M^{[\![r]\!]A}\rangle)^{B\{v^A\leftarrow r\}}) & \triangleq & \mathsf{w}(N^B)\langle v^A{:=}r, \mathsf{w}(M^{[\![r]\!]A})\rangle \\
\mathsf{w}(([\alpha^A]M^A)^\perp) & \triangleq & [\alpha^A]\,\mathsf{w}(M^A) \\
\mathsf{w}((\mu\alpha^A.M^\perp)^A) & \triangleq & \mu\alpha^A.\,\mathsf{w}(M^\perp) \\
\mathsf{w}((M^A +_{\mathrm{L}} t)^A) & \triangleq & \mathsf{w}(M^A)+t \\
\mathsf{w}((s +_{\mathrm{R}} N^B)^B) & \triangleq & s+\mathsf{w}(N^B)
\end{array}
$$

We will show that the associated witness of a term M^A is the only proof witness s such that $\Theta;\Gamma;\Delta\vdash M^A\,|\,s$ can be derived (Lemma 4.4.5).

Remark 4.4.4 Note that $\mathsf{w}((!M^A)^{[\![s]\!]A}) = !s$ rather than $!\,\mathsf{w}(M^A)$. This is to ensure that $\mathsf{w}((!M^A)^{[\![s]\!]A})$ is the same witness used in the judgement obtained through the use of T-$\Box$I.

Lemma 4.4.5 If $\Theta;\Gamma;\Delta\vdash M^B\,|\,s$ is derivable, then $s=\mathsf{w}(M^B)$.

Proof.- By induction on the derivation of $\Theta;\Gamma;\Delta\vdash M^B\,|\,s$. We consider the last step of the derivation.

- T-Var: in this case $M^B=s=\mathsf{w}(M^B)=x^B$.
- T-VarM: in this case $M^B=s=\mathsf{w}(M^B)=v^B$.
- T-$\supset$I: here $s=\lambda x^A.t$, $B=A\supset C$, $\mathsf{w}(M^B)=\lambda x^A.\,\mathsf{w}(N^C)$ and $M=(\lambda x^A.N^C)^B$ for some term N^C, where $\Theta;\Gamma,x^A;\Delta\vdash N^C\,|\,t$ is derivable by hypothesis. By induction hypothesis, $t=\mathsf{w}(N^C)$, and thus $s=\lambda x^A.\,\mathsf{w}(N^C)=\mathsf{w}(M^B)$.
- T-$\supset$E: $M^B=(M_1M_2)^B$, $\mathsf{w}(M^B)=\mathsf{w}(M_1)\cdot\mathsf{w}(M_2)$ and $s=s_1\cdot s_2$, where $\Theta;\Gamma;\Delta\vdash M_1^{A\supset B}\,|\,s_1$ and $\Theta;\Gamma;\Delta\vdash M_2^A\,|\,s_2$ are derivable by hypothesis. By induction hypothesis, $s_1=\mathsf{w}(M_1)$ and $s_2=\mathsf{w}(M_2)$. Therefore $s=\mathsf{w}(M_1)\cdot\mathsf{w}(M_2)=\mathsf{w}(M^B)$
- T-$\Box$I: $M^B=(!N^A)^{[\![t]\!]A}$, $B=[\![t]\!]A$ and $s=!t=\mathsf{w}(M^B)$.
- T-$\Box$E: in this case $M^B=(M_2\langle v^A{:=}r,M_1\rangle)^{C\{v^A\leftarrow r\}}$, $s=s_2\langle v^A{:=}r,s_1\rangle$, $B=C\{v^A\leftarrow r\}$, $\mathsf{w}(M^B)=\mathsf{w}(M_2)\langle v^A{:=}r,\mathsf{w}(M_1)\rangle$ and both $\Theta;\Gamma;\Delta\vdash M_1^{[\![r]\!]A}\,|\,s_1$ and $\Theta,v^A;\Gamma;\Delta\vdash M_2^C\,|\,s_2$ are derivable by hypothesis. From these two judgements, we can obtain the result by IH.
- T-Name: $M^B=([\alpha^A]N^A)^\perp$, $s=[\alpha^A]t$, $B=\perp$, $\mathsf{w}(M^B)=[\alpha^A]\,\mathsf{w}(N^A)$, $\Delta=\Delta',\alpha^A$ and the judgement $\Theta;\Gamma;\Delta',\alpha^A\vdash N^A\,|t$ is derivable by hypothesis. By IH, $t=\mathsf{w}(N^A)$, and thus $s=[\alpha^A]\,\mathsf{w}(N^A)=\mathsf{w}(M^B)$.
- T-NAbs: $M^B=(\mu\alpha^B.N^\perp)^B$, $s=\mu\alpha^B.t$, $\mathsf{w}(M^B)=\mu\alpha^B.\,\mathsf{w}(N^\perp)$ and the judgement $\Theta;\Gamma;\Delta,\alpha^B\vdash N^\perp\,|\,t$ is derivable by hypothesis. We can obtain the results by IH, T-NAbs and Eq-$\mu\alpha$.
- T-PlusL: $M^B=(N^B +_{\mathrm{L}} s_2)^B$, $s=s_1+s_2$, $\mathsf{w}(M^B)=\mathsf{w}(N^B)+s_2$ and $\Theta;\Gamma;\Delta\vdash N^B\,|\,s_1$ is derivable by hypothesis. By IH, $s_1=\mathsf{w}(N^B)$, and therefore $s=\mathsf{w}(N^B)+s_2=\mathsf{w}(M^B)$.
- T-PlusR: this case is analogous to the previous one.

□

Lemma 4.4.6 (Inversion) If $\Theta; \Gamma; \Delta \vdash N^D \,|\, u$ is derivable, then:

- if $N^D = x^A$, then $x^A \in \Gamma$, $u = x^A$ and $D = A$;
- if $N^D = v^A$, then $v^A \in \Theta$, $u = v^A$ and $D = A$;
- if $N^D = (\lambda x^A.M^B)^{A \supset B}$, then $\Theta; \Gamma, x^A; \Delta \vdash M^B \,|\, s$ is derivable for some s, and $u = \lambda x^A.s$ and $D = A \supset B$;
- if $N^D = (M_1^{A \supset B} M_2^A)^B$, then both $\Theta; \Gamma; \Delta \vdash M_1^{A \supset B} \,|\, s$ and $\Theta; \Gamma; \Delta \vdash M_2^A \,|\, t$ are derivable for some s and t, and $u = s \cdot t$ and $D = B$;
- if $N^D = (!M^A)^{[\![t]\!]A}$, then $\exists s$ s.t. $\Theta; \cdot; \cdot \vdash M^A \,|\, s$ is derivable, and $u = !t$, $\Theta; \cdot; \cdot \vdash s \equiv t : A$ and $D = [\![t]\!]A$;
- if $N^D = (M_2^B \langle v^A{:=}r, M_1 \rangle)^{B\{v^A \leftarrow r^A\}}$, then both $\Theta; \Gamma; \Delta \vdash M_1^{[\![r]\!]A} \,|\, s$ and $\Theta, v^A; \Gamma; \Delta \vdash M_2^B \,|\, t$ are derivable for some s and t, and $u = t\langle v^A{:=}r, s \rangle$ and $D = B\{v^A \leftarrow r^A\}$;
- if $N^D = ([\alpha^A]M^A)^{\perp}$, then $\exists\ \Delta'$, s s.t. $\Delta = \Delta', \alpha^A$, $\Theta; \Gamma; \Delta', \alpha^A \vdash M^A \,|\, s$ is derivable, and $u = [\alpha^A]s$ and $D = \perp$;
- if $N^D = (\mu\alpha^A.M^{\perp})^A$, then $\Theta; \Gamma; \Delta, \alpha^A \vdash M^{\perp} \,|\, s$ is derivable for some s, and $u = \mu\alpha^A.s$ and $D = A$;
- if $N^D = (M^A +_{\mathrm{L}} t)^A$, then $\Theta; \Gamma; \Delta \vdash M^A \,|\, s$ is derivable for some s, and $u = s + t$ and $D = A$;
- if $N^D = (s +_{\mathrm{R}} M^B)^B$, then $\Theta; \Gamma; \Delta \vdash M^B \,|\, t$ is derivable for some t, and $u = s + t$ and $D = B$.

Proof.- By structural induction on N. One and only one rule applies to each of the cases, and the only rule which can modify the witness is T-$\Box\mathsf{I}$, which replaces it by an equivalent proof witness for the same type under the same contexts. □

4.4.1 Substitutions and Reduction

We will now define the four kinds of substitutions involved in λ^{LP}: *truth variable substitution*, *validity variable substitution*, *falsehood variable renaming* and *structural substitution*. These are summarized in Fig. 4.7. In each of the substitutions depicted in the figure, the target is a λ^{LP}-term. However, we shall also need variants of these substitutions in which the target if a formula and also a proof witness. At all times we assume that bound variables are renamed to ensure that no free variable is captured or that there is no clash of variable names.

Truth variable substitution over proof witnesses ($r\{a^A \leftarrow s^A\}$) is defined recursively as follows:

Truth Variable Substitution	**Validity Variable Substitution**
$M^B\{a^A \leftarrow N^A\}$	$M^B\{v^A \leftarrow N^A; t\}$
Falsehood Variable Renaming	**Structural Substitution**
$M^B\{\alpha^A \leftarrow \beta^A\}$	$M^B(\!\!([\alpha^{A\supset C}](\bullet) \leftarrow [\beta^C](\bullet)N^A)\!\!)$

Figure 4.7: Summary of substitutions over λ^{LP}-terms

$$
\begin{array}{llcll}
y^A & \{y^A \leftarrow s\} & \triangleq & s & \\
x^B & \{y^A \leftarrow s\} & \triangleq & x^B, & \text{if } x^B \neq y^A \\
v^B & \{y^A \leftarrow s\} & \triangleq & v^B & \\
(\lambda y^A.t) & \{y^A \leftarrow s\} & \triangleq & \lambda y^A.t & \\
(\lambda x^B.t) & \{y^A \leftarrow s\} & \triangleq & \lambda x^B.(t\{y^A \leftarrow s\}), & \text{if } x^B \neq y^A \\
r \cdot t & \{y^A \leftarrow s\} & \triangleq & r\{y^A \leftarrow s\} \cdot (t\{y^A \leftarrow s\}) & \\
!t & \{y^A \leftarrow s\} & \triangleq & !t & \\
t\langle v^B{:=}u, r\rangle & \{y^A \leftarrow s\} & \triangleq & t\{y^A \leftarrow s\}\langle v^B{:=}u, r\{y^A \leftarrow s\}\rangle & \\
r + t & \{y^A \leftarrow s\} & \triangleq & (r\{y^A \leftarrow s\}) + (t\{y^A \leftarrow s\}) & \\
([\alpha^B]t) & \{y^A \leftarrow s\} & \triangleq & [\alpha^B](t\{y^A \leftarrow s\}) & \\
(\mu\alpha^B.t) & \{y^A \leftarrow s\} & \triangleq & \mu\alpha^B.(t\{y^A \leftarrow s\}) &
\end{array}
$$

The above definition is extended to terms ($M\{a^A \leftarrow N^A\}$) as follows:

$$
\begin{array}{llcl}
y^A & \{a^A \leftarrow N^A\} & \triangleq & N^A \\
x^B & \{a^A \leftarrow N^A\} & \triangleq & x^B \text{ if } x^B \neq y^A \\
v^B & \{a^A \leftarrow N^A\} & \triangleq & v^B \\
(\lambda y^A.M^B)^{A\supset B} & \{a^A \leftarrow N^A\} & \triangleq & (\lambda y^A.M^B)^{A\supset B} \\
(\lambda x^B.M^B)^{A\supset B} & \{a^A \leftarrow N^A\} & \triangleq & (\lambda x^B.(M^B\{a^A \leftarrow N^A\}))^{A\supset B} \text{ if } x^B \neq y^A \\
(M_1^{C\supset B} M_2^C)^B & \{a^A \leftarrow N^A\} & \triangleq & (M_1^{C\supset B}\{a^A \leftarrow N^A\}(M_2^C\{a^A \leftarrow N^A\}))^B \\
(!M^B)^{[\![s]\!]B} & \{a^A \leftarrow N^A\} & \triangleq & (!M^B)^{[\![s]\!]B} \\
(M_1\langle v^B{:=}u, M_2\rangle)^{B\{v^A \leftarrow u\}} & \{a^A \leftarrow N^A\} & \triangleq & (M_1\{a^A \leftarrow N^A\}\langle v^B{:=}u, M_2\{a^A \leftarrow N^A\}\rangle)^{B\{v^A \leftarrow u\}} \\
([\alpha^B]M^B)^{\perp} & \{a^A \leftarrow N^A\} & \triangleq & ([\alpha^B](M^B\{a^A \leftarrow N^A\}))^{\perp} \\
(\mu\alpha^B.M^{\perp})^B & \{a^A \leftarrow N^A\} & \triangleq & (\mu\alpha^B.(M^{\perp}\{a^A \leftarrow N^A\}))^B \\
(M^B +_{\mathrm{L}} t)^B & \{a^A \leftarrow N^A\} & \triangleq & (M^B\{a^A \leftarrow N^A\} +_{\mathrm{L}} t\{a^A \leftarrow \mathsf{w}(N^A)\})^B \\
(s +_{\mathrm{R}} M^B)^B & \{a^A \leftarrow N^A\} & \triangleq & (s\{a^A \leftarrow \mathsf{w}(N^A)\} +_{\mathrm{R}} M^B\{a^A \leftarrow N^A\})^B
\end{array}
$$

Truth variable substitution over formulas ($B\{y^A \leftarrow s^A\}$) is defined recursively as:

$$
\begin{array}{llcl}
P & \{y^A \leftarrow s\} & \triangleq & P \\
\perp & \{y^A \leftarrow s\} & \triangleq & \perp \\
B \supset C & \{y^A \leftarrow s\} & \triangleq & B\{y^A \leftarrow s\} \supset C\{y^A \leftarrow s\} \\
[\![t]\!]B & \{y^A \leftarrow s\} & \triangleq & [\![t\{y^A \leftarrow s\}]\!]B\{y^A \leftarrow s\}
\end{array}
$$

Note that substitution operates only on the immediate level: it does not affect superindices ($x^{[\![y^A]\!]A}\{y^A \leftarrow s\} = x^{[\![y^A]\!]A}$ and not $x^{[\![s]\!]A}$).

It is also important that truth and falsehood variables within the scope of a !, as well as those present in r within the term $(M_1\langle v^B{:=}r, M_2\rangle)^{B\{v^A\leftarrow r\}}$, are unaffected by substitutions. Otherwise, substitutions would be able to change the type of a term. For example, $(!((\lambda y^A.y^A)+_{\mathsf{L}} x^B))^{[\![(\lambda y^A.y^A)+x^B]\!]A\supset A}\{x^B \leftarrow z^B\}$ would become $(!((\lambda y^A.y^A)+_{\mathsf{L}} z^B))^{[\![(\lambda y^A.y^A)+z^B]\!]A\supset A}$, and $v^{A\supset A}\langle v^{A\supset A}{:=}(\lambda y^A.y^A)+_{\mathsf{L}} x^B, !((\lambda y^A.y^A)+_{\mathsf{L}} x^B)\rangle^{[\![(\lambda y^A.y^A)+x^B]\!]A\supset A}\{x^B \leftarrow z^B\}$ would become $v^{A\supset A}\langle v^{A\supset A}{:=}(\lambda y^A.y)+_{\mathsf{L}} z^B, !((\lambda y^A.y)+_{\mathsf{L}} z^B)\rangle^{[\![(\lambda y^A.y)+z^B]\!]A\supset A}$.

Validity variable substitution over formulas and proof witnesses is defined as follows:

$$
\begin{array}{lll}
P & \{v^A \leftarrow s\} & \triangleq P \\
\bot & \{v^A \leftarrow s\} & \triangleq \bot \\
(B \supset C) & \{v^A \leftarrow s\} & \triangleq B\{v^A \leftarrow s\} \supset C\{v^A \leftarrow s\} \\
[\![t]\!]B & \{v^A \leftarrow s\} & \triangleq [\![t\{v^A \leftarrow s\}]\!]B\{v^A \leftarrow s\}
\end{array}
$$

$$
\begin{array}{lll}
x^B & \{v^A \leftarrow s\} & \triangleq x^B \\
v^A & \{v^A \leftarrow s\} & \triangleq s \\
w^B & \{v^A \leftarrow s\} & \triangleq w^B \text{ if } w^B \neq v^A \\
(\lambda x^B.t) & \{v^A \leftarrow s\} & \triangleq \lambda x^B.(t\{v^A \leftarrow s\}) \\
r \cdot t & \{v^A \leftarrow s\} & \triangleq r\{v^A \leftarrow s\} \cdot (t\{v^A \leftarrow s\}) \\
!t & \{v^A \leftarrow s\} & \triangleq !(t\{v^A \leftarrow s\}) \\
t\langle w^B{:=}u, r\rangle & \{v^A \leftarrow s\} & \triangleq t\{v^A{\leftarrow} s\}\langle w^B{:=}u\{v^A{\leftarrow} s\}, r\{v^A{\leftarrow} s\}\rangle \text{ if } w^B \neq v^A \\
t\langle v^A{:=}u, r\rangle & \{v^A \leftarrow s\} & \triangleq t\langle v^A{:=}u, r\rangle \text{ (because } v^A \notin \mathsf{FVV}(r)) \\
r + t & \{v^A \leftarrow s\} & \triangleq (r\{v^A \leftarrow s\}) + (t\{v^A \leftarrow s\}) \\
([\alpha^B]t) & \{v^A \leftarrow s\} & \triangleq [\alpha^B](t\{v^A \leftarrow s\}) \\
(\mu\alpha^{B\supset C}.t) & \{v^A \leftarrow s\} & \triangleq \mu\alpha^{B\supset C}.(t\{v^A \leftarrow s\})
\end{array}
$$

The assumption that $v^A \notin \mathsf{FVV}(r)$ in the definition of $t\langle v^A{:=}u, r\rangle\{v^A \leftarrow s\}$ is based on the variable convention: a free variable cannot have the same name as a bound variable (if this were the case, then renaming the bound variable would fix the problem).

Validity variable substitution over terms is defined as follows:

$$
\begin{array}{llcl}
x^B & \{v^A \leftarrow N^A; t\} & \triangleq & x^B \\
v^A & \{v^A \leftarrow N^A; t\} & \triangleq & N^A \\
w^B & \{v^A \leftarrow N^A; t\} & \triangleq & w^B \text{ if } w^B \neq v^A \\
(\lambda x^B.M^C)^{B \supset C} & \{v^A \leftarrow N^A; t\} & \triangleq & (\lambda x^B.(M^C\{v^A \leftarrow N^A; t\}))^{B \supset (C\{v^A \leftarrow t\})} \\
(M_1^{B \supset C} M_2^B)^C & \{v^A \leftarrow N^A; t\} & \triangleq & (M_1^{B \supset C}\{v^A \leftarrow N^A; t\}(M_2^B\{v^A \leftarrow N^A; t\}))^{C\{v^A \leftarrow t\}} \\
(!M^B)^{[\![s]\!]B} & \{v^A \leftarrow N^A; t\} & \triangleq & !(M^B\{v^A \leftarrow N^A; t\})^{([\![s]\!]B)\{v^A \leftarrow t\}} \\
M_2^C\langle v^A{:=}r, M_1\rangle & \{v^A \leftarrow N^A; t\} & \triangleq & M_2^C\langle v^A{:=}r, M_1\rangle \text{ (because } v^A \notin \mathsf{FVV}(M_1^{[\![r]\!]A})) \\
(M_2^C\langle w^B{:=}r, M_1\rangle)^D\{v^A \leftarrow N^A; t\} & & \triangleq & \begin{pmatrix} M_2^C\{v^A \leftarrow N^A; t\} \\ \langle w^B{:=}r\{v^A \leftarrow t\}, M_1\{v^A \leftarrow N^A; t\}\rangle \end{pmatrix}^{D\{v^A \leftarrow t\}} \\
 & & & \text{if } w^B \neq v^A \text{ (where } D = C\{w^B \leftarrow r\}) \\
([\alpha^B]M^B)^\perp & \{v^A \leftarrow N^A; t\} & \triangleq & ([\alpha^B](M^B\{v^A \leftarrow N^A; t\}))^\perp \\
(\mu\alpha^B.M^\perp)^B & \{v^A \leftarrow N^A; t\} & \triangleq & (\mu\alpha^B.M^\perp\{v^A \leftarrow N^A; t\})^B \\
(M^B +_{\mathrm{L}} s)^B & \{v^A \leftarrow N^A; t\} & \triangleq & (M^B\{v^A \leftarrow N^A; t\} +_{\mathrm{L}} s\{v^A \leftarrow t\})^{B\{v^A \leftarrow t\}} \\
(s +_{\mathrm{R}} M_2^C)^C & \{v^A \leftarrow N^A; t\} & \triangleq & (s\{v^A \leftarrow t\} +_{\mathrm{R}} M_2^C\{v^A \leftarrow N^A; t\})^{C\{v^A \leftarrow t\}}
\end{array}
$$

Remark 4.4.7 For every formula B (resp. term M and proof witness s), for every proof witness t (resp. term N^A, proof witness t), if $v^A \notin \mathsf{FVV}(t)$, then $v^A \notin \mathsf{FVV}(B\{v^A \leftarrow t\})$ (resp. $v^A \notin \mathsf{FVV}(M\{v^A \leftarrow N^A; t\})$, $v^A \notin \mathsf{FVV}(s\{v^A \leftarrow t\})$). An analogous result holds for all other forms of substitution (truth variable substitution, falsehood variable renaming and structural substitution).

Remark 4.4.8 If $v^A \notin \mathsf{FVV}(B)$ (resp. $v^A \notin \mathsf{FVV}(M)$, $v^A \notin \mathsf{FVV}(s)$), then $B\{v^A \leftarrow t\} = B$ (resp. $M\{v^A \leftarrow N^A; t\} = M$, $s\{v^A \leftarrow t\} = s$). An analogous result holds for all other forms of substitution.

As for falsehood variables, there are two types of substitutions which apply to them. One is a simple renaming, and the other is the structural substitution, which shall be defined later on. Falsehood variable renaming over proof witnesses is defined as follows:

$$
\begin{array}{llcl}
x^B & \{\alpha^A \leftarrow \beta^A\} & \triangleq & x^B \\
v^B & \{\alpha^A \leftarrow \beta^A\} & \triangleq & v^B \\
(\lambda x^B.t) & \{\alpha^A \leftarrow \beta^A\} & \triangleq & \lambda x^B.(t\{\alpha^A \leftarrow \beta^A\}) \\
r \cdot t & \{\alpha^A \leftarrow \beta^A\} & \triangleq & r\{\alpha^A \leftarrow \beta^A\} \cdot (t\{\alpha^A \leftarrow \beta^A\}) \\
!t & \{\alpha^A \leftarrow \beta^A\} & \triangleq & !t \\
t\langle v^B{:=}u, r\rangle & \{\alpha^A \leftarrow \beta^A\} & \triangleq & t\{\alpha^A \leftarrow \beta^A\}\langle v^B{:=}u, r\{\alpha^A \leftarrow \beta^A\}\rangle \\
r + t & \{\alpha^A \leftarrow \beta^A\} & \triangleq & (r\{\alpha^A \leftarrow \beta^A\}) + (t\{\alpha^A \leftarrow \beta^A\}) \\
([\alpha^A]t) & \{\alpha^A \leftarrow \beta^A\} & \triangleq & [\beta^A](t\{\alpha^A \leftarrow \beta^A\}) \\
([\gamma^B]t) & \{\alpha^A \leftarrow \beta^A\} & \triangleq & [\gamma^B](t\{\alpha^A \leftarrow \beta^A\}) \text{ if } \gamma^B \neq \alpha^A \\
(\mu\alpha^A.t) & \{\alpha^A \leftarrow \beta^A\} & \triangleq & \mu\alpha^A.t \\
(\mu\gamma^B.t) & \{\alpha^A \leftarrow \beta^A\} & \triangleq & \mu\gamma^B.(t\{\alpha^A \leftarrow \beta^A\}) \text{ if } \gamma^B \neq \alpha^A
\end{array}
$$

The above definition is extended to terms ($M\{\alpha^A \leftarrow \beta^A\}$) analogously. Falsehood variable renaming over formulas ($D\{\alpha^A \leftarrow \beta^A\}$) is defined recursively as:

$$
\begin{array}{lll}
P \quad \{\alpha^A \leftarrow \beta^A\} & \triangleq & P \\
\bot \quad \{\alpha^A \leftarrow \beta^A\} & \triangleq & \bot \\
B \supset C\{\alpha^A \leftarrow \beta^A\} & \triangleq & B\{\alpha^A \leftarrow \beta^A\} \supset C\{\alpha^A \leftarrow \beta^A\} \\
[\![t]\!]B \ \{\alpha^A \leftarrow \beta^A\} & \triangleq & [\![t\{\alpha^A \leftarrow \beta^A\}]\!]B\{\alpha^A \leftarrow \beta^A\}
\end{array}
$$

Structural substitution $(M(\!|\,[\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)N^A\,|\!))$ is defined as follows:

$$
\begin{array}{llll}
x^D & (\!|\,[\alpha](\bullet) \leftarrow [\beta](\bullet)N\,|\!) & \triangleq & x^D \\
v^D & (\!|\,[\alpha](\bullet) \leftarrow [\beta](\bullet)N\,|\!) & \triangleq & v^D \\
\lambda x^D.M^C & (\!|\,[\alpha](\bullet) \leftarrow [\beta](\bullet)N\,|\!) & \triangleq & \lambda x^D.(M^C(\!|\,[\alpha](\bullet) \leftarrow [\beta](\bullet)N\,|\!)) \\
M_1^{D\supset C}M_2^D & (\!|\,[\alpha](\bullet) \leftarrow [\beta](\bullet)N\,|\!) & \triangleq & M_1^{D\supset C}(\!|\,[\alpha](\bullet) \leftarrow [\beta](\bullet)N\,|\!)(M_2^D(\!|\,[\alpha](\bullet) \leftarrow [\beta](\bullet)N\,|\!)) \\
(!M^D)^{[\![s]\!]B} & (\!|\,[\alpha](\bullet) \leftarrow [\beta](\bullet)N\,|\!) & \triangleq & (!M^D)^{[\![s]\!]D} \\
M\langle v^D{:=}r,M'\rangle(\!|\,[\alpha](\bullet) \leftarrow [\beta](\bullet)N\,|\!) & & \triangleq & M(\!|\,[\alpha](\bullet) \leftarrow [\beta](\bullet)N\,|\!)\langle v^D{:=}r,M'(\!|\,[\alpha](\bullet) \leftarrow [\beta](\bullet)N\,|\!)\rangle \\
M +_{\mathrm{L}} t & (\!|\,[\alpha](\bullet) \leftarrow [\beta](\bullet)N\,|\!) & \triangleq & (M(\!|\,[\alpha](\bullet) \leftarrow [\beta](\bullet)N\,|\!)) +_{\mathrm{L}} (t(\!|\,[\alpha](\bullet) \leftarrow [\beta](\bullet)\,\mathsf{w}(N)\,|\!)) \\
s +_{\mathrm{R}} N & (\!|\,[\alpha](\bullet) \leftarrow [\beta](\bullet)N\,|\!) & \triangleq & (s(\!|\,[\alpha](\bullet) \leftarrow [\beta](\bullet)\,\mathsf{w}(N)\,|\!)) +_{\mathrm{R}} (N(\!|\,[\alpha](\bullet) \leftarrow [\beta](\bullet)N\,|\!)) \\
[\alpha^{A\supset B}]M & (\!|\,[\alpha](\bullet) \leftarrow [\beta](\bullet)N\,|\!) & \triangleq & [\beta^B]MN \\
[\gamma^D]M & (\!|\,[\alpha](\bullet) \leftarrow [\beta](\bullet)N\,|\!) & \triangleq & [\gamma^D](M(\!|\,[\alpha](\bullet) \leftarrow [\beta](\bullet)N\,|\!)) \text{ if } \gamma^D \neq \alpha^{A\supset B} \\
\mu\alpha^{A\supset B}.M & (\!|\,[\alpha](\bullet) \leftarrow [\beta](\bullet)N\,|\!) & \triangleq & \mu\alpha^{A\supset B}.M \\
\mu\gamma^D.M & (\!|\,[\alpha](\bullet) \leftarrow [\beta](\bullet)N\,|\!) & \triangleq & \mu\gamma^D.(M(\!|\,[\alpha](\bullet) \leftarrow [\beta](\bullet)N\,|\!)) \text{ if } \gamma^D \neq \alpha^{A\supset B}
\end{array}
$$

Structural substitution over proof witnesses is defined analogously (ignoring the additional proof witness, replacing M and N by s and t, $+_{\mathrm{L}}$ and $+_{\mathrm{R}}$ by $+$, and term application by $\cdot$).

Lemma 4.4.9 For validity variable substitution: if $w^C \notin \mathsf{FVV}(t)$, then:

1. $s\{w^C{\leftarrow}r\}\{v^A{\leftarrow}t\} = s\{v^A{\leftarrow}t\}\{w^C{\leftarrow}r\{v^A{\leftarrow}t\}\}$.
2. $B\{w^C{\leftarrow}r\}\{v^A{\leftarrow}t\} = B\{v^A{\leftarrow}t\}\{w^C{\leftarrow}r\{v^A{\leftarrow}t\}\}$.

Analogous results hold for all other kinds of substitution, with respect to themselves and each other.

Proof.-

1. By induction on s. Most cases are straightforward. We show some sample cases.
   For validity variable substitution with respect to itself:
   - $s = v^A$: in this case $s\{w^C{\leftarrow}r\}\{v^A{\leftarrow}t\} = v^A\{v^A{\leftarrow}t\} = t$ and $s\{v^A{\leftarrow}t\}\{w^C{\leftarrow}r\{v^A{\leftarrow}t\}\} = t\{w^C{\leftarrow}r\{v^A{\leftarrow}t\}\} = t$.
   - $s = w^C$: here $s\{w^C{\leftarrow}r\}\{v^A{\leftarrow}t\} = r\{v^A{\leftarrow}t\}$ and $s\{v^A{\leftarrow}t\}\{w^C{\leftarrow}r\{v^A{\leftarrow}t\}\} = w^C\{w^C{\leftarrow}r\{v^A{\leftarrow}t\}\} = r\{v^A{\leftarrow}t\}$.
   - $s = s_2\langle v^A{:=}u, s_1\rangle$: here $s\{w^C{\leftarrow}r\}\{v^A{\leftarrow}t\} = s\{w^C{\leftarrow}r\}$, and $s\{v^A{\leftarrow}t\}\{w^C{\leftarrow}r\{v^A{\leftarrow}t\}\} = s\{w^C{\leftarrow}r\{v^A{\leftarrow}t\}\}$. By the variable convention, since v^A is bound in s, then $v^A \notin \mathsf{FVV}(r)$, and thus $\{w^C{\leftarrow}r\{v^A{\leftarrow}t\}\} = \{w^C{\leftarrow}r\}$.
   - $s = s_2\langle w^C{:=}u, s_1\rangle$: here $s\{w^C{\leftarrow}r\}\{v^A{\leftarrow}t\} = s\{v^A{\leftarrow}t\} = s\{v^A{\leftarrow}t\}\{w^C{\leftarrow}r\{v^A{\leftarrow}t\}\}$.

For truth variable substitution with respect to itself:

- $s = y^A$: in this case $s\{x^C \leftarrow r\}\{y^A \leftarrow t\} = y^A\{y^A \leftarrow t\} = t$ and $s\{y^A \leftarrow t\}\{x^C \leftarrow r\{y^A \leftarrow t\}\} = t\{x^C \leftarrow r\{y^A \leftarrow t\}\} = t$.
- $s = x^C$: here $s\{x^C \leftarrow r\}\{y^A \leftarrow t\} = r\{y^A \leftarrow t\}$ and $s\{y^A \leftarrow t\}\{x^C \leftarrow r\{y^A \leftarrow t\}\} = x^C\{x^C \leftarrow r\{y^A \leftarrow t\}\} = r\{y^A \leftarrow t\}$.
- $s = \lambda y^A.u$: here $s\{x^C \leftarrow r\}\{y^A \leftarrow t\} = s\{x^C \leftarrow r\}$, and $s\{y^A \leftarrow t\}\{x^C \leftarrow r\{y^A \leftarrow t\}\} = s\{x^C \leftarrow r\{y^A \leftarrow t\}\}$. By the variable convention, since y^A is bound in s, then $y^A \notin \mathsf{FVT}(r)$, and thus $\{x^C \leftarrow r\{y^A \leftarrow t\}\} = \{x^C \leftarrow r\}$.
- $s = \lambda x^C.u$: here $s\{x^C \leftarrow r\}\{y^A \leftarrow t\} = s\{y^A \leftarrow t\} = s\{y^A \leftarrow t\}\{x^C \leftarrow r\{y^A \leftarrow t\}\}$.

2. By induction on B, using 1. All cases are straightforward.

□

Remark 4.4.10 If $y^A \notin \mathsf{FVT}(s)$, then $s\{y^A \leftarrow t\} = s$. Analogous results hold for all forms of substitution over proof witnesses, terms and formulas.

Definition 4.4.11 (Reduction in λ^{LP}) λ^{LP}-reduction is defined as the compatible closure of the following rules. The first set of rules arises from the principal cases of normalisation of derivations, and the second from the permutative cases.

Principal rules:

$$
\begin{array}{lll}
\beta: & (\lambda x^A.M^B)N^A & \to \quad M^B\{x^A \leftarrow N^A\}, \\
\gamma: & M^B\langle v^A{:=}r,\ !N^A\rangle & \to \quad M^B\{v^A \leftarrow N^A, r\}, \quad \text{if } \mathsf{FVT}(N^A) = \mathsf{FVF}(N^A) = \emptyset \\
\mu: & [\beta^A]\mu\alpha^A.M^\perp & \to \quad M^\perp\{\alpha^A \leftarrow \beta^A\} \\
\zeta: & (\mu\alpha^{A\supset B}.M^\perp)N^A & \to \quad \mu\beta^B.M^\perp(\!|\, [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)N^A \,|\!) \\
\theta: & \mu\alpha^A.[\alpha^A]M^A & \to \quad M^A, \qquad \text{if } \alpha^A \notin \mathsf{FVF}(M^A)
\end{array}
$$

Permutative rules:

$$
\begin{array}{lll}
\psi_L: & (M^{A\supset B} +_{\mathrm{L}} t)^{A\supset B} N^A & \to \quad (M^{A\supset B} N^A)^B +_{\mathrm{L}} t \\
\psi_R: & (s +_{\mathrm{R}} M^{A\supset B})^{A\supset B} N^A & \to \quad s +_{\mathrm{R}} (M^{A\supset B} N^A)^B \\
\phi_L: & N^B\langle v^A{:=}r, (M^{[\![r]\!]A} +_{\mathrm{L}} t)\rangle & \to \quad (N^B\langle v^A{:=}r, M\rangle)^{B\{v^A \leftarrow r^A\}} +_{\mathrm{L}} t \\
\phi_R: & N^B\langle v^A{:=}r, (s +_{\mathrm{R}} M^{[\![r]\!]A})\rangle & \to \quad s +_{\mathrm{R}} (N^B\langle v^A{:=}r, M\rangle)^{B\{v^A \leftarrow r^A\}} \\
\chi_L: & [\beta^A](M^A +_{\mathrm{L}} t)^A & \to \quad ([\beta^A]M^A)^\perp +_{\mathrm{L}} t \\
\chi_R: & [\beta^B](s +_{\mathrm{R}} N^B)^B & \to \quad s +_{\mathrm{R}} ([\beta^B]N^B)^\perp \\
\iota_L: & \mu\alpha^A.(M^\perp +_{\mathrm{L}} t)^\perp & \to \quad (\mu\alpha^A.M^\perp) +_{\mathrm{L}} t, \qquad \text{if } \alpha^A \notin \mathsf{FVF}(t) \\
\iota_R: & \mu\alpha^A.(s +_{\mathrm{R}} N^\perp)^\perp & \to \quad s +_{\mathrm{R}} (\mu\alpha^A.N^\perp), \qquad \text{if } \alpha^A \notin \mathsf{FVF}(s)
\end{array}
$$

The restrictions to rules γ, θ, ι_L and ι_R prevent the creation of free variables upon reduction. Bound variables may be renamed before reduction to avoid capture.

4.4.2 Metatheoretical Results

Lemma 4.4.12 (Truth Variable Substitution) If $\Theta;\Gamma,y^A;\Delta \vdash M^B \,|\, s$ and $\Theta;\Gamma;\Delta \vdash N^A \,|\, t$ are derivable, then so is: $\Theta;\Gamma;\Delta \vdash M\{y^A \leftarrow N\}^B \,|\, s\{y^A \leftarrow t\}$.

Proof.- By induction on the derivation of $\Theta;\Gamma,y^A;\Delta \vdash M^B \,|\, s$. Keep in mind that, by Lemma 4.4.5, $\mathsf{w}(N^A) = t$. We show the most relevant cases. The complete proof can be found in Appendix B.2.

- T-Var: in this case $s = x^B = M^B$ and $\Theta;\Gamma,y^A;\Delta \vdash M^B \,|\, s{=}\Theta;\Gamma',x^B;\Delta \vdash x^B \,|\, x^B$. There are two possibilities.
  - If $x^B = y^A$, then $\Gamma = \Gamma'$, $B = A$, $M\{y^A \leftarrow N\} = N$ and $s\{y^A \leftarrow t\}$. Then $\Theta;\Gamma;\Delta \vdash M\{y^A \leftarrow N\}^B \,|\, s\{y^A \leftarrow t\} = \Theta;\Gamma;\Delta \vdash N^A \,|\, t$, which is derivable by hypothesis.
  - If $x^B \neq y^A$, then $x^B \in \Gamma$ and $M\{y^A \leftarrow N\} = x^B = s\{y^A \leftarrow t\}$. Then, by T-Var, $\Theta;\Gamma;\Delta \vdash x^B \,|\, x^B$ is derivable.
- T-$\supset$I: here $s = \lambda x^C.s'$, $B = C \supset D$ and $M = (\lambda x^C.M')^B$ for some term M', where $\Theta;\Gamma,y^A,x^C;\Delta \vdash M'^D \,|\, s'$ is derivable by hypothesis. Again, there are two possibilities:
  - If $x^C = y^A$, then $M\{y^A \leftarrow N\} = M$, $s\{y^A \leftarrow t\} = s$ and $\Gamma,y^A,x^C{=}\Gamma,x^C$. We can derive $\Theta;\Gamma;\Delta \vdash M^B \,|\, s$ from $\Theta;\Gamma,x^C;\Delta \vdash M'^D \,|\, s'$ by T-$\supset$I.
  - If $x^C \neq y^A$, then $M\{y^A \leftarrow N\} = \lambda x^C.(M'\{y^A \leftarrow N\})$ and $s\{y^A \leftarrow t\} = \lambda x^C.(s'\{y^A \leftarrow t\})$. By induction hypothesis, the judgment $\Theta;\Gamma,x^C;\Delta \vdash M'\{y^A \leftarrow N\}^D \,|\, s'\{y^A \leftarrow t\}$ is derivable. Thus, we can derive $\Theta;\Gamma;\Delta \vdash M\{y^A \leftarrow N\}^B \,|\, s\{y^A \leftarrow t\}$ by T-$\supset$I.
- T-$\supset$E: $M = (M_1M_2)^B$ and $s = s_1 \cdot s_2$, where $\Theta;\Gamma,y^A;\Delta \vdash M_1^{C\supset B} \,|\, s_1$ and $\Theta;\Gamma,y^A;\Delta \vdash M_2^C \,|\, s_2$ are derivable by hypothesis. By IH, $\Theta;\Gamma;\Delta \vdash M_1\{y^A \leftarrow N\}^{C\supset B} \,|\, s_1\{y^A \leftarrow t\}$ and $\Theta;\Gamma;\Delta \vdash M_2\{y^A \leftarrow N\}^C \,|\, s_2\{y^A \leftarrow t\}$ are also derivable. We obtain $\Theta;\Gamma;\Delta \vdash M\{y^A \leftarrow N\}^B \,|\, s\{y^A \leftarrow t\}$ by T-$\supset$E.
- T-$\Box$I: $M = (!M')^{[\![s']\!]C}$, $s = !s'$, $B = [\![s']\!]C$ and there is a proof witness r s.t. $\Theta;\cdot;\cdot \vdash M'^C \,|\, r$ and $\Theta;\cdot;\cdot \vdash r \equiv s' : C$ are derivable by hypothesis. Since $M\{y^A \leftarrow N\} = M$ and $s\{y^A \leftarrow t\} = s$, then $\Theta;\Gamma;\Delta \vdash M\{y^A \leftarrow N\}^B \,|\, s\{y^A \leftarrow t\}$ is already derivable by hypothesis.

$\Box$

Lemma 4.4.13 (Validity Variable Substitution)

1. If $\Theta,v^A;\Gamma;\Delta \vdash M^B \,|\, s$ and $\Theta;\cdot;\cdot \vdash N^A \,|\, t$ are derivable, then so is: $\Theta;\Gamma;\Delta \vdash M\{v^A \leftarrow N^A,t\}^{B\{v^A \leftarrow t\}} \,|\, s\{v^A \leftarrow t\}$.
2. If $\Theta,v^A;\Gamma;\Delta \vdash s \equiv r : B$ and $\Theta;\cdot;\cdot \vdash N^A \,|\, t$ are derivable, then so is $\Theta;\Gamma;\Delta \vdash s\{v^A \leftarrow t\} \equiv r\{v^A \leftarrow t\} : B\{v^A \leftarrow t\}$.

Proof.- By mutual induction on the derivations of $\Theta,v^A;\Gamma;\Delta \vdash M^B \,|\, s$ and $\Theta,v^A;\Gamma;\Delta \vdash s \equiv r : B$. We consider the last step of each derivation. We show the most relevant cases. The complete proofs can be found in Appendix B.2.

For 1:

- T-VarM: $M = s = w^B$. there are now two cases:
  - $w^B = v^A$: here $A = B$, $M\{v^A \leftarrow N^A, t\} = N$ and $s\{v^A \leftarrow t\} = t$. Note that, since $A = B$, then $v^A \notin \mathsf{FVV}(B)$. Then $\Theta; \Gamma; \Delta \vdash M\{v^A \leftarrow N^A, t\}^{B\{v^A \leftarrow t\}} \mid s\{v^A \leftarrow t\} = \Theta; \Gamma; \Delta \vdash N^A \mid t$, which is derivable by hypothesis.
  - $w^B \neq v^A$: here $M = s = M\{v^A \leftarrow N^A, t\} = s\{v^A \leftarrow t\} = w^B$ and $w^B \in \Theta$. By freshness condition, $v^A \notin \mathsf{FVV}(B)$ and thus $B\{v^A \leftarrow t\} = B$. By T-VarM, $\Theta; \Gamma; \Delta \vdash w^B \mid w^B$ is derivable.
- T-□I: $M = (!M')^{[\![s_1]\!]C}$, $s = !s_1$, $B = [\![s_1]\!]C$ and both $\Theta, v^A; \cdot; \cdot \vdash M'^C \mid s_2$ and $\Theta, v^A; \cdot; \cdot \vdash s_2 \equiv s_1 : C$ are derivable by hypothesis. By induction hypothesis (1), $\Theta; \cdot; \cdot \vdash M'\{v^A \leftarrow N^A, t\}^{C\{v^A \leftarrow t\}} \mid s_2\{v^A \leftarrow t\}$ is also derivable and, by IH (2), so is $\Theta; \cdot; \cdot \vdash s_2\{v^A \leftarrow t\} \equiv s_1\{v^A \leftarrow t\} : C\{v^A \leftarrow t\}$. We can obtain $\Theta; \Gamma; \Delta \vdash M\{v^A \leftarrow N^A, t\}^{([\![s_1]\!]C)\{v^A \leftarrow t\}} \mid s\{v^A \leftarrow t\}$ by T-□I.
- T-□E: $M = (N'\langle w^D := r, M'\rangle)^{C\{w^D \leftarrow r\}}$, $s = s_2\langle w^D := r, s_1\rangle$, $B = C\{w^D \leftarrow r\}$, and the judgments $\Theta, v^A; \Gamma; \Delta \vdash M'^{[\![r]\!]D} \mid s_1$ and $\Theta, v^A, w^D; \Gamma; \Delta \vdash N'^C \mid s_2$ are derivable. There are now two cases to consider:
  - $w^D = v^A$: here $M\{v^A \leftarrow N, t\} = M$, $s\{v^A \leftarrow t\} = s$, $\Theta, v^A, w^D = \Theta, v^A$ and $B\{v^A \leftarrow t\} = B$ (since $w^D = v^A$ and $B = C\{w^D \leftarrow r\}$, this means that $v^A \notin \mathsf{FVV}(B)$). We can derive $\Theta, v^A; \Gamma; \Delta \vdash M^B \mid s$ from $\Theta, v^A; \Gamma; \Delta \vdash M'^{[\![r]\!]D} \mid s_1$ and $\Theta, v^A, w^D; \Gamma; \Delta \vdash N'^C \mid s_2$ by T-□E, and obtain $\Theta; \Gamma; \Delta \vdash M^B \mid s$ by Strengthening (since v^A is not free in M^B nor in s).
  - $w^D \neq v^A$: in this case $M\{v^A \leftarrow N, t\} = (N'\{v^A \leftarrow N, t\}$ $\langle w^D := r\{v^A \leftarrow t\}, M'\{v^A \leftarrow N, t\}\rangle)^{C\{w^D \leftarrow r\}\{v^A \leftarrow t\}}$, and $s\{v^A \leftarrow t\} =$ $s_2\{v^A \leftarrow t\}\langle w^D := r\{v^A \leftarrow t\}, s_1\{v^A \leftarrow t\}\rangle$. Note that, by freshness convention, $v^A \notin \mathsf{FVV}(D)$, and thus $([\![r]\!]D)\{v^A \leftarrow t\} = [\![r\{v^A \leftarrow t\}]\!]D$. Also, by the variable convention, $w^D \notin \mathsf{FVV}(t)$ (nor $\mathsf{FVV}(N^A)$).
    By induction hypothesis (1), both $\Theta; \Gamma; \Delta \vdash M'\{v^A \leftarrow N^A, t\}^{([\![r]\!]D)\{v^A \leftarrow t\}} \mid s_1\{v^A \leftarrow t\}$ and $\Theta, w^D; \Gamma; \Delta \vdash N'\{v^A \leftarrow N^A, t\}^{C\{v^A \leftarrow t\}} \mid s_2\{v^A \leftarrow t\}$ are derivable. We can obtain $\Theta; \Gamma; \Delta \vdash M\{v^A \leftarrow N^A, t\}^{C\{v^A \leftarrow t\}\{w^D \leftarrow r\{v^A \leftarrow t\}\}} \mid s\{v^A \leftarrow t\}$ by T-□E.
    All that remains is to prove that $C\{v^A \leftarrow t\}\{w^D \leftarrow r\{v^A \leftarrow t\}\} = C\{w^D \leftarrow r\}\{v^A \leftarrow t\}$, but this holds by Lemma 4.4.9, since $w^D \notin \mathsf{FVV}(t)$.

For 2:

- Eq-β: $s = (\lambda x^C.s_1) \cdot s_2$, $r = s_1\{x^C \leftarrow s_2\}$, and both $\Theta, v^A; \Gamma, x^C; \Delta \vdash B \mid s_1$ and $\Theta, v^A; \Gamma; \Delta \vdash C \mid s_2$ are derivable by hypothesis.
  By IH (1), $\Theta; \Gamma, x^C; \Delta \vdash B\{v^A \leftarrow t\} \mid s_1\{v^A \leftarrow t\}$ and $\Theta; \Gamma; \Delta \vdash C\{v^A \leftarrow t\} \mid s_2\{v^A \leftarrow t\}$ are also derivable. Note that, by freshness convention, $C\{v^A \leftarrow t\} = C$.
  By Eq-β, we can derive $\Theta; \Gamma; \Delta \vdash (\lambda x^C.s_1\{v^A \leftarrow t\}) \cdot (s_2\{v^A \leftarrow t\}) \equiv s_1\{v^A \leftarrow t\}\{x^C \leftarrow s_2\{v^A \leftarrow t\}\} : B\{v^A \leftarrow t\}$, which, by Lemma 4.4.9, is the same as $\Theta; \Gamma; \Delta \vdash ((\lambda x^C.s_1) \cdot s_2)\{v^A \leftarrow t\} \equiv s_1\{x^C \leftarrow s_2\}\{v^A \leftarrow t\} : B\{v^A \leftarrow t\}$.
- Eq-γ: $B = C\{w^D \leftarrow s_1\}$, $s = s_2\langle w^D := s_1, !s_1\rangle$, $r = s_2\{w^D \leftarrow s_1\}$ and both $\Theta, v^A; \cdot; \cdot \vdash D \mid s_1$ and $\Theta, v^A, w^D; \Gamma; \Delta \vdash C \mid s_2$ are derivable by hypothesis.

There are two possibilities:

- $v^A = w^D$: in this case $s\{v^A \leftarrow t\} = s$, $r\{v^A \leftarrow t\} = r$ and $B\{v^A \leftarrow t\} = B$. $\Theta, v^A; \Gamma; \Delta \vdash s \equiv r : B$ is already derivable by hypothesis and, by Strengthening (since $v^A \notin \mathsf{FVV}(s)$), so is $\Theta; \Gamma; \Delta \vdash s \equiv r : B$.
- $v^A \neq w^D$: here, by IH (1), we can derive both $\Theta; \cdot; \cdot \vdash D\{v^A \leftarrow t\} \mid s_1\{v^A \leftarrow t\}$ and $\Theta, w^D; \Gamma; \Delta \vdash C\{v^A \leftarrow t\} \mid s_2\{v^A \leftarrow t\}$.
  By Eq-γ, $\Theta; \Gamma; \Delta \vdash s_2\{v^A \leftarrow t\}\langle w^D{:=}s_1\{v^A \leftarrow t\}, !s_1\{v^A \leftarrow t\}\rangle \equiv s_2\{v^A \leftarrow t\}\{w^D \leftarrow s_1\{v^A \leftarrow t\}\} : C\{v^A \leftarrow t\}\{w^D \leftarrow s_1\{v^A \leftarrow t\}\}$ is derivable. And this is the same as $\Theta; \Gamma; \Delta \vdash (s_2\langle w^D{:=}s_1, !s_1\rangle)\{v^A \leftarrow t\} \equiv (s_2\{w^D \leftarrow s_1\})\{v^A \leftarrow t\} : C\{w^D \leftarrow s_1\}\{v^A \leftarrow t\}$ by Lemma 4.4.9.

- Eq-ζ: $s = (\mu\alpha^{C \supset B}.s_1) \cdot s_2$, $r = \mu\beta^B.s_1(\!|\, [\alpha^{C \supset B}](\bullet) \leftarrow [\beta^B](\bullet)s_2 \,|\!)$ and both $\Theta, v^A; \Gamma; \Delta, \alpha^{C \supset B} \vdash \bot \mid s_1$ and $\Theta, v^A; \Gamma; \Delta \vdash C \mid s_2$ are derivable by hypothesis. By IH (1), we can derive $\Theta; \Gamma; \Delta, \alpha^{C \supset B} \vdash \bot \mid s_1\{v^A \leftarrow t\}$ and $\Theta; \Gamma; \Delta \vdash C\{v^A \leftarrow t\} \mid s_2\{v^A \leftarrow t\}$. Additionally, by freshness convention, $B\{v^A \leftarrow t\} = B$ and $C\{v^A \leftarrow t\} = C$. By Eq-ζ, $\Theta; \Gamma; \Delta \vdash (\mu\alpha^{C \supset B}.s_1\{v^A \leftarrow t\}) \cdot s_2\{v^A \leftarrow t\} \equiv \mu\beta^B.s_1\{v^A \leftarrow t\}(\!|\, [\alpha^{C \supset B}](\bullet) \leftarrow [\beta^B](\bullet)s_2\{v^A \leftarrow t\} \,|\!) : B$ is derivable, and this is the same as $\Theta; \Gamma; \Delta \vdash s\{v^A \leftarrow t\} \equiv r\{v^A \leftarrow t\} : B$ by Lemma 4.4.9.

- Eq-ϕ_L: $B = C\{w^D \leftarrow s_2\}$, $s = s_1\langle w^D{:=}s_2, (s_3 + s_4)\rangle$, $r = s_1\langle w^D{:=}s_2, s_3\rangle + s_4$, both $\Theta, v^A; \Gamma; \Delta \vdash [\![s_2]\!]D \mid s_3$ and $\Theta, w^D, v^A; \Gamma; \Delta \vdash C \mid s_1$ are derivable by hypothesis and $w^D \notin \mathsf{FVV}(s_4)$. There are two possibilities:
  - $v^A = w^D$: in this case $s\{v^A \leftarrow t\} = s$, $r\{v^A \leftarrow t\} = r$ and $B\{v^A \leftarrow t\} = B$. $\Theta, v^A; \Gamma; \Delta \vdash s \equiv r : B$ is already derivable by hypothesis and, by Strengthening (since $v^A \notin \mathsf{FVV}(s)$), so is $\Theta; \Gamma; \Delta \vdash s \equiv r : B$.
  - $v^A \neq w^D$: here, by IH (1), we can derive both $\Theta; \Gamma; \Delta \vdash [\![s_2\{v^A \leftarrow t\}]\!]D \mid s_3\{v^A \leftarrow t\}$ and $\Theta, w^D; \Gamma; \Delta \vdash C\{v^A \leftarrow t\} \mid s_1\{v^A \leftarrow t\}$. Note that $D\{v^A \leftarrow t\} = D$ by freshness convention.
    By Eq-ϕ_L, we can derive:
    $\Theta; \Gamma; \Delta \vdash s\{v^A \leftarrow t\} \equiv r\{v^A \leftarrow t\} : C\{v^A \leftarrow t\}\{w^D \leftarrow s_2\{v^A \leftarrow t\}\}$, which is the same as $\Theta; \Gamma; \Delta \vdash s\{v^A \leftarrow t\} \equiv r\{v^A \leftarrow t\} : B\{v^A \leftarrow t\}$ by Lemma 4.4.9.

- Eq-$\langle\rangle$: $s = s_1\langle w^D{:=}t', s_2\rangle$, $r = r_1\langle w^D{:=}t', r_2\rangle$, $B = C\{w^D \leftarrow t'\}$ and both $\Theta, v^A; \Gamma; \Delta \vdash s_2 \equiv r_2 : [\![t']\!]D$ and $\Theta, w^D, v^A; \Gamma; \Delta \vdash s_1 \equiv r_1 : C$ are derivable by hypothesis. By freshness convention, $D\{v^A \leftarrow t\} = D$.

  There are two possibilities:
  - $v^A = w^D$: in this case $s\{v^A \leftarrow t\} = s$, $r\{v^A \leftarrow t\} = r$ and $B\{v^A \leftarrow t\} = B$. $\Theta, v^A; \Gamma; \Delta \vdash s \equiv r : B$ is already derivable by hypothesis and, by Strengthening (since $v^A \notin \mathsf{FVV}(s)$), so is $\Theta; \Gamma; \Delta \vdash s \equiv r : B$.
  - $v^A \neq w^D$: here, by IH (2), we can derive $\Theta; \Gamma; \Delta \vdash s_2\{v^A \leftarrow t\} \equiv r_2\{v^A \leftarrow t\} : [\![t'\{v^A \leftarrow t\}]\!]D$ and $\Theta, w^D; \Gamma; \Delta \vdash s_1\{v^A \leftarrow t\} \equiv r_1\{v^A \leftarrow t\} : C\{v^A \leftarrow t\}$. By Eq-$\langle\rangle$, we obtain $\Theta; \Gamma; \Delta \vdash s\{v^A \leftarrow t\} \equiv r\{v^A \leftarrow t\} : C\{v^A \leftarrow t\}\{W^D \leftarrow t'\{v^A \leftarrow t\}\}$, which is the same as $\Theta; \Gamma; \Delta \vdash s\{v^A \leftarrow t\} \equiv r\{v^A \leftarrow t\} : B\{v^A \leftarrow t\}$ by Lemma 4.4.9.

□

Lemma 4.4.14 (Validity Variable Substitution with Equivalence)
If $\Theta, v^A; \Gamma; \Delta \vdash M^B \,|\, s$, $\Theta; \cdot; \cdot \vdash N^A \,|\, r$ and $\Theta; \cdot; \cdot \vdash r \equiv t : A$ are derivable, then there exists s' such that both $\Theta; \Gamma; \Delta \vdash M^B\{v^A \leftarrow N^A, t\}^{B\{v^A \leftarrow t\}} \,|\, s'$ and $\Theta; \Gamma; \Delta \vdash s' \equiv s\{v^A \leftarrow t\} : B\{v^A \leftarrow t\}$ are derivable.

Proof.- By induction on the derivation of $\Theta; \Gamma; \Delta \vdash M^B \,|\, s$. We show the most relevant cases. The complete proof can be found in Appendix B.2.

- T-Var: $M^B = s = M\{v^A \leftarrow N^A, t\} = s\{v^A \leftarrow t\} = x^B$ and $x^B \in \Gamma$. Note that, due to the freshness condition and since $x^B \in \Gamma$, then $v^A \notin \mathsf{FVV}(B)$, and thus $B\{v^A \leftarrow t\} = B$. We know by hypothesis that $\Theta, v^A; \Gamma; \Delta \vdash x^B \,|\, s$ is derivable. By Strengthening (since $v^A \notin \mathsf{FVV}(x^B)$), so is $\Theta; \Gamma; \Delta \vdash x^B \,|\, x^B$. Take $s' = x^B$.
- T-VarM: $M^B = s = w^B$. There are now two cases:
  - $w^B = v^A$: here $A = B$, $M\{v^A \leftarrow N^A, t\} = N^A$ and $s\{v^A \leftarrow t\} = t$. Note that, since $A = B$, then $v^A \notin \mathsf{FVV}(B)$. Take $s' = r$. Then $\Theta; \Gamma; \Delta \vdash M\{v^A \leftarrow N^A, t\}^{B\{v^A \leftarrow t\}} \,|\, s' = \Theta; \Gamma; \Delta \vdash N^A \,|\, r$, which is derivable by hypothesis.
  - $w^B \neq v^A$: here $M = s = M\{v^A \leftarrow N^A, t\} = s\{v^A \leftarrow t\} = w^B$ and $w^B \in \Theta$. Take $s' = w^B$. By freshness condition, $v^A \notin \mathsf{FVV}(B)$ and thus $B\{v^A \leftarrow t\} = B$. By T-VarM, $\Theta; \Gamma; \Delta \vdash w^B \,|\, w^B$ is derivable.
- T-$\supset$I: $s = \lambda x^C.s'$, $B = C \supset D$, $M = (\lambda x^C.M_1)^B$ and $\Theta, v^A; \Gamma, x^C; \Delta \vdash M_1^D \,|\, s_1$ is derivable by hypothesis. By induction hypothesis, the judgements $\Theta; \Gamma, x^C; \Delta \vdash M_1\{v^A \leftarrow N^A, t\}^{D\{v^A \leftarrow t\}} \,|\, s_1'$ and $\Theta; \Gamma, x^C; \Delta \vdash s_1' \equiv s_1\{v^A \leftarrow t\} : D\{v^A \leftarrow t\}$ are both derivable. Take $s' = \lambda x^C.s_1'$. We obtain $\Theta; \Gamma; \Delta \vdash M\{v^A \leftarrow N^A, t\}^{B\{v^A \leftarrow t\}} \,|\, \lambda x^C.s_1'$ by T-$\supset$I, and $\Theta; \Gamma; \Delta \vdash s' \equiv s\{v^A \leftarrow t\} : A$ by Eq-λ.

  Note that $v^A \notin \mathsf{FVT}(C)$, since x^C and v^A are both present in the contexts involved in the judgement $\Theta, v^A; \Gamma, x^C; \Delta \vdash M'^D \,|\, s'$; thus $(C \supset D)\{v^A \leftarrow t\} = C \supset (D\{v^A \leftarrow t\})$.
- T-$\supset$E: $s = s_1 \cdot s_2$, $M = M_1 M_2$, and both $\Theta, v^A; \Gamma; \Delta \vdash M_1^{C \supset B} \,|\, s_1$ and $\Theta, v^A; \Gamma; \Delta \vdash M^C \,|\, s_2$ are derivable by hypothesis. By induction hypothesis, the following judgements are derivable:
  1. $\Theta; \Gamma; \Delta \vdash M_1\{v^A \leftarrow N^A, t\}^{(C\{v^A \leftarrow t\}) \supset (B\{v^A \leftarrow t\})} \,|\, s_1'$
  2. $\Theta; \Gamma; \Delta \vdash M_2\{v^A \leftarrow N^A, t\}^{C\{v^A \leftarrow t\}} \,|\, s_2'$
  3. $\Theta; \Gamma; \Delta \vdash s_1' \equiv s_1\{v^A \leftarrow t\} : (C\{v^A \leftarrow t\}) \supset (B\{v^A \leftarrow t\})$
  4. $\Theta; \Gamma; \Delta \vdash s_2' \equiv s_2\{v^A \leftarrow t\} : C\{v^A \leftarrow t\}$

  Take $s' = s\{v^A \leftarrow r\} = s_1\{v^A \leftarrow r\} \cdot s_2\{v^A \leftarrow t\}$.

  By T-$\supset$E from 1. and 2., we can derive:
  $\Theta; \Gamma; \Delta \vdash M_1\{v^A \leftarrow N^A, t\} M_2\{v^A \leftarrow N^A, t\}^{B\{v^A \leftarrow t\}} \,|\, s_1' \cdot s_2'$.

  And by Eq-$\cdot$ from 3. and 4.:
  $\Theta; \Gamma; \Delta \vdash s_1' \cdot s_2' \equiv s_1\{v^A \leftarrow r\} \cdot s_2\{v^A \leftarrow t\} : B\{v^A \leftarrow t\}$.
- T-$\Box$I: $M = (!M_1)^{[\![s_1]\!]C}$, $s = !s_1$, $B = [\![s_1]\!]C$ and both $\Theta, v^A; \cdot; \cdot \vdash M_1^C \,|\, s_2$ and $\Theta, v^A; \cdot; \cdot \vdash s_2 \equiv s_1 : C$ are derivable by hypothesis.

By IH, we can derive both $\Theta; \cdot; \cdot \vdash M_1\{v^A \leftarrow N^A, t\}^{C\{v^A \leftarrow t\}} \mid s_2'$ and $\Theta; \cdot; \cdot \vdash s_2' \equiv s_2\{v^A \leftarrow t\} : C\{v^A \leftarrow t\}$.
Since $\Theta, v^A; \cdot; \cdot \vdash s_2 \equiv s_1 : C$ is derivable, then by Lemma 4.4.13 so is $\Theta; \cdot; \cdot \vdash s_2\{v^A \leftarrow t\} \equiv s_1\{v^A \leftarrow t\} : C\{v^A \leftarrow t\}$. And, by Eq-Trans, $\Theta; \cdot; \cdot \vdash s_2' \equiv s_1\{v^A \leftarrow t\} : C\{v^A \leftarrow t\}$.
Take $s' = s\{v^A \leftarrow t\} = !(s_1\{v^A \leftarrow t\})$. By T-□I, we obtain:
$\Theta; \Gamma; \Delta \vdash !(M_1\{v^A \leftarrow N^A, t\})^{[\![s_1\{v^A \leftarrow t\}]\!]C\{v^A \leftarrow t\}} \mid !(s_1\{v^A \leftarrow t\})$.

- T-□E: $M = (N'\langle w^D{:=}r', M'\rangle)^{C\{w^D \leftarrow r'\}}$, $s = s_2\langle w^D{:=}r', s_1\rangle$, $B = C\{w^D \leftarrow r'\}$, and the judgments $\Theta, v^A; \Gamma; \Delta \vdash M'^{[\![r']\!]D} \mid s_1$ and $\Theta, v^A, w^D; \Gamma; \Delta \vdash N'^C \mid s_2$ are derivable. Take $s' = s\{v^A \leftarrow t\}$. There are now two cases to consider:
  - $w^D = v^A$: here $M\{v^A \leftarrow N, t\} = M$, $s\{v^A \leftarrow t\} = s$, $\Theta, v^A, w^D = \Theta, v^A$ and $B\{v^A \leftarrow t\} = B$ (since $w^D = v^A$ and $B = C\{w^D \leftarrow r'\}$, this means that $v^A \notin \mathsf{FVV}(B)$). We can derive $\Theta, v^A; \Gamma; \Delta \vdash M^B \mid s$ from $\Theta, v^A; \Gamma; \Delta \vdash M'^{[\![r']\!]D} \mid s_1$ and $\Theta, v^A, w^D; \Gamma; \Delta \vdash N'^C \mid s_2$ by T-□E, and obtain $\Theta; \Gamma; \Delta \vdash M^B \mid s$ by Strengthening (since v^A is not free in either M or s).
  - $w^D \neq v^A$: in this case $M\{v^A \leftarrow N, t\} = (N'\{v^A \leftarrow N, t\}$ $\langle w^D{:=}r\{v^A \leftarrow t\}, M'\{v^A \leftarrow N, t\}\rangle)^{C\{w^D \leftarrow r\}\{v^A \leftarrow t\}}$, and $s\{v^A \leftarrow t\} =$ $s_2\{v^A \leftarrow t\}\langle w^D{:=}r\{v^A \leftarrow t\}, s_1\{v^A \leftarrow t\}\rangle$. Note that, by freshness convention, $v^A \notin \mathsf{FVV}(D)$, and thus $([\![r']\!]D)\{v^A \leftarrow t\} = [\![r'\{v^A \leftarrow t\}]\!]D$. Also, by the variable convention, $w^D \notin \mathsf{FVV}(t)$ (nor $\mathsf{FVV}(N)$).
    By IH, the following judgements are derivable for some witnesses s_3 and s_4:

    $$\begin{array}{l}\Theta; \Gamma; \Delta \vdash M'\{v^A \leftarrow N, t\}^{[\![r'\{v^A \leftarrow t\}]\!]D} \mid s_4\\ \Theta, w^D; \Gamma; \Delta \vdash N'\{v^A \leftarrow N, t\}^{C\{v^A \leftarrow t\}} \mid s_3\\ \Theta; \Gamma; \Delta \vdash s_3 \equiv s_2\{v^A \leftarrow t\} : C\{v^A \leftarrow t\}\\ \Theta, w^D; \Gamma; \Delta \vdash s_4 \equiv s_1\{v^A \leftarrow t\} : [\![r'\{v^A \leftarrow t\}]\!]D\end{array}$$

    We can use T-□E to derive:
    $\Theta; \Gamma; \Delta \vdash N'\{v^A \leftarrow N, t\}\langle w^D{:=}r'\{v^A \leftarrow t\}, M'\{v^A \leftarrow N, t\}\rangle^{C\{v^A \leftarrow t\}\{w^D \leftarrow (r'\{v^A \leftarrow t\})\}}$
    $\mid s_4\langle w^D{:=}r'\{v^A \leftarrow t\}, s_3\rangle$.
    All that remains is to prove that $C\{v^A \leftarrow t\}\{w^D \leftarrow r'\{v^A \leftarrow t\}\} = C\{w^D \leftarrow r'\}\{v^A \leftarrow t\}$, but this holds by Lemma 4.4.9, since $w^D \notin \mathsf{FVV}(t)$.
    Take $s' = s_4\langle w^D{:=}r'\{v^A \leftarrow t\}, s_3\rangle$. We can derive $\Theta; \Gamma; \Delta \vdash s' \equiv s\{v^A \leftarrow t\} : C\{w^D \leftarrow r'\}\{v^A \leftarrow t\}$ by Eq-$\langle\rangle$.

□

Lemma 4.4.15 (Falsehood Variable Renaming) If the judgement $\Theta; \Gamma; \Delta, \alpha^A, \beta^A \vdash M^B \mid s$ is derivable, then so is $\Theta; \Gamma; \Delta, \beta^A \vdash M\{\alpha^A \leftarrow \beta^A\}^B \mid s\{\alpha^A \leftarrow \beta^A\}$.

Proof.- By induction on the derivation of $\Theta; \Gamma; \Delta, \alpha^A, \beta^A \vdash M^B \mid s$. We show the most relevant cases. The complete proof can be found in Appendix B.2.

- T-Name: $M = ([\gamma^C]M')^\perp$, $s = [\gamma^C]s'$, $B = \perp$, $\Delta = \Delta', \gamma^C$. $\Theta; \Gamma; \Delta', \gamma^C, \alpha^A, \beta^A \vdash M'^C \mid s'$ is derivable by hypothesis and, by IH, $\Theta; \Gamma; \Delta', \gamma^C, \beta^A \vdash M'\{\alpha^A \leftarrow \beta^A\}^C \mid s'\{\alpha^A \leftarrow \beta^A\}$

is also derivable. Whether or not $\gamma^C = \alpha^A$, we can derive $\Theta; \Gamma; \Delta', \gamma^C, \beta^A \vdash M\{\alpha^A \leftarrow \beta^A\}^\perp \,|\, s'\{\alpha^A \leftarrow \beta^A\}$ from $\Theta; \Gamma; \Delta', \gamma^C, \beta^A \vdash M'\{\alpha^A \leftarrow \beta^A\}^C \,|\, s\{\alpha^A \leftarrow \beta^A\}$ by T-Name.

- T-NAbs: $M = (\mu\gamma^B.M')^B$, $s = \mu\gamma^B.s'$ and $\Theta; \Gamma; \Delta, \alpha^A, \beta^A, \gamma^B \vdash M'^\perp \,|\, s'$ is derivable by hypothesis.

  There are two possibilities:

  - $\gamma^B = \alpha^A$: in this case $M\{\alpha^A \leftarrow \beta^A\} = M$ and $s\{\alpha^A \leftarrow \beta^A\} = s$. We can obtain $\Theta; \Gamma; \Delta, \beta^A \vdash M^A \,|\, s$ from $\Theta; \Gamma; \Delta, \beta^A, \alpha^A \vdash M'^\perp \,|\, s'$ by T-NAbs.
  - $\gamma^B \neq \alpha^A$: here $\Theta; \Gamma; \Delta, \beta^A, \gamma^B \vdash M'\{\alpha^A \leftarrow \beta^A\}^\perp \,|\, s'\{\alpha^A \leftarrow \beta^A\}$ is derivable by IH, and thus we can derive $\Theta; \Gamma; \Delta, \beta^A \vdash M\{\alpha^A \leftarrow \beta^A\}^B \,|\, s\{\alpha^A \leftarrow \beta^A\}$ by T-Name.

□

Lemma 4.4.16 (Structural Substitution) If both $\Theta; \Gamma; \Delta, \alpha^{A \supset B} \vdash M^D \,|\, s$ and $\Theta; \Gamma; \Delta \vdash N^A \,|\, t$ are derivable, then so is $\Theta; \Gamma; \Delta, \beta^B \vdash M(\!|\, [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)N^A \,|\!)^D \,|\, s(\!|\, [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)t \,|\!)$.

Proof.- By induction on the derivation of $\Theta; \Gamma; \Delta \vdash M^D \,|\, s$. Keep in mind that, by Lemma 4.4.5, $\mathsf{w}(N^A) = t$. We show the most relevant cases. The complete proof can be found in Appendix B.2.

- T-Name: $M = ([\gamma^C]M')^\perp$, $s = [\gamma^C]s'$, $D = \perp$, $\Delta = \Delta_1, \gamma^C$ and the judgment $\Theta; \Gamma; \Delta_1, \gamma^C, \alpha^{A\supset B} \vdash M'^C \,|\, s'$ is derivable by hypothesis. There are two possibilities:

  - $\gamma^C = \alpha^{A\supset B}$: $C = A \supset B$, $M(\!|\, [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)N^A \,|\!) = ([\beta^B]M'N^A)^\perp$ and $s(\!|\, [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)t \,|\!) = [\beta^B]s' \cdot t$. Since $\Theta; \Gamma; \Delta \vdash N^A \,|\, t$ is derivable by hypothesis, then by Weakening so is $\Theta; \Gamma; \Delta, \beta^B \vdash N^A \,|\, t$. We can thus construct the following derivation:

$$\dfrac{\dfrac{\Theta; \Gamma; \Delta, \beta^B \vdash M'^{A\supset B} \,|\, s' \quad \Theta; \Gamma; \Delta, \beta^B \vdash N^A \,|\, t}{\Theta; \Gamma; \Delta, \beta^B \vdash (M'N^A)^B \,|\, s' \cdot t}\ \text{T-}\supset\text{E}}{\Theta; \Gamma; \Delta, \beta^B \vdash [\beta^B](M'N^A)^\perp \,|\, [\beta^B]s' \cdot t}\ \text{T-Name}$$

  - $\gamma^C \neq \alpha^{A\supset B}$: $M(\!|\, [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)N^A \,|\!) = [\gamma^C](M'(\!|\, [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)N^A \,|\!))$ and $s(\!|\, [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)t \,|\!) = [\gamma^C](s'(\!|\, [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)t \,|\!))$. The result follows by IH and T-Name.

- T-NAbs: $M = (\mu\gamma^D.M')^D$, $s = \mu\gamma^D.s'$ and $\Theta; \Gamma; \Delta, \alpha^{A\supset B}, \gamma^D \vdash M'^\perp \,|\, s'$ is derivable by hypothesis. There are two cases to consider:

  - $\gamma^D = \alpha^{A\supset B}$: here $M(\!|\, [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)N^A \,|\!) = M$ and $s(\!|\, [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)t \,|\!)$ $= s$. Since $\Theta; \Gamma; \Delta, \alpha^{A\supset B} \vdash M^D \,|\, s$ is derivable and $\alpha^{A\supset B}$ is not free in either M or s, then, by Strengthening and Weakening, we can derive $\Theta; \Gamma; \Delta, \beta^B \vdash M^D \,|\, s$.
  - $\gamma^D \neq \alpha^{A\supset B}$: $M(\!|\, [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)N^A \,|\!) = (\mu\gamma^D.M'(\!|\, [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)N^A \,|\!))^D$ and $s(\!|\, [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)t \,|\!) = \mu\gamma^D.(s'(\!|\, [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)t \,|\!))$. By IH we can derive $\Theta; \Gamma; \Delta, \beta^B, \gamma^D \vdash M'(\!|\, [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)N^A \,|\!)^\perp$ $\,|\, s'(\!|\, [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)t \,|\!)$.
    The result follows by T-NAbs.

□

Corollary 4.4.17 If $\Theta;\Gamma;\Delta,\alpha^{A\supset B}\vdash M^{\perp}\,|\,s$ and $\Theta;\Gamma;\Delta\vdash N^{A}\,|\,t$ are both derivable, then so is $\Theta;\Gamma;\Delta\vdash(\mu\beta^{B}.M^{\perp}(\!|\,[\alpha^{A\supset B}](\bullet)\leftarrow[\beta^{B}](\bullet)N^{A}\,|\!)^{\perp})^{B}\,|\,\mu\beta^{B}.s(\!|\,[\alpha^{A\supset B}](\bullet)\leftarrow[\beta^{B}](\bullet)t\,|\!)$.

Proof.- From the hypotheses, we know by Lemma 4.4.16 that the judgement $\Theta;\Gamma;\Delta,\beta^{B}\vdash M^{\perp}(\!|\,[\alpha^{A\supset B}](\bullet)\leftarrow[\beta^{B}](\bullet)N^{A}\,|\!)^{\perp}\,|\,s(\!|\,[\alpha^{A\supset B}](\bullet)\leftarrow[\beta^{B}](\bullet)t\,|\!)$ is derivable. The result follows by T-NAbs. Note that, since $\beta^{B}\neq\alpha^{A\supset B}$, then $(\mu\beta^{B}.M^{\perp})(\!|\,[\alpha](\bullet)\leftarrow[\beta](\bullet)N\,|\!)=\mu\beta^{B}.(M^{\perp}(\!|\,[\alpha](\bullet)\leftarrow[\beta](\bullet)N\,|\!))$, and analogously for s.

□

Lemma 4.4.18 If $\Theta;\Gamma;\Delta\vdash M^{A}\,|\,s$ is derivable, then $\mathsf{FVT}(s)\subseteq\mathsf{FVT}(M^{A})$ and $\mathsf{FVF}(s)\subseteq\mathsf{FVF}(M^{A})$.

Proof.- By induction on the derivation of $\Theta;\Gamma;\Delta\vdash M^{A}\,|\,s$. The T-Var and T-VarM cases are straightforward. For T-□I, $\mathsf{FVT}(M^{A})=\mathsf{FVF}(M^{A})=\mathsf{FVT}(s)=\mathsf{FVF}(s)=\cdot$. In all other cases, the result is obtained by induction hypothesis and basic set operations. □

Lemma 4.4.19 If the judgment $\Theta;\Gamma;\Delta\vdash M^{D}\,|\,s$ is derivable and $M^{D}\to N^{D}$ by reducing a redex at the root of M^{D}, then $\Theta;\Gamma;\Delta\vdash N^{D}\,|\,s'$ is derivable for some witness s' such that $\Theta;\Gamma;\Delta\vdash s\equiv s'\colon D$.

Proof.- We must consider which reduction rule was used.

- β : $M=(\lambda x^{A}.M_1^{B})M_2^{A}$, $N=M_1^{B}\{x^{A}\leftarrow M_2^{A}\}$ and, by Inversion Lemma (used twice), $D=B$, $s=(\lambda x^{A}.t)\cdot t'$ and both $\Theta;\Gamma;\Delta\vdash M_2^{A}\,|\,t'$ and $\Theta;\Gamma,x^{A};\Delta\vdash M_1^{B}\,|\,t$ are derivable. By Lemma 4.4.12, we can derive $\Theta;\Gamma;\Delta\vdash M_1^{B}\{x^{A}\leftarrow M_2^{A}\}^{B}\,|\,t\{x^{A}\leftarrow t'\}$. And, by Eq-$\beta$, $\Theta;\Gamma;\Delta\vdash(\lambda x^{A}.t)\cdot t'\equiv t\{x^{A}\leftarrow t'\}\colon B$.
- γ : in this case $M=(M_1^{B}\langle v^{A}{:=}r,\,!M_2^{A}\rangle)^{B\{v^{A}\leftarrow r\}}$, $N=(M_1^{B}\{v^{A}\leftarrow M_2^{A},r\})^{B\{v^{A}\leftarrow r\}}$ and, by Inversion Lemma (twice), $D=B\{v^{A}\leftarrow r\}$, $s=t\langle v^{A}{:=}r,\,!r\rangle$ and there is a witness r' such that $\Theta;\cdot;\cdot\vdash r'\equiv r\colon A$ and both $\Theta;\cdot;\cdot\vdash M_2^{A}\,|\,r'$ and $\Theta,v^{A};\Gamma;\Delta\vdash M_1^{B}\,|\,t$ are derivable. By Lemma 4.4.14, both $\Theta;\Gamma;\Delta\vdash M_1^{B}\{v^{A}\leftarrow M_2^{A},r\}^{B\{v^{A}\leftarrow r\}}\,|\,s'$ and $\Theta;\Gamma;\Delta\vdash s'\equiv t\{v^{A}\leftarrow r\}\colon B\{v^{A}\leftarrow r\}$ are derivable for some witness s'; and, by Eq-Symm, we derive $\Theta;\Gamma;\Delta\vdash t\{v^{A}\leftarrow r\}\equiv s'\colon B\{v^{A}\leftarrow r\}$. By Eq-$\gamma$, we can derive $\Theta;\Gamma;\Delta\vdash s\equiv t\{v^{A}\leftarrow r\}\colon B\{v^{A}\leftarrow r\}$. And finally, by Eq-Trans, $\Theta;\Gamma;\Delta\vdash s\equiv s'\colon B\{v^{A}\leftarrow r\}$.
- μ : $M=[\beta^{A}]\mu\alpha^{A}.M_1^{\perp}$, $N=M_1^{\perp}\{\alpha^{A}\leftarrow\beta^{A}\}$ and, by Inversion Lemma (twice), $s=[\beta^{A}]\mu\alpha^{A}.t$, $D=\perp$, $\Delta=\Delta',\beta^{A}$ and both $\Theta;\Gamma;\Delta',\beta^{A}\vdash(\mu\alpha^{A}.M_1^{\perp})^{A}\,|\,\mu\alpha^{A}.t$ and $\Theta;\Gamma;\Delta',\alpha^{A},\beta^{A}\vdash M_1^{\perp}\,|\,t$ are derivable. Then, by Lemma 4.4.15, $\Theta;\Gamma;\Delta',\beta^{A}\vdash M_1^{\perp}\{\alpha^{A}\leftarrow\beta^{A}\}^{\perp}\,|\,t\{\alpha^{A}\leftarrow\beta^{A}\}$ is also derivable. And, by Eq-μ, $\Theta;\Gamma;\Delta',\beta^{A}\vdash[\beta^{A}]\mu\alpha^{A}.t\equiv t\{\alpha^{A}\leftarrow\beta^{A}\}\colon\perp$.
- ζ : $M=(\mu\alpha^{A\supset B}.M_1^{\perp})^{A\supset B}M_2^{A}$, $N=\mu\beta^{B}.M_1^{\perp}(\!|\,[\alpha^{A\supset B}](\bullet)\leftarrow[\beta^{B}](\bullet)M_2^{A}\,|\!)$ and, by Inversion Lemma (twice), $D=B$, $s=(\mu\alpha^{A\supset B}.s_1)\cdot s_2$ and the judgements $\Theta;\Gamma;\Delta\vdash M_2^{A}\,|\,s_2$, $\Theta;\Gamma;\Delta\vdash(\mu\alpha^{A\supset B}.M_1)^{A\supset B}\,|\,\mu\alpha^{A\supset B}.s_1$ and $\Theta;\Gamma;\Delta,\alpha^{A\supset B}\vdash M_1^{\perp}\,|\,s_1$ are derivable. By Corollary 4.4.17, we can derive $\Theta;\Gamma;\Delta\vdash(\mu\beta^{B}.M_1^{\perp}(\!|\,[\alpha^{A\supset B}](\bullet)\leftarrow[\beta^{B}](\bullet)M_2^{A}\,|\!))^{B}\,|\,\mu\beta^{B}.s_1(\!|\,[\alpha^{A\supset B}](\bullet)\leftarrow[\beta^{B}](\bullet)s_2\,|\!)$. And by Eq-$\zeta$, $\Theta;\Gamma;\Delta\vdash(\mu\alpha^{A\supset B}.s_1)\cdot s_2\equiv s_1(\!|\,[\alpha^{A\supset B}](\bullet)\leftarrow[\beta^{B}](\bullet)s_2\,|\!)\colon B$.

- $\theta : M = \mu\alpha^A.[\alpha^A]M_1^A$, $N = M_1^A$ and $\alpha^A \notin \mathsf{FVF}(M^A)$. By Inversion Lemma (twice), $D = A$, $s = \mu\alpha^A.[\alpha^A]t$ and both $\Theta;\Gamma;\Delta,\alpha^A \vdash ([\alpha^A]M_1^A)^\perp \,|\, [\alpha^A]t$ and $\Theta;\Gamma;\Delta,\alpha^A \vdash M_1^A \,|\, t$ are derivable. Since $\alpha^A \notin \mathsf{FVF}(M_1^A)$ – and, by Lemma 4.4.18, $\alpha^A \notin \mathsf{FVF}(t)$ –, then $\Theta;\Gamma;\Delta \vdash M_1^A \,|\, t$ is derivable and, by Eq-Θ, $\Theta;\Gamma;\Delta \vdash \mu\alpha^A.[\alpha^A]t \equiv t : A$.
- $\psi_L : M = (M_1^{A\supset B} +_{\mathrm{L}} s_2)^{A\supset B} M_3^A$ and $N = (M_1^{A\supset B} M_3^A)^B +_{\mathrm{L}} s_2$. By Inversion Lemma (twice), $D = B$, $s = (s_1 + s_2) \cdot s_3$, and the judgements $\Theta;\Gamma;\Delta \vdash M_1^{A\supset B} \,|\, s_1$, $\Theta;\Gamma;\Delta \vdash M_3^A \,|\, s_3$ and $\Theta;\Gamma;\Delta \vdash (M_1 +_{\mathrm{L}} s_2)^{A\supset B} \,|\, s_1 + s_2$ are derivable. By T-$\supset$E, we can derive $\Theta;\Gamma;\Delta \vdash (M_1^{A\supset B} M_3^A)^B \,|\, s_1 \cdot s_3$. By T-PlusL, we obtain $\Theta;\Gamma;\Delta \vdash (M_1^{A\supset B} M_3^A) +_{\mathrm{L}} s_2^B \,|\, (s_1 \cdot s_3) + s_2$. And, by Eq-$\psi_L$, $\Theta;\Gamma;\Delta \vdash (s_1 + s_2) \cdot s_3 \equiv (s_1 \cdot s_3) + s_2 : B$.
- $\psi_R : M = (s_1 +_{\mathrm{R}} M_2^{A\supset B})^{A\supset B} M_3^A$ and $N = s_1 +_{\mathrm{R}} (M_2^{A\supset B} M_3^A)^B$. This case is analogous to the previous one.
- $\phi_L : M = L^B\langle v^A{:=}r, (M_1^{[\![r]\!]A} +_{\mathrm{L}} s_2)\rangle$ and $N = (L^B\langle v^A{:=}r, M_1\rangle)^{B\{v^A \leftarrow r^A\}} +_{\mathrm{L}} s_2$. By Inversion Lemma (twice), $D = B\{v^A \leftarrow r\}$, $s = t\langle v^A{:=}r, (s_1 + s_2)\rangle$, and all three judgements $\Theta, v^A;\Gamma;\Delta \vdash L^B \,|\, t$, $\Theta;\Gamma;\Delta \vdash (M_1^{[\![r]\!]A} +_{\mathrm{L}} s_2)^{[\![r]\!]A} \,|\, s_1 + s_2$ and $\Theta;\Gamma;\Delta \vdash M_1^{[\![r]\!]A} \,|\, s_1$ are derivable. By T-$\Box$E, $\Theta;\Gamma;\Delta \vdash (L^B\langle v^A{:=}r, M_1\rangle)^{B\{v^A \leftarrow r\}} \,|\, t\langle v^A{:=}r, s_1\rangle$. By T-PlusL, the judgment $\Theta;\Gamma;\Delta \vdash N^{B\{v^A \leftarrow r\}} \,|\, (t\langle v^A{:=}r, s_1\rangle) + s_2$ is derivable. And, by Eq-ϕ_L, so is $\Theta;\Gamma;\Delta \vdash t\langle v^A{:=}r, (s_1 + s_2)\rangle \equiv (t\langle v^A{:=}r, s_1\rangle) + s_2 : B\{v^A \leftarrow r\}$.
- ϕ_R : This case is analogous to the previous one.
- $\chi_L : M = [\beta^A](M_1^A +_{\mathrm{L}} s_2)^A$ and $N = ([\beta^A]M_1^A)^\perp +_{\mathrm{L}} s_2$. By Inversion Lemma (twice), $D = \perp$, $s = [\beta^A]s_1 + s_2$, $\Delta = \Delta', \beta^A$ and the judgements $\Theta;\Gamma;\Delta',\beta^A \vdash (M_1^A +_{\mathrm{L}} s_2)^A \,|\, s_1 + s_2$ and $\Theta;\Gamma;\Delta',\beta^A \vdash M_1^A \,|\, s_1$ are derivable. By T-Name, we can derive $\Theta;\Gamma;\Delta',\beta^A \vdash ([\beta^A]M_1^A)^\perp \,|\, [\beta^A]s_1$. By T-PlusL, we obtain $\Theta;\Gamma;\Delta',\beta^A \vdash ([\beta^A]M_1^A)^\perp +_{\mathrm{L}} s_2^\perp \,|\, ([\beta^A]s_1) + s_2$. And finally, by Eq-$\chi_L$, $\Theta;\Gamma;\Delta',\beta^A \vdash [\beta^A]s_1 + s_2 \equiv ([\beta^A]s_1) + s_2 : \perp$.
- χ_R : This case is analogous to the previous one.
- $\iota_L : M = \mu\alpha^A.(M_1^\perp +_{\mathrm{L}} s_2)^\perp$ and $N = (\mu\alpha^A.M_1^\perp)^A +_{\mathrm{L}} s_2$. By Inversion Lemma (twice), $D = A$, $s = \mu\alpha^A.s_1 + s_2$, and the judgements $\Theta;\Gamma;\Delta',\alpha^A \vdash (M_1^\perp +_{\mathrm{L}} s_2)^\perp \,|\, s_1 + s_2$ and $\Theta;\Gamma;\Delta',\alpha^A \vdash M_1^\perp \,|\, s_1$ are derivable. By T-NAbs, we can derive $\Theta;\Gamma;\Delta \vdash (\mu\alpha^A.M_1^\perp)^A \,|\, \mu\alpha^A.s_1$. By T-PlusL, we obtain $\Theta;\Gamma;\Delta \vdash (\mu\alpha^A.M_1^\perp)^A +_{\mathrm{L}} s_2^A \,|\, (\mu\alpha^A.s_1) + s_2$. And finally, by Eq-$\chi_L$, $\Theta;\Gamma;\Delta \vdash \mu\alpha^A.s_1 + s_2 \equiv (\mu\alpha^A.s_1) + s_2 : A$.
- ι_R : This case is analogous to the previous one.

  Note that in each case s and s' are the same as the witnesses s and t for the corresponding equivalence rule from Lemma 4.2.15.

□

Lemma 4.4.20 (Type Preservation) If $\Theta;\Gamma;\Delta \vdash M^B|s$ is derivable and $M^B \to N^B$, then $\Theta;\Gamma;\Delta \vdash N^B \,|\, s'$ is derivable for some witness s' such that $\Theta;\Gamma;\Delta \vdash s \equiv s' : B$.

Proof.- By induction on M^B. If the reduction takes place on the root, then the result holds by Lemma 4.4.19. We will now consider all other cases.

- If $M^B = x^B$ or $M^B = v^B$, no reduction may take place, thus the result holds trivially.

- If $M^B = (\lambda x^A.M_1^C)^{A \supset C}$: then $B = A \supset C$ and $N^B = (\lambda x^A.N_1^C)^{B'}$, where $M_1^C \to N_1^C$. By Inversion Lemma, $\Theta; \Gamma, x^A; \Delta \vdash M_1^C \,|\, t$ is derivable for some t, and $s = \lambda x^A.t$. Then, by IH, $\Theta; \Gamma, x^A; \Delta \vdash N_1^C \,|\, t'$ is derivable, where $\Theta; \Gamma, x^A; \Delta \vdash t \equiv t' : C$. By Eq-$\lambda$, we obtain $\Theta; \Gamma; \Delta \vdash \lambda x^A.t \equiv \lambda x^A.t' : A \supset C$. And, by T-$\supset$I, $\Theta; \Gamma; \Delta \vdash (\lambda x^A.N_1^C)^{A \supset C} \,|\, \lambda x^A.t'$.
- If $M^B = (M_1^{A \supset B} M_2^A)^B$. By Inversion Lemma, $s = s_1 \cdot s_2$ and both $\Theta; \Gamma; \Delta \vdash M_1^{A \supset B} \,|\, s_1$ and $\Theta; \Gamma; \Delta \vdash M_2^A \,|\, s_2$ are derivable. There are two possibilities:
  - $N^B = (N_1^{A \supset B} M_2^A)^B$, with $M_1^{A \supset B} \to N_1^{A \supset B}$: by IH, we can derive $\Theta; \Gamma; \Delta \vdash N_1^{A \supset B} \,|\, s_1'$ and $\Theta; \Gamma; \Delta \vdash s_1 \equiv s_1' : A \supset B$. We can obtain $\Theta; \Gamma; \Delta \vdash (N_1^{A \supset B} M_2^A)^B \,|\, s_1' \cdot s_2$ by T-$\supset$E, and $\Theta; \Gamma; \Delta \vdash s_1 \cdot s_2 \equiv s_1' \cdot s_2 : B$ by Eq-Refl and Eq-$\cdot$.
  - $N^B = (M_1^{A \supset B} N_2^A)^B$, with $M_2^A \to N_2^A$: this case is analogous to the previous one.
- If $M^B = (!M_1^A)^{[\![r]\!]A}$: in this case $B = [\![r]\!]A$, $N^B = (!N_1^A)^{[\![r]\!]A}$ where $M_1^A \to N_1^A$ and, by Inversion Lemma, there is a witness t such that both $\Theta; \cdot; \cdot \vdash M_1^A \,|\, t$ and $\Theta; \cdot; \cdot \vdash t \equiv r : A$ are derivable and $s = !t$.

  By IH, we can derive $\Theta; \cdot; \cdot \vdash N_1^A \,|\, t'$ and $\Theta; \cdot; \cdot \vdash t \equiv t' : A$.

  By Eq-Symm and Eq-Trans, we obtain $\Theta; \cdot; \cdot \vdash t' \equiv r : A$. And, by T-$\Box$I, $\Theta; \Gamma; \Delta \vdash (!N_1^A)^{[\![r]\!]A} \,|\, !r$. $\Theta; \Gamma; \Delta \vdash t' \equiv r : A$ is obtained by Weakening.
- If $M^B = (M_2^C \langle v^A := r, M_1 \rangle)^{C\{v^A \leftarrow r\}}$: then $B = C\{v^A \leftarrow r\}$ and, by Inversion Lemma, both $\Theta; \Gamma; \Delta \vdash M_1^{[\![r]\!]A} \,|\, s_1$ and $\Theta, v^A; \Gamma; \Delta \vdash M_2^B \,|\, s_2$ are derivable for some s_1 and s_2, and $s = s_2 \langle v^A := r, s_1 \rangle$. There are two possibilities:
  - $N^B = (M_2^C \langle v^A := r, N_1 \rangle)^{C\{v^A \leftarrow r\}}$ with $M_1^{[\![r]\!]A} \to N_1^{[\![r]\!]A}$.
  - $N^B = (N_2^C \langle v^A := r, M_1 \rangle)^{C\{v^A \leftarrow r\}}$ with $M_2^C \to N_2^C$.

  In both cases the result follows by IH, T-$\Box$E, Eq-Refl and Eq-$\langle\rangle$.
- If $M^B = ([\alpha^A]M_1^A)^\perp$: $B = \perp$, $N^B = ([\alpha^A]N_1^A)^\perp$ where $M_1^A \to N_1^A$ and, by Inversion Lemma, $s = [\alpha^A]s_1$, $\Delta = \Delta', \alpha^A$ and $\Theta; \Gamma; \Delta', \alpha^A \vdash M_1^A \,|\, s_1$ is derivable. By IH, we can derive $\Theta; \Gamma; \Delta', \alpha^A \vdash M_1^A \,|\, s_1'$ and $\Theta; \Gamma; \Delta', \alpha^A \vdash s_1 \equiv s_1' : A$. The result follows by T-Name and Eq-$[\alpha]$.
- If $M^B = (\mu\alpha^B.M_1^\perp)^B$: $N^B = (\mu\alpha^A.N_1^\perp)^B$ where $M_1^\perp \to N_1^\perp$ and, by Inversion Lemma, $\Theta; \Gamma; \Delta, \alpha^A \vdash M_1^\perp \,|\, s_1$ is derivable for some s_1, and $s = \mu\alpha^A.s_1$. By IH, we can derive $\Theta; \Gamma; \Delta, \alpha^A \vdash N_1^\perp \,|\, s_1'$ and $\Theta; \Gamma; \Delta', \alpha^A \vdash s_1 \equiv s_1' : A$. The result follows by T-NAbs and Eq-$\mu\alpha$.
- If $M^B = (M_1^B +_{\mathrm{L}} r)^B$: $N^B = (N_1^B +_{\mathrm{L}} r)^B$ where $M_1^B \to N_1^B$ and, by Inversion Lemma, $s = s_1 + r$ and $\Theta; \Gamma; \Delta \vdash M_1^B \,|\, s_1$ is derivable. The result follows by IH, T-PlusL and Eq-$+_{\mathrm{L}}$.
- If $M^B = (r +_{\mathrm{R}} M_2^B)^B$: this case is analogous to the previous one.

$\Box$

Finally, we will show the proof of Lemma 4.2.15: if $\Theta; \Gamma; \Delta \vdash s \equiv t : B$, both $\Theta; \Gamma; \Delta \vdash B \,|\, s$ and $\Theta; \Gamma; \Delta \vdash B \,|\, t$ are derivable.

Proof.- By induction on the derivation of $\Theta; \Gamma; \Delta \vdash s \equiv t : B$. We analyze the last rule used.

- Eq-β: $s = (\lambda x^C.s') \cdot t'$, $t = s'\{x^C \leftarrow t'\}$, and both $\Theta; \Gamma, x^C; \Delta \vdash B \,|\, s'$ and $\Theta; \Gamma; \Delta \vdash C \,|\, t'$ are derivable by hypothesis. Using $\supset$I on the former, we can derive $\Theta; \Gamma; \Delta \vdash C \supset B \,|\, \lambda x^C.s'$, and then obtain $\Theta; \Gamma; \Delta \vdash B \,|\, (\lambda x^C.s') \cdot t'$ by $\supset$E.

  $\Theta; \Gamma; \Delta \vdash B \,|\, t$ can be derived from the premises by Lemma 4.4.12, using the appropriate terms to encode the derivations of $\Theta; \Gamma, x^C; \Delta \vdash B \,|\, s'$ and $\Theta; \Gamma; \Delta \vdash C \,|\, t'$.
- Eq-γ: $B = C\{v^D \leftarrow s'\}$, $s = t'\langle v^D{:=}s', !s'\rangle$, $t = t'\{v^D \leftarrow s'\}$ and both $\Theta; \cdot; \cdot \vdash D \,|\, s'$ and $\Theta, v^D; \Gamma; \Delta \vdash C \,|\, t'$ are derivable by hypothesis. We can obtain $\Theta; \Gamma; \Delta \vdash B \,|\, s$ by using $\Box$I (or $\Box$I') and $\Box$E.

  $\Theta; \Gamma; \Delta \vdash B \,|\, t$ can be derived from the premises by Lemma 4.4.13.
- Eq-μ: $B = \bot$, $s = [\beta^C]\mu\alpha^C.s'$, $t = s'\{\alpha^C \leftarrow \beta^C\}$, $\Delta = \Delta', \beta^C$ and $\Theta; \Gamma; \Delta', \alpha^C, \beta^C \vdash \bot \,|\, s'$ is derivable by hypothesis. By NAbs, we can derive $\Theta; \Gamma; \Delta', \beta^C \vdash C \,|\, \mu\alpha^C.s'$, and then obtain $\Theta; \Gamma; \Delta', \beta^C \vdash \bot \,|\, [\beta^C]\mu\alpha^C.s'$ by Name.

  $\Theta; \Gamma; \Delta \vdash B \,|\, t$ can be derived from the premises by Lemma 4.4.15.
- Eq-ζ: $s = (\mu\alpha^{A \supset B}.s') \cdot t'$, $t = \mu\beta^B.s' (\!| [\alpha^{A \supset B}](\bullet) \leftarrow [\beta^B](\bullet)t' |\!)$ and both $\Theta; \Gamma; \Delta, \alpha^{A \supset B} \vdash \bot \,|\, s'$ and $\Theta; \Gamma; \Delta \vdash A \,|\, t'$ are derivable by hypothesis. By NAbs, we can derive $\Theta; \Gamma; \Delta \vdash A \supset B \,|\, \mu\alpha^{A \supset B}.s'$, and then obtain $\Theta; \Gamma; \Delta \vdash B \,|\, (\mu\alpha^{A \supset B}.s') \cdot t'$ by $\supset$E.

  $\Theta; \Gamma; \Delta \vdash B \,|\, t$ can be derived from the premises by Lemma 4.4.16.
- Eq-θ: $s = \mu\alpha^B.[\alpha^B]t$ and $\Theta; \Gamma; \Delta \vdash B \,|\, t$ is derivable by hypothesis. By Weakening, $\Theta; \Gamma; \Delta, \alpha^B \vdash B \,|\, t$ is also derivable. We can use Name and NAbs to derive $\Theta; \Gamma; \Delta \vdash B \,|\, \mu\alpha^B.[\alpha^B]t$. Note that we already know $\Theta; \Gamma; \Delta \vdash B \,|\, t$ is derivable.
- Eq-ψ_L: $s = (s' + r) \cdot t'$, $t = (s' \cdot t') +_{\mathbb{L}} r$ and both $\Theta; \Gamma; \Delta \vdash C \supset B \,|\, s'$ and $\Theta; \Gamma; \Delta \vdash C \,|\, t'$ are derivable by hypothesis. By PlusL, we can derive $\Theta; \Gamma; \Delta \vdash C \supset B \,|\, s' + r$. And then, by $\supset$E, we obtain $\Theta; \Gamma; \Delta \vdash B \,|\, s$. Conversely, using $\supset$E first, we can derive $\Theta; \Gamma; \Delta \vdash B \,|\, s' \cdot t'$. And, by PlusL, $\Theta; \Gamma; \Delta \vdash B \,|\, t$.
- Eq-ψ_R: this case is analogous to the previous one, using PlusR instead of PlusL.
- Eq-ϕ_L: $B = C\{v^D \leftarrow r\}$, $s = u\langle v^D{:=}r, (s' + t')\rangle$, $t = u\langle v^D{:=}r, s'\rangle + t'$ and both $\Theta; \Gamma; \Delta \vdash [\![r]\!]D \,|\, s'$ and $\Theta, v^D; \Gamma; \Delta \vdash C \,|\, u$ are derivable by hypothesis. $\Theta; \Gamma; \Delta \vdash B \,|\, s$ is obtained by using PlusL and then $\Box$E, while $\Theta; \Gamma; \Delta \vdash B \,|\, t$ is obtained by using $\Box$E and then PlusL.
- Eq-ϕ_R: this case is analogous to the previous one, using PlusR instead of PlusL.
- Eq-χ_L: analogous to Eq-ϕ_L and Eq-ψ_L, using PlusL and then Name to obtain $\Theta; \Gamma; \Delta \vdash B \,|\, s$, and conversely Name and then PlusL to obtain $\Theta; \Gamma; \Delta \vdash B \,|\, t$.
- Eq-χ_R: this case is analogous to the previous one, using PlusR instead of PlusL.
- Eq-ι_L: analogous to the previous cases, using PlusL and NAbs.
- Eq-ι_R: this case is analogous to the previous one, using PlusR instead of PlusL.
- Eq-Refl: $\Theta; \Gamma; \Delta \vdash B \,|\, s$ is already derivable by hypothesis.
- Eq-Symm: here $\Theta; \Gamma; \Delta \vdash B \,|\, s$ and $\Theta; \Gamma; \Delta \vdash B \,|\, t$ are derivable by IH.
- Eq-Trans: same as above.
- Eq-λ: $B = A \supset C$, $s = \lambda x^A.s'$, $t = \lambda x^A.t'$ and, by IH, both $\Theta; \Gamma, x^A; \Delta \vdash C \,|\, s'$ and $\Theta; \Gamma, x^A; \Delta \vdash C \,|\, t'$ are derivable. We obtain $\Theta; \Gamma; \Delta \vdash A \supset C \,|\, \lambda x^A.s'$ and $\Theta; \Gamma; \Delta \vdash A \supset C \,|\, \lambda x^A.t'$ by $\supset$I.

$$\frac{}{x^A : \Gamma, x^A \vdash A, \Delta}\,\mathsf{Ax} \qquad \frac{}{\texttt{unit} : \Gamma \vdash \mathbf{1}, \Delta}\,\mathsf{Unit}$$

$$\frac{M : \Gamma \vdash B, \Delta}{\lambda x^A.M : \Gamma \backslash \{x^A\} \vdash A \supset B, \Delta}\,\supset\mathsf{I} \qquad \frac{M : \Gamma \vdash A \supset B, \Delta \quad N : \Gamma \vdash A, \Delta}{MN : \Gamma \vdash B, \Delta}\,\supset\mathsf{E}$$

$$\frac{M : \Gamma \vdash A, \Delta}{[\alpha^A]M : \Gamma \vdash \Delta, \alpha^A}\,\mu_1 \qquad \frac{M : \Gamma \vdash \Delta, \alpha^A}{\mu\alpha^A.M : \Gamma \vdash A, \Delta}\,\mu_2$$

Figure 4.8: Typing rules for $\lambda\mu^1$

- All other cases (Eq-$\cdot$, Eq-$\langle\rangle$, Eq-$[\alpha]$, Eq-$\mu\alpha$, Eq-$\dashv_{\mathrm{L}}$ and Eq-$\dashv_{\mathrm{R}}$) are analogous to Eq-λ, using the IH and the corresponding inference rule on both sides.

□

Corollary 4.4.21 If $\Theta; \Gamma; \Delta \vdash (!M^B)^A | t$ is derivable and $M^B \to N^B$, then $\Theta; \Gamma; \Delta \vdash (!N^B)^A \,|\, t$ is derivable.

Proof.- By Inversion Lemma, $A = [\![r]\!]B$, $t = !r$ for some proof witness r, and there is an s such that both $\Theta; \cdot; \cdot \vdash M^B | s$ and $\Theta; \cdot; \cdot \vdash s \equiv r : B$ are derivable. By Lemma 4.4.20, there is an s' such that both $\Theta; \cdot; \cdot \vdash N^B | s'$ and $\Theta; \cdot; \cdot \vdash s' \equiv s : B$ are derivable. By Eq-Trans, $\Theta; \cdot; \cdot \vdash s' \equiv r : B$ is also derivable. And, by T-□I, so is $\Theta; \Gamma; \Delta \vdash (!N^B)^{[\![r]\!]B} \,|\, !r$. □

4.5 Normalisation

We prove strong normalisation (SN) of λ^{LP}-reduction – and therefore proof normalisation – by mapping λ^{LP}-terms into terms of Parigot's $\lambda\mu$-calculus with unit type ($\lambda\mu^1$) and then resorting to SN of $\lambda\mu$ (Prop. 2.5.9). Since we are working with propositional logic, we will only use the pure (propositional) $\lambda\mu$-calculus, rather than its second-order extension. $\lambda\mu^1$ inherits the inference schemes and reduction rules from $\lambda\mu$ along with all its properties, since `unit` is not involved in the reduction. For clarity, we adapt Parigot's notation replacing A^x (resp. A^α) by x^A (resp. α^A).

$\lambda\mu^1$-judgements take the form $M : \Gamma \vdash \Delta$, with M a $\lambda\mu^1$-term, Γ a truth context and Δ a falsehood context. The inference rules of $\lambda\mu^1$ are given in Fig. 4.8

Remark 4.5.1 Parigot's presentation does not include $\bot$ as a formula or type. We include it for mapping purposes, assuming that $\bot, \Delta = \Delta$. Also, for simplicity, we use only one Γ and only one Δ in $\supset\mathsf{E}$. By weakening and strengthening, this is equivalent to the original formulation, where different contexts were used for each premise and then joined in the succedent.

Recall the reduction rules of $\lambda\mu^1$ from Definition 2.5.8:

$$
\begin{array}{lll}
R_1: & (\lambda x^A.M)N & \to \quad M\{x^A \leftarrow N\} \\
R_2: & (\mu\alpha^{A\supset B}.M)N & \to \quad \mu\beta^B.M\{\!|\, [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)N \,|\!\} \\
S_1: & [\beta^A]\mu\alpha^A.M & \to \quad M\{\alpha^A \leftarrow \beta^A\} \\
S_2: & \mu\alpha^A.[\alpha^A]M & \to \quad M, \qquad\qquad \text{if } \alpha^A \notin \mathsf{FVF}(M)
\end{array}
$$

Note that the above rules correspond, respectively, to rules β, ζ, μ and θ in λ^{LP}.

As mentioned above, in order to prove SN of λ^{LP}, we introduce a mapping $\langle\!| \cdot |\!\rangle$, which associates types (formulas) and terms (proofs) in λ^{LP} with types and terms in $\lambda\mu^{\mathbf{1}}$. The modal type $[\![s]\!]A$ is mapped to a functional type whose domain is the unit type $\mathbf{1}$ and whose co-domain is the mapping of A. Since $\lambda\mu^{\mathbf{1}}$ has truth and falsehood variables but not validity variables, the mapping of validity variables will rely on a new set of truth variables in $\lambda\mu^{\mathbf{1}}$.

Definition 4.5.2 The mapping for λ^{LP}-types (HLP-formulas) is as follows:

$$
\begin{array}{rcl}
\langle\!| P |\!\rangle & \triangleq & P \\
\langle\!| \bot |\!\rangle & \triangleq & \bot \\
\langle\!| A \supset B |\!\rangle & \triangleq & \langle\!| A |\!\rangle \supset \langle\!| B |\!\rangle \\
\langle\!| [\![s]\!]A |\!\rangle & \triangleq & \mathbf{1} \supset \langle\!| A |\!\rangle
\end{array}
$$

That of $\lambda\mu^{\mathbf{1}}$-terms as follows:

$$
\begin{array}{rcll}
\langle\!| x^A |\!\rangle & \triangleq & x^{\langle\!| A |\!\rangle} & \\
\langle\!| v^A |\!\rangle & \triangleq & (x_v^{\mathbf{1}\supset\langle\!| A |\!\rangle})\mathtt{unit} & \\
\langle\!| (\lambda x^A.M^B)^{A\supset B} |\!\rangle & \triangleq & \lambda x^{\langle\!| A |\!\rangle}.\langle\!| M^B |\!\rangle & \\
\langle\!| (M^{A\supset B}N^B)^B |\!\rangle & \triangleq & \langle\!| M^{A\supset B} |\!\rangle\langle\!| N^B |\!\rangle & \\
\langle\!| ([\alpha^A]M^A)^\bot |\!\rangle & \triangleq & [\alpha^{\langle\!| A |\!\rangle}]\langle\!| M^A |\!\rangle & \\
\langle\!| (\mu\alpha^A.M^\bot)^A |\!\rangle & \triangleq & \mu\alpha^{\langle\!| A |\!\rangle}.\langle\!| M^\bot |\!\rangle & \\
\langle\!| (!M^A)^{[\![s]\!]A} |\!\rangle & \triangleq & \lambda x^{\mathbf{1}}.\langle\!| M^A |\!\rangle, & x^{\mathbf{1}} \text{ fresh} \\
\langle\!| (N^B\langle v^A{:=}r, M^{[\![r]\!]A}\rangle)^{B\{v^A\leftarrow r\}} |\!\rangle & \triangleq & (\lambda x_v^{\mathbf{1}\supset\langle\!| A |\!\rangle}.\langle\!| N^B |\!\rangle)\langle\!| M^{[\![r]\!]A} |\!\rangle & \\
\langle\!| (M^A +_{\mathrm{L}} t)^A |\!\rangle & \triangleq & \langle\!| M^A |\!\rangle & \\
\langle\!| (s +_{\mathrm{R}} M^A)^A |\!\rangle & \triangleq & \langle\!| M^A |\!\rangle &
\end{array}
$$

That of contexts as follows:

$$
\begin{array}{rcl}
\langle\!| \cdot |\!\rangle & \triangleq & \cdot \\
\langle\!| (\Theta, v^A) |\!\rangle & \triangleq & \langle\!| \Theta |\!\rangle, x_v^{\mathbf{1}\supset\langle\!| A |\!\rangle} \\
\langle\!| (\Gamma, x^A) |\!\rangle & \triangleq & \langle\!| \Gamma |\!\rangle, x^{\langle\!| A |\!\rangle} \\
\langle\!| (\Delta, \alpha^A) |\!\rangle & \triangleq & \langle\!| \Delta |\!\rangle, \alpha^{\langle\!| A |\!\rangle}
\end{array}
$$

Remark 4.5.3 The truth $\lambda\mu^{\mathbf{1}}$-variables which appear in the translations of different validity λ^{LP}-variables are different from each other, and from the translations of all truth λ^{LP}-variables.

Lemma 4.5.4 For every formula C, validity variable v^A and proof witness r, $\langle\!| C\{v^A \leftarrow r\} |\!\rangle = \langle\!| C |\!\rangle$.

Proof.- By structural induction on C. The only interesting case is when $C = [\![s]\!]B$. In this case, $\langle\!| C\{v^A \leftarrow r\} |\!\rangle = \langle\!| [\![s\{v^A \leftarrow r\}]\!]B\{v^A \leftarrow r\} |\!\rangle = \mathbf{1} \supset \langle\!| B\{v^A \leftarrow r\} |\!\rangle =_{\mathrm{IH}} \mathbf{1} \supset \langle\!| B |\!\rangle = \langle\!| C |\!\rangle$. $\square$

Lemma 4.5.5 If $\Theta;\Gamma;\Delta \vdash M^A \,|\, s$ is derivable in λ^{LP}, then $\langle\!| M |\!\rangle : \langle\!| \Theta |\!\rangle \cup \langle\!| \Gamma |\!\rangle \vdash \langle\!| A |\!\rangle, \langle\!| \Delta |\!\rangle$ is derivable in $\lambda\mu^{\mathbf{1}}$.

Proof.- By induction on the derivation of $\Theta;\Gamma;\Delta \vdash M^A \,|\, s$. We consider the last step of the derivation.

- Var: in this case $M = x^A$ and $\Gamma = \Gamma', x^A$. $\langle\!| M |\!\rangle = x^{\langle\!| A |\!\rangle}$ and $\langle\!| \Gamma |\!\rangle = \langle\!| \Gamma' |\!\rangle, x^{\langle\!| A |\!\rangle}$. $x^{\langle\!| A |\!\rangle} : \langle\!| \Theta |\!\rangle \cup \langle\!| \Gamma |\!\rangle \vdash \langle\!| A |\!\rangle$ is derivable in $\lambda\mu^{\mathbf{1}}$ by Ax.
- VarM: $M = v^A$, $\Theta = \Theta', v^A$, $\langle\!| M |\!\rangle = (x_v^{\mathbf{1}\supset\langle\!| A |\!\rangle})\mathtt{unit}$ and $\langle\!| \Theta |\!\rangle = \langle\!| \Theta' |\!\rangle, x_v^{\mathbf{1}\supset\langle\!| A |\!\rangle}$. We can construct the following derivation in $\lambda\mu^{\mathbf{1}}$.

$$\dfrac{\dfrac{}{x_v^{\mathbf{1}\supset\langle\!| A |\!\rangle} : \langle\!| \Theta' |\!\rangle, x_v^{\mathbf{1}\supset\langle\!| A |\!\rangle} \cup \langle\!| \Gamma |\!\rangle \vdash \mathbf{1}\supset\langle\!| A |\!\rangle, \langle\!| \Delta |\!\rangle}\,\mathsf{Ax} \qquad \dfrac{}{\mathtt{unit} : \langle\!| \Theta' |\!\rangle, x_v^{\mathbf{1}\supset\langle\!| A |\!\rangle} \cup \langle\!| \Gamma |\!\rangle \vdash \mathbf{1}, \langle\!| \Delta |\!\rangle}\,\mathsf{Unit}}{(x_v^{\mathbf{1}\supset\langle\!| A |\!\rangle})\mathtt{unit} : \langle\!| \Theta' |\!\rangle, x_v^{\mathbf{1}\supset\langle\!| A |\!\rangle} \cup \langle\!| \Gamma |\!\rangle \vdash \langle\!| A |\!\rangle, \langle\!| \Delta |\!\rangle}\,\supset\mathsf{E}$$

- $\supset$ I: here $M = (\lambda x^B.N^C)^{B\supset C}$, $A = B \supset C$, $\langle\!| M |\!\rangle = \lambda x^{\langle\!| B |\!\rangle}.\langle\!| N^C |\!\rangle$, $\langle\!| A |\!\rangle = \langle\!| B |\!\rangle \supset \langle\!| C |\!\rangle$ and, by induction hypothesis, $\langle\!| N^C |\!\rangle : \langle\!| \Theta |\!\rangle \cup \langle\!| \Gamma |\!\rangle, x^{\langle\!| B |\!\rangle} \vdash \langle\!| C |\!\rangle, \langle\!| \Delta |\!\rangle$ is derivable in $\lambda\mu^{\mathbf{1}}$. If $x^{\langle\!| B |\!\rangle} \notin \langle\!| \Gamma |\!\rangle$, then we can derive $\lambda x^{\langle\!| B |\!\rangle}.\langle\!| N^C |\!\rangle : \langle\!| \Theta |\!\rangle \cup \langle\!| \Gamma |\!\rangle \vdash \langle\!| B |\!\rangle \supset \langle\!| C |\!\rangle, \langle\!| \Delta |\!\rangle$ by $\supset$ I. Otherwise, we derive $\lambda x^{\langle\!| B |\!\rangle}.\langle\!| N^C |\!\rangle : \langle\!| \Theta |\!\rangle \cup \langle\!| \Gamma |\!\rangle \setminus \{x^{\langle\!| B |\!\rangle}\} \vdash \langle\!| B |\!\rangle \supset \langle\!| C |\!\rangle, \langle\!| \Delta |\!\rangle$ by $\supset$ I, and then obtain $\lambda x^{\langle\!| B |\!\rangle}.\langle\!| N^C |\!\rangle : \langle\!| \Theta |\!\rangle \cup \langle\!| \Gamma |\!\rangle \vdash \langle\!| B |\!\rangle \supset \langle\!| C |\!\rangle, \langle\!| \Delta |\!\rangle$ by Weakening.
- $\supset$ E: here $M = M_1^{B\supset A} M_2^B$, $\langle\!| M |\!\rangle = \langle\!| M_1^{B\supset A} |\!\rangle \langle\!| M_2^B |\!\rangle$ and, by IH, both $\langle\!| M_1^{B\supset A} |\!\rangle : \langle\!| \Theta |\!\rangle \cup \langle\!| \Gamma |\!\rangle \vdash \langle\!| B |\!\rangle \supset \langle\!| A |\!\rangle, \langle\!| \Delta |\!\rangle$ and $\langle\!| M_2^B |\!\rangle : \langle\!| \Theta |\!\rangle \cup \langle\!| \Gamma |\!\rangle \vdash \langle\!| B |\!\rangle, \langle\!| \Delta |\!\rangle$ are derivable in $\lambda\mu^{\mathbf{1}}$. We obtain $\langle\!| M_1^{B\supset A} |\!\rangle \langle\!| M_2^B |\!\rangle : \langle\!| \Theta |\!\rangle \cup \langle\!| \Gamma |\!\rangle \vdash \langle\!| A |\!\rangle, \langle\!| \Delta |\!\rangle$ by $\supset$ E.
- $\Box$I: $M = (!N^B)^{[\![r]\!]B}$, $A = [\![r]\!]B$, $\langle\!| M |\!\rangle = \lambda x^{\mathbf{1}}.\langle\!| N^B |\!\rangle$, $\langle\!| A |\!\rangle = \mathbf{1} \supset \langle\!| B |\!\rangle$ and, by IH, $\langle\!| N^B |\!\rangle : \langle\!| \Theta |\!\rangle \vdash \langle\!| B |\!\rangle$ is derivable in $\lambda\mu^{\mathbf{1}}$. We can derive $\lambda x^{\mathbf{1}}.\langle\!| N^B |\!\rangle : \langle\!| \Theta |\!\rangle \vdash \mathbf{1} \supset \langle\!| B |\!\rangle$ by Weakening and $\supset$ I, and then obtain $\lambda x^{\mathbf{1}}.\langle\!| N^B |\!\rangle : \langle\!| \Theta |\!\rangle \cup \langle\!| \Gamma |\!\rangle \vdash \mathbf{1} \supset \langle\!| B |\!\rangle, \langle\!| \Delta |\!\rangle$ by Weakening.
- $\Box$E: $(M = M_1^C\langle v^B{:=}r, M_2\rangle)^{C\{v^B \leftarrow r\}}$, $A = C\{v^B \leftarrow r\}$, $\langle\!| M |\!\rangle = \lambda x_v^{\mathbf{1}\supset\langle\!| B |\!\rangle}.\langle\!| M_2^{[\![r]\!]B} |\!\rangle \langle\!| M_1^C |\!\rangle$ and, by Lemma 4.5.4, $\langle\!| A |\!\rangle = \langle\!| C |\!\rangle$. We can construct the following derivation.

$$\dfrac{\dfrac{\dfrac{}{\langle\!| M_1 |\!\rangle : \langle\!| \Theta |\!\rangle, x_v^{\mathbf{1}\supset\langle\!| B |\!\rangle} \cup \langle\!| \Gamma |\!\rangle \vdash \langle\!| C |\!\rangle, \langle\!| \Delta |\!\rangle}\,\mathrm{IH}}{\lambda x_v^{\mathbf{1}\supset\langle\!| B |\!\rangle}.\langle\!| M_1 |\!\rangle : \langle\!| \Theta |\!\rangle \cup \langle\!| \Gamma |\!\rangle \vdash (\mathbf{1}\supset\langle\!| B |\!\rangle)\supset\langle\!| C |\!\rangle, \langle\!| \Delta |\!\rangle}\,\supset\mathsf{I} \qquad \dfrac{}{\langle\!| M_2 |\!\rangle : \langle\!| \Theta |\!\rangle \cup \langle\!| \Gamma |\!\rangle \vdash \mathbf{1}\supset\langle\!| B |\!\rangle, \langle\!| \Delta |\!\rangle}\,\mathrm{IH}}{\langle\!| M |\!\rangle : \langle\!| \Theta |\!\rangle \cup \langle\!| \Gamma |\!\rangle \vdash \langle\!| C |\!\rangle, \langle\!| \Delta |\!\rangle}\,\supset\mathsf{E}$$

- Name: $M = ([\alpha^B]N^B)^{\perp}$, $\Delta = \Delta', \alpha^B$, $A = \langle\!| A |\!\rangle = \perp$, $\langle\!| M |\!\rangle = [\alpha^{\langle\!| B |\!\rangle}]\langle\!| N^B |\!\rangle$ and $\langle\!| \Delta |\!\rangle = \langle\!| \Delta' |\!\rangle, \alpha^{\langle\!| B |\!\rangle}$. By IH, $\langle\!| N^B |\!\rangle : \langle\!| \Theta |\!\rangle \cup \langle\!| \Gamma |\!\rangle \vdash \langle\!| B |\!\rangle, \langle\!| \Delta |\!\rangle$ is derivable, thus we can derive $[\alpha^{\langle\!| B |\!\rangle}]\langle\!| N^B |\!\rangle : \langle\!| \Theta |\!\rangle \cup \langle\!| \Gamma |\!\rangle \vdash \langle\!| \Delta |\!\rangle, \alpha^{\langle\!| B |\!\rangle}$ by μ_1. And – since $\alpha^{\langle\!| B |\!\rangle} \in \langle\!| \Delta |\!\rangle$ – $\langle\!| \Delta |\!\rangle, \alpha^{\langle\!| B |\!\rangle} = \langle\!| \Delta |\!\rangle = \perp, \langle\!| \Delta |\!\rangle$.
- NAbs: $M = (\mu\alpha^A.N^{\perp})^A$, $\langle\!| M |\!\rangle = \mu\alpha^{\langle\!| A |\!\rangle}.\langle\!| N^{\perp} |\!\rangle$ and, by IH, $\langle\!| N^{\perp} |\!\rangle : \langle\!| \Theta |\!\rangle \cup \langle\!| \Gamma |\!\rangle \vdash \langle\!| \Delta |\!\rangle, \alpha^{\langle\!| A |\!\rangle}$ is derivable. We can derive $\mu\alpha^{\langle\!| A |\!\rangle}.\langle\!| N^{\perp} |\!\rangle : \langle\!| \Theta |\!\rangle \cup \langle\!| \Gamma |\!\rangle \vdash \langle\!| A |\!\rangle, \langle\!| \Delta |\!\rangle$ by μ_2.
- PlusL: $M = (N^A +_{\mathrm{L}} t)^A$ and $\langle\!| M |\!\rangle = \langle\!| N^A |\!\rangle$. $\langle\!| N^A |\!\rangle : \langle\!| \Theta |\!\rangle \cup \langle\!| \Gamma |\!\rangle \vdash \langle\!| A |\!\rangle, \langle\!| \Delta |\!\rangle$ is derivable by IH.
- PlusR: this case is analogous to the previous one.

□

Lemma 4.5.6 For every λ^{LP}-term M:

1. if $w^C \notin \mathsf{FVV}(M)$, then $x_w^{\mathbf{1}\supset\langle C\rangle} \notin \mathsf{FVT}(\langle\!| M |\!\rangle)$;
2. if $\beta^C \notin \mathsf{FVF}(M)$, then $\beta^{\langle C\rangle} \notin \mathsf{FVF}(\langle\!| M |\!\rangle)$.

Proof.- By structural induction on M. Note that a $\lambda\mu^{\mathbf{1}}$-variable named $x_v^{\mathbf{1}\supset\langle A\rangle}$ may only be obtained from the translation of a validity variable, not a truth variable.

- $M = x^A$: $\langle\!| M |\!\rangle = x^{\langle A\rangle} \neq x_w^{\mathbf{1}\supset\langle C\rangle}$. The result is immediate.
- $M = v^A \neq w^C$: $\langle\!| M |\!\rangle = (x_v^{\mathbf{1}\supset\langle A\rangle})\mathtt{unit}$. Since $x_v^{\mathbf{1}\supset\langle A\rangle} \neq x_w^{\mathbf{1}\supset\langle C\rangle}$, the result is clear.
- $M = (\lambda x^A.M_1^B)^{A\supset B}$: $\langle\!| M |\!\rangle = \lambda x^{\langle A\rangle}.\langle\!| M_1^B |\!\rangle$. Since $w^C \notin \mathsf{FVV}(M_1^B)$ and $\beta^C \notin \mathsf{FVF}(M_1^B)$, then, by IH, $x_w^{\mathbf{1}\supset\langle C\rangle} \notin \mathsf{FVT}(\langle\!| M_1^C |\!\rangle)$ and $\beta^{\langle C\rangle} \notin \mathsf{FVF}(\langle\!| M_1^C |\!\rangle)$. Therefore, $x_w^{\mathbf{1}\supset\langle C\rangle} \notin \mathsf{FVT}(\langle\!| M |\!\rangle)$ and $\beta^{\langle C\rangle} \notin \mathsf{FVF}(\langle\!| M |\!\rangle)$.
- $M = (M_1^{A\supset B} M_2^A)^B$: here $w^C \notin \mathsf{FVV}(M_1^{A\supset B}) \cup \mathsf{FVV}(M_2^A)$ and $\beta^C \notin \mathsf{FVF}(M_1^{A\supset B}) \cup \mathsf{FVF}(M_2^A)$. By IH, $x_w^{\mathbf{1}\supset\langle C\rangle} \notin \mathsf{FVT}(\langle\!| M_1^{A\supset B} |\!\rangle) \cup \mathsf{FVT}(\langle\!| M_2^A |\!\rangle)$
$= \mathsf{FVT}(\langle\!| M_1^{A\supset B} |\!\rangle \langle\!| M_2^A |\!\rangle) = \mathsf{FVT}(\langle\!| M |\!\rangle)$, and analogously for β^C.
- $M = (!M_1^A)^{[\![s]\!]A}$: here $\langle\!| M |\!\rangle = \lambda y^{\mathbf{1}}.\langle\!| M_1^A |\!\rangle$, $w^C \notin \mathsf{FVV}(M_1^A)$ and $\beta^C \notin \mathsf{FVF}(M_1^A)$. By IH, $x_w^{\mathbf{1}\supset\langle C\rangle} \notin \mathsf{FVT}(\langle\!| M_1^A |\!\rangle)$ and $\beta^{\langle C\rangle} \notin \mathsf{FVF}(\langle\!| M_1^A |\!\rangle)$. Therefore, neither is free in $\lambda y^{\mathbf{1}}.\langle\!| M_1^A |\!\rangle$.
- $M = (M_1^B \langle v^A{:=}r, M_2\rangle)^{B\{v^A \leftarrow r\}}$: $\langle\!| M |\!\rangle = (\lambda x_v^{\mathbf{1}\supset\langle A\rangle}.\langle\!| M_1^B |\!\rangle)\langle\!| M_2^{[\![r]\!]A} |\!\rangle$, either $w^C = v^A$ or $w^C \notin \mathsf{FVV}(M_2^{[\![r]\!]A}) \cup \mathsf{FVV}(M_1^B)$, and $\beta^C \notin \mathsf{FVF}(M_2^{[\![r]\!]A}) \cup \mathsf{FVF}(M_1^B)$. If $w^C = v^A$, then $x_v^{\mathbf{1}\supset\langle A\rangle} = x_w^{\mathbf{1}\supset\langle C\rangle}$, and thus $x_w^{\mathbf{1}\supset\langle C\rangle}$ is not free in $\langle\!| M |\!\rangle$ (since $x_v^{\mathbf{1}\supset\langle A\rangle} \notin \mathsf{FVT}(\langle\!| M_2^{[\![r]\!]A} |\!\rangle)$ by variable convention and IH). Otherwise, by IH, $x_w^{\mathbf{1}\supset\langle C\rangle} \notin \mathsf{FVT}(\langle\!| M_2^{[\![r]\!]A} |\!\rangle) \cup \mathsf{FVT}(\langle\!| M_1^B |\!\rangle)$. In either case, $\beta^C \notin \mathsf{FVF}(\langle\!| M_2^{[\![r]\!]A} |\!\rangle) \cup \mathsf{FVF}(\langle\!| M_1^B |\!\rangle)$ by IH Therefore, neither $\beta^{\langle C\rangle}$ nor $x_w^{\mathbf{1}\supset\langle C\rangle}$ is free in $\langle\!| M |\!\rangle$.
- $M = ([\alpha^A]M_1^A)^{\perp}$: here $\alpha^A \neq \beta^C$, $w^C \notin \mathsf{FVV}(M_1^A)$ and $\beta^C \notin \mathsf{FVF}(M_1^A)$. By IH, $x_w^{\mathbf{1}\supset\langle C\rangle} \notin \mathsf{FVT}(\langle\!| M_1^A |\!\rangle)$ and $\beta^{\langle C\rangle} \notin \mathsf{FVF}(\langle\!| M_1^A |\!\rangle)$. And, clearly, $\alpha^{\langle A\rangle} \neq \beta^{\langle C\rangle}$. Thus, neither $\beta^{\langle C\rangle}$ nor $x_w^{\mathbf{1}\supset\langle C\rangle}$ is free in $[\alpha^{\langle A\rangle}]\langle\!| M_1^A |\!\rangle = \langle\!| M |\!\rangle$.
- $M = (\mu\alpha^A.M_1^{\perp})^A$: $\langle\!| M |\!\rangle = \mu\alpha^{\langle A\rangle}.\langle\!| M_1^{\perp} |\!\rangle$, $w^C \notin \mathsf{FVV}(M_1^{\perp})$, and either $\alpha^A = \beta^C$ or $\beta^C \notin \mathsf{FVF}(M_1^C)$. If $\alpha^A = \beta^C$, then $\alpha^{\langle A\rangle} = \beta^{\langle C\rangle}$ and thus $\beta^{\langle C\rangle} \notin \mathsf{FVF}(\mu\alpha^{\langle A\rangle}.\langle\!| M_1^{\perp} |\!\rangle)$. Otherwise, by IH, $\beta^{\langle C\rangle} \notin \mathsf{FVF}(\langle\!| M_1^{\perp} |\!\rangle)$ and, again, $\beta^{\langle C\rangle} \notin \mathsf{FVF}(\mu\alpha^{\langle A\rangle}.\langle\!| M_1^{\perp} |\!\rangle)$. In either case $x_w^{\mathbf{1}\supset\langle C\rangle} \notin \mathsf{FVT}(\langle\!| M_1^{\perp} |\!\rangle)$, and therefore $x_w^{\mathbf{1}\supset\langle C\rangle} \notin \mathsf{FVT}(\langle\!| M |\!\rangle)$.
- $M = (M_1^A +_{\mathrm{L}} s)^A$: $\langle\!| M |\!\rangle = \langle\!| M_1^A |\!\rangle$. The result holds by IH.
- $M = (s +_{\mathrm{R}} M_1^B)^B$: this case is analogous to the previous one.

□

Lemma 4.5.7 For all λ^{LP}-terms M, N, for every truth variable x^A:
$\langle\!| M |\!\rangle\{x^{\langle A\rangle} \leftarrow \langle\!| N |\!\rangle\} = \langle\!| M\{x^A \leftarrow N\} |\!\rangle$.

Proof.- By structural induction on M. Keep in mind that the truth $\lambda\mu^{\mathsf{1}}$-variables which appear in the translations of different validity λ^{LP}-variables are different from each other, and from the translations of all truth λ^{LP}-variables.

- $M = x^A$: $\llparenthesis M \rrparenthesis\{x^{\langle A\rangle} \leftarrow \llparenthesis N \rrparenthesis\} = x^{\langle A\rangle}\{x^{\langle A\rangle} \leftarrow \llparenthesis N \rrparenthesis\} = \llparenthesis N \rrparenthesis = \llparenthesis x^A\{x^A \leftarrow N\} \rrparenthesis$.
- $M = y^B$ with $y^B \neq x^A$:

$$\begin{array}{ll} \llparenthesis M \rrparenthesis\{x^{\langle A\rangle} \leftarrow \llparenthesis N \rrparenthesis\} & = \\ y^{\langle B\rangle}\{x^{\langle A\rangle} \leftarrow \llparenthesis N \rrparenthesis\} & = \\ y^{\langle B\rangle} & = \\ \llparenthesis y^B \rrparenthesis & = \\ \llparenthesis y^B\{x^A \leftarrow N\} \rrparenthesis. & \end{array}$$

- $M = v^B$:

$$\begin{array}{ll} \llparenthesis M \rrparenthesis\{x^{\langle A\rangle} \leftarrow \llparenthesis N \rrparenthesis\} & = \\ ((y_v^{\mathbf{1}\supset\langle B\rangle})\mathtt{unit})\{x^{\langle A\rangle} \leftarrow \llparenthesis N \rrparenthesis\} & = \\ (y_v^{\mathbf{1}\supset\langle B\rangle}\{x^{\langle A\rangle} \leftarrow \llparenthesis N \rrparenthesis\})(\mathtt{unit}\{x^{\langle A\rangle} \leftarrow \llparenthesis N \rrparenthesis\}) & = \\ (y_v^{\mathbf{1}\supset\langle B\rangle})\mathtt{unit} & = \\ \llparenthesis v^B \rrparenthesis & = \\ \llparenthesis v^B\{x^A \leftarrow N\} \rrparenthesis. & \end{array}$$

- $M = (\lambda x^A.M_1^B)^{A\supset B}$:

$$\begin{array}{ll} \llparenthesis M \rrparenthesis\{x^{\langle A\rangle} \leftarrow \llparenthesis N \rrparenthesis\} & = \\ (\lambda x^{\langle A\rangle}.\llparenthesis M_1^B \rrparenthesis)\{x^{\langle A\rangle} \leftarrow \llparenthesis N \rrparenthesis\} & = \\ \lambda x^{\langle A\rangle}.\llparenthesis M_1^B \rrparenthesis & = \\ \llparenthesis (\lambda x^A.M_1^B)^{A\supset B} \rrparenthesis & = \\ \llparenthesis (\lambda x^A.M_1^B)^{A\supset B}\{x^A \leftarrow N\} \rrparenthesis. & \end{array}$$

- $M = (\lambda y^C.M_1^B)^{C\supset B}$ with $y^C \neq x^A$:

$$\begin{array}{ll} \llparenthesis M \rrparenthesis\{x^{\langle A\rangle} \leftarrow \llparenthesis N \rrparenthesis\} & = \\ (\lambda y^{\langle C\rangle}.\llparenthesis M_1^B \rrparenthesis)\{x^{\langle A\rangle} \leftarrow \llparenthesis N \rrparenthesis\} & = \\ \lambda y^{\langle C\rangle}.(\llparenthesis M_1^B \rrparenthesis\{x^{\langle A\rangle} \leftarrow \llparenthesis N \rrparenthesis\}) & =_{\mathrm{IH}} \\ \lambda y^{\langle C\rangle}.(\llparenthesis M_1^B\{x^A \leftarrow N\} \rrparenthesis) & = \\ \llparenthesis \lambda y^C.(M_1^B\{x^A \leftarrow N\}) \rrparenthesis & = \\ \llparenthesis (\lambda y^C.M_1^B)^{C\supset B}\{x^A \leftarrow N\} \rrparenthesis. & \end{array}$$

- $M = (M_1^{C\supset B}M_2^C)^B$:

$$\begin{array}{ll} \llparenthesis M \rrparenthesis\{x^{\langle A\rangle} \leftarrow \llparenthesis N \rrparenthesis\} & = \\ (\llparenthesis M_1^{C\supset B} \rrparenthesis\llparenthesis M_1^C \rrparenthesis)\{x^{\langle A\rangle} \leftarrow \llparenthesis N \rrparenthesis\} & = \\ (\llparenthesis M_1^{C\supset B} \rrparenthesis\{x^{\langle A\rangle} \leftarrow \llparenthesis N \rrparenthesis\})(\llparenthesis M_1^C \rrparenthesis\{x^{\langle A\rangle} \leftarrow \llparenthesis N \rrparenthesis\}) & =_{\mathrm{IH}} \\ \llparenthesis M_1^{C\supset B}\{x^A \leftarrow N\} \rrparenthesis\llparenthesis M_1^C\{x^A \leftarrow N\} \rrparenthesis & = \\ \llparenthesis (M_1^{C\supset B}M_2^C)^B\{x^A \leftarrow N\} \rrparenthesis. & \end{array}$$

- $M = (!M_1^B)^{[\![s]\!]B}$:

$$
\begin{array}{ll}
\langle\!| M |\!\rangle\{x^{\langle A\rangle} \leftarrow \langle\!| N |\!\rangle\} & = \\
(\lambda y^{\mathbf{1}}.\langle\!| M_1^B |\!\rangle)\{x^{\langle A\rangle} \leftarrow \langle\!| N |\!\rangle\} & = \\
\lambda y^{\mathbf{1}}.(\langle\!| M_1^B |\!\rangle\{x^{\langle A\rangle} \leftarrow \langle\!| N |\!\rangle\}) & =_{\mathrm{IH}} \\
\lambda y^{\mathbf{1}}.\langle\!| M_1^B\{x^A \leftarrow N\} |\!\rangle & = \\
\langle\!| (!(M_1^B\{x^A \leftarrow N\}))^{[\![s]\!]B} |\!\rangle & = \\
\langle\!| (!M_1^B)^{[\![s]\!]B}\{x^A \leftarrow N\} |\!\rangle.
\end{array}
$$

- $M = (M_1^C\langle v^B{:=}r, M_2\rangle)^{C\{v^B \leftarrow r\}}$:

$$
\begin{array}{ll}
\langle\!| M |\!\rangle\{x^{\langle A\rangle} \leftarrow \langle\!| N |\!\rangle\} & = \\
((\lambda y_v^{\mathbf{1}\supset\langle B\rangle}.\langle\!| M_1^C |\!\rangle)\langle\!| M_2^{[\![r]\!]B} |\!\rangle)\{x^{\langle A\rangle} \leftarrow \langle\!| N |\!\rangle\} & = \\
((\lambda y_v^{\mathbf{1}\supset\langle B\rangle}.\langle\!| M_1^C |\!\rangle)\{x^{\langle A\rangle} \leftarrow \langle\!| N |\!\rangle\})(\langle\!| M_2^{[\![r]\!]B} |\!\rangle\{x^{\langle A\rangle} \leftarrow \langle\!| N |\!\rangle\}) & = \\
(\lambda y_v^{\mathbf{1}\supset\langle B\rangle}.(\langle\!| M_1^C |\!\rangle\{x^{\langle A\rangle} \leftarrow \langle\!| N |\!\rangle\}))(\langle\!| M_2^{[\![r]\!]B} |\!\rangle\{x^{\langle A\rangle} \leftarrow \langle\!| N |\!\rangle\}) & =_{\mathrm{IH}} \\
(\lambda y_v^{\mathbf{1}}.\langle\!| M_1^C\{x^A \leftarrow N\} |\!\rangle)(\langle\!| M_2^{[\![r]\!]B}\{x^A \leftarrow N\} |\!\rangle) & = \\
\langle\!| (M_1^C\{x^A \leftarrow N\}\langle v^B{:=}r, M_2\{x^A \leftarrow N\}\rangle)^{C\{v^B \leftarrow r\}} |\!\rangle & = \\
\langle\!| (M_1^C\langle v^B{:=}r, M_2\rangle)^{C\{v^B \leftarrow r\}}\{x^A \leftarrow N\} |\!\rangle.
\end{array}
$$

- $M = ([\alpha^B]M_1^B)^{\perp}$:

$$
\begin{array}{ll}
\langle\!| M |\!\rangle\{x^{\langle A\rangle} \leftarrow \langle\!| N |\!\rangle\} & = \\
([\alpha^{\langle B\rangle}]\langle\!| M_1^B |\!\rangle)\{x^{\langle A\rangle} \leftarrow \langle\!| N |\!\rangle\} & = \\
[\alpha^{\langle B\rangle}](\langle\!| M_1^B |\!\rangle\{x^{\langle A\rangle} \leftarrow \langle\!| N |\!\rangle\}) & =_{\mathrm{IH}} \\
[\alpha^{\langle B\rangle}]\langle\!| M_1^B\{x^A \leftarrow N\} |\!\rangle & = \\
\langle\!| ([\alpha^B](M_1^B\{x^A \leftarrow N\}))^{\perp} |\!\rangle & = \\
\langle\!| ([\alpha^B]M_1^B)^{\perp}\{x^A \leftarrow N\} |\!\rangle.
\end{array}
$$

- $M = (\mu\alpha^B.M_1^{\perp})^B$: this case is analogous to the previous one.
- $M = (M_1^B +_{\mathrm{L}} s)^B$:

$$
\begin{array}{ll}
\langle\!| M |\!\rangle\{x^{\langle A\rangle} \leftarrow \langle\!| N |\!\rangle\} & = \\
\langle\!| M_1^B |\!\rangle\{x^{\langle A\rangle} \leftarrow \langle\!| N |\!\rangle\} =_{\mathrm{IH}} \langle\!| M_1^B\{x^A \leftarrow N\} |\!\rangle & = \\
\langle\!| (M_1^B\{x^A \leftarrow N\} +_{\mathrm{L}} s\{x^A \leftarrow \mathsf{w}(N)\})^B |\!\rangle & = \\
\langle\!| (M_1^B +_{\mathrm{L}} s)^B\{x^A \leftarrow N\} |\!\rangle.
\end{array}
$$

- $M = (s +_{\mathrm{R}} M_1^B)^B$: this case is analogous to the previous one.

□

Lemma 4.5.8 For all λ^{LP}-terms M, N, for every validity variable v^A, proof witness r and truth variable $y^{\mathbf{1}} \notin \mathsf{FVT}(\langle\!| N |\!\rangle)$:

$$\langle\!| M |\!\rangle\{x_v^{\mathbf{1}\supset\langle A\rangle} \leftarrow \lambda y^{\mathbf{1}}.\langle\!| N |\!\rangle\} \twoheadrightarrow_{R_1} \langle\!| M\{v^A \leftarrow N, r\} |\!\rangle$$

where $\twoheadrightarrow_{R_1}$ is the reflexive transitive closure of β-reduction (rule R_1 in $\lambda\mu^{\mathbf{1}}$).

Proof.- By structural induction on M.

- $M = z^B$:

$$\begin{array}{ll}
\llparenthesis M \rrparenthesis\{x_v^{\mathbf{1}\supset\langle A\rangle} \leftarrow \lambda y^{\mathbf{1}}.\llparenthesis N \rrparenthesis\} & = \\
z^{\langle B\rangle}\{x_v^{\mathbf{1}\supset\langle A\rangle} \leftarrow \lambda y^{\mathbf{1}}.\llparenthesis N \rrparenthesis\} & = \\
z^{\langle B\rangle} & = \\
\llparenthesis z^B \rrparenthesis & = \\
\llparenthesis z^B\{v^A \leftarrow N,r\} \rrparenthesis. &
\end{array}$$

- $M = v^A$:

$$\begin{array}{ll}
\llparenthesis M \rrparenthesis\{x_v^{\mathbf{1}\supset\langle A\rangle} \leftarrow \lambda y^{\mathbf{1}}.\llparenthesis N \rrparenthesis\} & = \\
((x_v^{\mathbf{1}\supset\langle A\rangle})\mathtt{unit})\{x_v^{\mathbf{1}\supset\langle A\rangle} \leftarrow \lambda y^{\mathbf{1}}.\llparenthesis N \rrparenthesis\} & = \\
(x_v^{\mathbf{1}\supset\langle A\rangle}\{x_v^{\mathbf{1}\supset\langle A\rangle} \leftarrow \lambda y^{\mathbf{1}}.\llparenthesis N \rrparenthesis\})(\mathtt{unit}\{x_v^{\mathbf{1}\supset\langle A\rangle} \leftarrow \lambda y^{\mathbf{1}}.\llparenthesis N \rrparenthesis\}) & = \\
(\lambda y^{\mathbf{1}}.\llparenthesis N \rrparenthesis)\mathtt{unit} & \rightarrow_\beta \\
\llparenthesis N \rrparenthesis & = \\
\llparenthesis v^A\{v^A \leftarrow N,r\} \rrparenthesis. &
\end{array}$$

- $M = w^B$ with $w^B \neq v^A$:

$$\begin{array}{ll}
\llparenthesis M \rrparenthesis\{x_v^{\mathbf{1}\supset\langle A\rangle} \leftarrow \lambda y^{\mathbf{1}}.\llparenthesis N \rrparenthesis\} & = \\
((x_w^{\mathbf{1}\supset\langle B\rangle})\mathtt{unit})\{x_v^{\mathbf{1}\supset\langle A\rangle} \leftarrow \lambda y^{\mathbf{1}}.\llparenthesis N \rrparenthesis\} & = \\
(x_w^{\mathbf{1}\supset\langle B\rangle})\mathtt{unit} & = \\
\llparenthesis w^B \rrparenthesis & = \\
\llparenthesis w^B\{v^A \leftarrow N,r\} \rrparenthesis. &
\end{array}$$

- $M = (\lambda z^C.M_1^B)^{C\supset B}$:

$$\begin{array}{ll}
\llparenthesis M \rrparenthesis\{x_v^{\mathbf{1}\supset\langle A\rangle} \leftarrow \lambda y^{\mathbf{1}}.\llparenthesis N \rrparenthesis\} & = \\
(\lambda z^{\langle C\rangle}.\llparenthesis M_1^B \rrparenthesis)\{x_v^{\mathbf{1}\supset\langle A\rangle} \leftarrow \lambda y^{\mathbf{1}}.\llparenthesis N \rrparenthesis\} & = \\
\lambda z^{\langle C\rangle}.(\llparenthesis M_1^B \rrparenthesis\{x_v^{\mathbf{1}\supset\langle A\rangle} \leftarrow \lambda y^{\mathbf{1}}.\llparenthesis N \rrparenthesis\}) & \overset{\text{IH}}{\twoheadrightarrow}_{R_1} \\
\lambda z^{\langle C\rangle}.\llparenthesis M_1^B\{v^A \leftarrow N,r\} \rrparenthesis & = \\
\llparenthesis (\lambda z^C.(M_1^B\{v^A \leftarrow N,r\}))^{C\supset B} \rrparenthesis & = \\
\llparenthesis (\lambda z^C.M_1^B)^{C\supset B}\{v^A \leftarrow N,r\} \rrparenthesis. &
\end{array}$$

- $M = (M_1^{C\supset B} M_2^C)^B$:

$$\begin{array}{ll}
\llparenthesis M \rrparenthesis\{x_v^{\mathbf{1}\supset\langle A\rangle} \leftarrow \lambda y^{\mathbf{1}}.\llparenthesis N \rrparenthesis\} & = \\
(\llparenthesis M_1^{C\supset B} \rrparenthesis \llparenthesis M_1^C \rrparenthesis)\{x_v^{\mathbf{1}\supset\langle A\rangle} \leftarrow \lambda y^{\mathbf{1}}.\llparenthesis N \rrparenthesis\} & = \\
(\llparenthesis M_1^{C\supset B} \rrparenthesis\{x_v^{\mathbf{1}\supset\langle A\rangle} \leftarrow \lambda y^{\mathbf{1}}.\llparenthesis N \rrparenthesis\})(\llparenthesis M_1^C \rrparenthesis\{x_v^{\mathbf{1}\supset\langle A\rangle} \leftarrow \lambda y^{\mathbf{1}}.\llparenthesis N \rrparenthesis\}) & \overset{\text{IH}}{\twoheadrightarrow}_{R_1} \\
\llparenthesis M_1^{C\supset B}\{v^A \leftarrow N,r\} \rrparenthesis \llparenthesis M_1^C\{v^A \leftarrow N,r\} \rrparenthesis & = \\
\llparenthesis (M_1^{C\supset B} M_2^C)^B\{v^A \leftarrow N,r\} \rrparenthesis. &
\end{array}$$

- $M = (!M_1^B)^{[\![s]\!]B}$:

$$
\begin{array}{ll}
\llparenthesis M \rrparenthesis\{x_v^{\mathbf{1}\supset\langle A\rangle} \leftarrow \lambda y^{\mathbf{1}}.\llparenthesis N \rrparenthesis\} & = \\
(\lambda x^{\mathbf{1}}.\llparenthesis M_1^B \rrparenthesis)\{x_v^{\mathbf{1}\supset\langle A\rangle} \leftarrow \lambda y^{\mathbf{1}}.\llparenthesis N \rrparenthesis\} & = \\
\lambda x^{\mathbf{1}}.(\llparenthesis M_1^B \rrparenthesis\{x_v^{\mathbf{1}\supset\langle A\rangle} \leftarrow \lambda y^{\mathbf{1}}.\llparenthesis N \rrparenthesis\}) & \xrightarrow{\text{IH}}_{R_1} \\
\lambda x^{\mathbf{1}}.\llparenthesis M_1^B\{v^A \leftarrow N,r\} \rrparenthesis & = \\
\llparenthesis (!(M_1^B\{v^A \leftarrow N,r\}))^{(\llbracket s \rrbracket B)\{v^A \leftarrow N,r\}} \rrparenthesis & = \\
\llparenthesis (!M_1^B)^{\llbracket s \rrbracket B}\{v^A \leftarrow N,r\} \rrparenthesis. &
\end{array}
$$

- $M = (M_1^C\langle v^A{:=}t, M_2\rangle)^{C\{v^A \leftarrow t\}}$: keep in mind that, by variable convention, $v^A \notin \mathsf{FVV}(M_2)$, and therefore, by Lemma 4.5.6, $x_v^A \notin \mathsf{FVT}(\llparenthesis M_2 \rrparenthesis)$.

$$
\begin{array}{ll}
\llparenthesis M \rrparenthesis\{x_v^{\mathbf{1}\supset\langle A\rangle} \leftarrow \lambda y^{\mathbf{1}}.\llparenthesis N \rrparenthesis\} & = \\
((\lambda x_v^{\mathbf{1}\supset\langle A\rangle}.\llparenthesis M_1^C \rrparenthesis)\llparenthesis M_2^{\llbracket t \rrbracket A} \rrparenthesis)\{x_v^{\mathbf{1}\supset\langle A\rangle} \leftarrow \lambda y^{\mathbf{1}}.\llparenthesis N \rrparenthesis\} & = \\
(\lambda x_v^{\mathbf{1}\supset\langle A\rangle}.\llparenthesis M_1^C \rrparenthesis)\llparenthesis M_2^{\llbracket t \rrbracket A} \rrparenthesis & = \\
\llparenthesis M \rrparenthesis & = \\
\llparenthesis M\{v^A \leftarrow N,r\} \rrparenthesis. &
\end{array}
$$

- $M = (M_1^C\langle w^B{:=}t, M_2\rangle)^{C\{w^B \leftarrow t\}}$ with $w^B \neq v^A$:

$$
\begin{array}{ll}
\llparenthesis M \rrparenthesis\{x_v^{\mathbf{1}\supset\langle A\rangle} \leftarrow \lambda y^{\mathbf{1}}.\llparenthesis N \rrparenthesis\} & = \\
((\lambda x_v^{\mathbf{1}\supset\langle B\rangle}.\llparenthesis M_1^C \rrparenthesis)\llparenthesis M_2^{\llbracket t \rrbracket B} \rrparenthesis)\{x_v^{\mathbf{1}\supset\langle A\rangle} \leftarrow \lambda y^{\mathbf{1}}.\llparenthesis N \rrparenthesis\} & = \\
((\lambda x_v^{\mathbf{1}\supset\langle B\rangle}.\llparenthesis M_1^C \rrparenthesis)\{x_v^{\mathbf{1}\supset\langle A\rangle} \leftarrow \lambda y^{\mathbf{1}}.\llparenthesis N \rrparenthesis\})(\llparenthesis M_2^{\llbracket t \rrbracket B} \rrparenthesis\{x_v^{\mathbf{1}\supset\langle A\rangle} \leftarrow \lambda y^{\mathbf{1}}.\llparenthesis N \rrparenthesis\}) & = \\
(\lambda x_w^{\mathbf{1}\supset\langle B\rangle}.(\llparenthesis M_1^C \rrparenthesis\{x_v^{\mathbf{1}\supset\langle A\rangle} \leftarrow \lambda y^{\mathbf{1}}.\llparenthesis N \rrparenthesis\}))(\llparenthesis M_2^{\llbracket t \rrbracket B} \rrparenthesis\{x_v^{\mathbf{1}\supset\langle A\rangle} \leftarrow \lambda y^{\mathbf{1}}.\llparenthesis N \rrparenthesis\}) & \xrightarrow{\text{IH}}_{R_1} \\
(\lambda x_w^{\mathbf{1}}.\llparenthesis M_1^C\{v^A \leftarrow N,r\} \rrparenthesis)(\llparenthesis M_2^{\llbracket t \rrbracket B}\{v^A \leftarrow N,r\} \rrparenthesis) & = \\
\llparenthesis (M_1^C\{v^A \leftarrow N,r\}\langle w^B{:=}t, M_2\{v^A \leftarrow N,r\}\rangle)^{C\{w^B \leftarrow t\}\{v^A \leftarrow r\}} \rrparenthesis & = \\
\llparenthesis (M_1^C\langle w^B{:=}t, M_2\rangle)^{C\{w^B \leftarrow t\}}\{v^A \leftarrow N,r\} \rrparenthesis. &
\end{array}
$$

- $M = ([\alpha^B]M_1^B)^{\perp}$:

$$
\begin{array}{ll}
\llparenthesis M \rrparenthesis\{x_v^{\mathbf{1}\supset\langle A\rangle} \leftarrow \lambda y^{\mathbf{1}}.\llparenthesis N \rrparenthesis\} & = \\
([\alpha^{\langle B\rangle}]\llparenthesis M_1^B \rrparenthesis)\{x_v^{\mathbf{1}\supset\langle A\rangle} \leftarrow \lambda y^{\mathbf{1}}.\llparenthesis N \rrparenthesis\} & = \\
[\alpha^{\langle B\rangle}](\llparenthesis M_1^B \rrparenthesis\{x_v^{\mathbf{1}\supset\langle A\rangle} \leftarrow \lambda y^{\mathbf{1}}.\llparenthesis N \rrparenthesis\}) & \xrightarrow{\text{IH}}_{R_1} \\
[\alpha^{\langle B\rangle}]\llparenthesis M_1^B\{v^A \leftarrow N,r\} \rrparenthesis & = \\
\llparenthesis ([\alpha^B](M_1^B\{v^A \leftarrow N,r\}))^{\perp} \rrparenthesis & = \\
\llparenthesis ([\alpha^B]M_1^B)^{\perp}\{v^A \leftarrow N,r\} \rrparenthesis. &
\end{array}
$$

- $M = (\mu\alpha^B.M_1^{\perp})^B$: this case is analogous to the previous one.
- $M = (M_1^B +_{\mathrm{L}} s)^B$:

$$
\begin{array}{ll}
\llparenthesis M \rrparenthesis\{x_v^{\mathbf{1}\supset\langle A\rangle} \leftarrow \lambda y^{\mathbf{1}}.\llparenthesis N \rrparenthesis\} & = \\
\llparenthesis M_1^B \rrparenthesis\{x_v^{\mathbf{1}\supset\langle A\rangle} \leftarrow \lambda y^{\mathbf{1}}.\llparenthesis N \rrparenthesis\} & \xrightarrow{\text{IH}}_{R_1} \\
\llparenthesis M_1^B\{v^A \leftarrow N,r\} \rrparenthesis & = \\
\llparenthesis (M_1^B\{v^A \leftarrow N,r\} +_{\mathrm{L}} s\{v^A \leftarrow r\})^B \rrparenthesis & = \\
\llparenthesis (M_1^B +_{\mathrm{L}} s)^B\{v^A \leftarrow N,r\} \rrparenthesis. &
\end{array}
$$

- $M = (s +_{\mathrm{R}} M_1^B)^B$: this case is analogous to the previous one.

$\square$

Lemma 4.5.9 For every λ^{LP}-term M, for all falsehood variables α^A, β^A:
$\langle\!| M |\!\rangle\{\alpha^{\langle\!| A |\!\rangle} \leftarrow \beta^{\langle\!| A |\!\rangle}\} = \langle\!| M\{\alpha^A \leftarrow \beta^A\} |\!\rangle$.

Proof.- Clear by structural induction on M. $\square$

Lemma 4.5.10 For all λ^{LP}-terms M, N^A, for all falsehood variables $\alpha^{A\supset B}$, β^B:
$\langle\!| M |\!\rangle(\!| [\alpha^{\langle\!| A\supset B |\!\rangle}](\bullet) \leftarrow [\beta^{\langle\!| B |\!\rangle}](\bullet)\langle\!| N^A |\!\rangle |\!) = \langle\!| M(\!| [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)N^A |\!) |\!\rangle$.

Proof.- By structural induction on M.

- $M = x^C$:

$$\begin{array}{ll}
\langle\!| M |\!\rangle(\!| [\alpha^{\langle\!| A\supset B |\!\rangle}](\bullet) \leftarrow [\beta^{\langle\!| B |\!\rangle}](\bullet)\langle\!| N^A |\!\rangle |\!) & = \\
x^{\langle\!| C |\!\rangle}(\!| [\alpha^{\langle\!| A\supset B |\!\rangle}](\bullet) \leftarrow [\beta^{\langle\!| B |\!\rangle}](\bullet)\langle\!| N^A |\!\rangle |\!) & = \\
x^{\langle\!| C |\!\rangle} & = \\
\langle\!| x^C |\!\rangle & = \\
\langle\!| x^C(\!| [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)N^A |\!) |\!\rangle. &
\end{array}$$

- $M = v^C$:

$$\begin{array}{ll}
\langle\!| M |\!\rangle(\!| [\alpha^{\langle\!| A\supset B |\!\rangle}](\bullet) \leftarrow [\beta^{\langle\!| B |\!\rangle}](\bullet)\langle\!| N^A |\!\rangle |\!) & = \\
((x_v^{\mathbf{1}\supset\langle\!| C |\!\rangle})\mathtt{unit})(\!| [\alpha^{\langle\!| A\supset B |\!\rangle}](\bullet) \leftarrow [\beta^{\langle\!| B |\!\rangle}](\bullet)\langle\!| N^A |\!\rangle |\!) & = \\
(x_v^{\mathbf{1}\supset\langle\!| C |\!\rangle})\mathtt{unit} & = \\
\langle\!| v^C |\!\rangle & = \\
\langle\!| v^C(\!| [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)N^A |\!) |\!\rangle. &
\end{array}$$

- $M = (\lambda x^C.M_1^D)^{C\supset D}$:

$$\begin{array}{ll}
\langle\!| M |\!\rangle(\!| [\alpha^{\langle\!| A\supset B |\!\rangle}](\bullet) \leftarrow [\beta^{\langle\!| B |\!\rangle}](\bullet)\langle\!| N^A |\!\rangle |\!) & = \\
(\lambda x^{\langle\!| C |\!\rangle}.\langle\!| M_1^D |\!\rangle)(\!| [\alpha^{\langle\!| A\supset B |\!\rangle}](\bullet) \leftarrow [\beta^{\langle\!| B |\!\rangle}](\bullet)\langle\!| N^A |\!\rangle |\!) & = \\
\lambda x^{\langle\!| C |\!\rangle}.(\langle\!| M_1^D |\!\rangle(\!| [\alpha^{\langle\!| A\supset B |\!\rangle}](\bullet) \leftarrow [\beta^{\langle\!| B |\!\rangle}](\bullet)\langle\!| N^A |\!\rangle |\!)) & =_{\mathrm{IH}} \\
\lambda x^{\langle\!| C |\!\rangle}.\langle\!| M_1^D(\!| [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)N^A |\!) |\!\rangle & = \\
\langle\!| (\lambda x^C.M_1^D(\!| [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)N^A |\!))^{C\supset D} |\!\rangle & = \\
\langle\!| (\lambda x^C.M_1^D)^{C\supset D}(\!| [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)N^A |\!) |\!\rangle. &
\end{array}$$

- $M = (M_1^{C\supset D} M_2^C)^D$:

$$\begin{array}{ll}
\langle\!| M |\!\rangle(\!| [\alpha^{\langle\!| A\supset B |\!\rangle}](\bullet) \leftarrow [\beta^{\langle\!| B |\!\rangle}](\bullet)\langle\!| N^A |\!\rangle |\!) & = \\
(\langle\!| M_1^{C\supset D} |\!\rangle\langle\!| M_1^C |\!\rangle)(\!| [\alpha^{\langle\!| A\supset B |\!\rangle}](\bullet) \leftarrow [\beta^{\langle\!| B |\!\rangle}](\bullet)\langle\!| N^A |\!\rangle |\!) & = \\
(\langle\!| M_1^{C\supset D} |\!\rangle(\!| [\alpha^{\langle\!| A\supset B |\!\rangle}](\bullet) \leftarrow [\beta^{\langle\!| B |\!\rangle}](\bullet)\langle\!| N^A |\!\rangle |\!))(\langle\!| M_1^C |\!\rangle(\!| [\alpha^{\langle\!| A\supset B |\!\rangle}](\bullet) \leftarrow [\beta^{\langle\!| B |\!\rangle}](\bullet)\langle\!| N^A |\!\rangle |\!)) & =_{\mathrm{IH}} \\
\langle\!| M_1^{C\supset D}(\!| [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)N^A |\!) |\!\rangle\langle\!| M_1^C(\!| [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)N^A |\!) |\!\rangle & = \\
\langle\!| (M_1^{C\supset D} M_2^C)^B(\!| [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)N^A |\!) |\!\rangle. &
\end{array}$$

- $M = (!M_1^C)^{[\![s]\!]C}$:

$$
\begin{array}{ll}
\langle\!\langle M \rangle\!\rangle (\!| [\alpha^{\langle A \supset B \rangle}](\bullet) \leftarrow [\beta^{\langle B \rangle}](\bullet) \langle\!\langle N^A \rangle\!\rangle |\!) & = \\
(\lambda y^{\mathbf{1}}.\langle\!\langle M_1^C \rangle\!\rangle)(\!| [\alpha^{\langle A \supset B \rangle}](\bullet) \leftarrow [\beta^{\langle B \rangle}](\bullet) \langle\!\langle N^A \rangle\!\rangle |\!) & = \\
\lambda y^{\mathbf{1}}.(\langle\!\langle M_1^C \rangle\!\rangle (\!| [\alpha^{\langle A \supset B \rangle}](\bullet) \leftarrow [\beta^{\langle B \rangle}](\bullet) \langle\!\langle N^A \rangle\!\rangle |\!)) & =_{\mathrm{IH}} \\
\lambda y^{\mathbf{1}}.\langle\!\langle M_1^C (\!| [\alpha^{A \supset B}](\bullet) \leftarrow [\beta^B](\bullet) N^A |\!) \rangle\!\rangle & = \\
\langle\!\langle (!(M_1^C (\!| [\alpha^{A \supset B}](\bullet) \leftarrow [\beta^B](\bullet) N^A |\!)))^{[\![s]\!]C} \rangle\!\rangle & = \\
\langle\!\langle (!M_1^C)^{[\![s]\!]C} (\!| [\alpha^{A \supset B}](\bullet) \leftarrow [\beta^B](\bullet) N^A |\!) \rangle\!\rangle. &
\end{array}
$$

- $M = (M_1^C \langle v^D{:=}r, M_2 \rangle)^{C\{v^D \leftarrow r\}}$:

$$
\begin{array}{ll}
\langle\!\langle M \rangle\!\rangle (\!| [\alpha^{\langle A \supset B \rangle}](\bullet) \leftarrow [\beta^{\langle B \rangle}](\bullet) \langle\!\langle N^A \rangle\!\rangle |\!) & = \\
((\lambda x_v^{\mathbf{1} \supset \langle D \rangle}.\langle\!\langle M_1^C \rangle\!\rangle)\langle\!\langle M_2 \rangle\!\rangle)(\!| [\alpha^{\langle A \supset B \rangle}](\bullet) \leftarrow [\beta^{\langle B \rangle}](\bullet) \langle\!\langle N^A \rangle\!\rangle |\!) & = \\
((\lambda x_v^{\mathbf{1} \supset \langle D \rangle}.\langle\!\langle M_1^C \rangle\!\rangle)(\!| [\alpha^{\langle A \supset B \rangle}](\bullet) \leftarrow [\beta^{\langle B \rangle}](\bullet) \langle\!\langle N^A \rangle\!\rangle |\!))(\langle\!\langle M_2 \rangle\!\rangle (\!| [\alpha^{\langle A \supset B \rangle}](\bullet) \leftarrow [\beta^{\langle B \rangle}](\bullet) \langle\!\langle N^A \rangle\!\rangle |\!)) & = \\
(\lambda x_v^{\mathbf{1} \supset \langle D \rangle}.\langle\!\langle M_1^C \rangle\!\rangle (\!| [\alpha^{\langle A \supset B \rangle}](\bullet) \leftarrow [\beta^{\langle B \rangle}](\bullet) \langle\!\langle N^A \rangle\!\rangle |\!))(\langle\!\langle M_2 \rangle\!\rangle (\!| [\alpha^{\langle A \supset B \rangle}](\bullet) \leftarrow [\beta^{\langle B \rangle}](\bullet) \langle\!\langle N^A \rangle\!\rangle |\!)) & =_{\mathrm{IH}} \\
(\lambda x_v^{\mathbf{1}}.\langle\!\langle M_1^C (\!| [\alpha^{A \supset B}](\bullet) \leftarrow [\beta^B](\bullet) N^A |\!) \rangle\!\rangle)(\langle\!\langle M_2 (\!| [\alpha^{A \supset B}](\bullet) \leftarrow [\beta^B](\bullet) N^A |\!) \rangle\!\rangle) & = \\
\langle\!\langle (M_1^C (\!| [\alpha^{A \supset B}](\bullet) \leftarrow [\beta^B](\bullet) N^A |\!))^{C\{v^D \leftarrow r\}} \langle v^D{:=}r, M_2 (\!| [\alpha^{A \supset B}](\bullet) \leftarrow [\beta^B](\bullet) N^A |\!) \rangle \rangle\!\rangle & = \\
\langle\!\langle (M_1^C \langle v^D{:=}r, M_2 \rangle)^{C\{v^D \leftarrow r\}} (\!| [\alpha^{A \supset B}](\bullet) \leftarrow [\beta^B](\bullet) N^A |\!) \rangle\!\rangle. &
\end{array}
$$

- $M = ([\alpha^{A \supset B}] M_1^{A \supset B})^{\perp}$:

$$
\begin{array}{ll}
\langle\!\langle M \rangle\!\rangle (\!| [\alpha^{\langle A \supset B \rangle}](\bullet) \leftarrow [\beta^{\langle B \rangle}](\bullet) \langle\!\langle N^A \rangle\!\rangle |\!) & = \\
[\alpha^{\langle A \supset B \rangle}] \langle\!\langle M_1^{A \supset B} \rangle\!\rangle (\!| [\alpha^{\langle A \supset B \rangle}](\bullet) \leftarrow [\beta^{\langle B \rangle}](\bullet) \langle\!\langle N^A \rangle\!\rangle |\!) & = \\
[\beta^{\langle B \rangle}] \langle\!\langle M_1^{A \supset B} \rangle\!\rangle \langle\!\langle N^A \rangle\!\rangle & = \\
\langle\!\langle ([\beta^B](M_1^{A \supset B} N^A)^B)^{\perp} \rangle\!\rangle & = \\
\langle\!\langle ([\alpha^{A \supset B}] M_1^{A \supset B})^{\perp} (\!| [\alpha^{A \supset B}](\bullet) \leftarrow [\beta^B](\bullet) N^A |\!) \rangle\!\rangle. &
\end{array}
$$

- $M = ([\gamma^C] M_1^C)^{\perp}$ with $\gamma^C \neq \alpha^{A \supset B}$:

$$
\begin{array}{ll}
\langle\!\langle M \rangle\!\rangle (\!| [\alpha^{\langle A \supset B \rangle}](\bullet) \leftarrow [\beta^{\langle B \rangle}](\bullet) \langle\!\langle N^A \rangle\!\rangle |\!) & = \\
([\gamma^{\langle C \rangle}] \langle\!\langle M_1^C \rangle\!\rangle)(\!| [\alpha^{\langle A \supset B \rangle}](\bullet) \leftarrow [\beta^{\langle B \rangle}](\bullet) \langle\!\langle N^A \rangle\!\rangle |\!) & = \\
[\gamma^{\langle C \rangle}](\langle\!\langle M_1^C \rangle\!\rangle (\!| [\alpha^{\langle A \supset B \rangle}](\bullet) \leftarrow [\beta^{\langle B \rangle}](\bullet) \langle\!\langle N^A \rangle\!\rangle |\!)) & =_{\mathrm{IH}} \\
[\gamma^{\langle C \rangle}] \langle\!\langle M_1^C (\!| [\alpha^{A \supset B}](\bullet) \leftarrow [\beta^B](\bullet) N^A |\!) \rangle\!\rangle & = \\
\langle\!\langle ([\gamma^C](M_1^C (\!| [\alpha^{A \supset B}](\bullet) \leftarrow [\beta^B](\bullet) N^A |\!)))^{\perp} \rangle\!\rangle & = \\
\langle\!\langle ([\gamma^C] M_1^C)^{\perp} (\!| [\alpha^{A \supset B}](\bullet) \leftarrow [\beta^B](\bullet) N^A |\!) \rangle\!\rangle. &
\end{array}
$$

- $M = (\mu \alpha^{A \supset B}.M_1^{\perp})^{A \supset B}$:

$$
\begin{array}{ll}
\langle\!\langle M \rangle\!\rangle (\!| [\alpha^{\langle A \supset B \rangle}](\bullet) \leftarrow [\beta^{\langle B \rangle}](\bullet) \langle\!\langle N^A \rangle\!\rangle |\!) & = \\
\langle\!\langle M \rangle\!\rangle & = \\
\langle\!\langle M (\!| [\alpha^{A \supset B}](\bullet) \leftarrow [\beta^B](\bullet) N^A |\!) \rangle\!\rangle. &
\end{array}
$$

- $M = (\mu \gamma^C.M_1^{\perp})^C$ with $\gamma^C \neq \alpha^{A \supset B}$:

$$
\begin{array}{ll}
\langle\!\langle M \rangle\!\rangle (\!| [\alpha^{\langle A \supset B \rangle}](\bullet) \leftarrow [\beta^{\langle B \rangle}](\bullet) \langle\!\langle N^A \rangle\!\rangle |\!) & = \\
(\mu \gamma^{\langle C \rangle}.\langle\!\langle M_1^{\perp} \rangle\!\rangle)(\!| [\alpha^{\langle A \supset B \rangle}](\bullet) \leftarrow [\beta^{\langle B \rangle}](\bullet) \langle\!\langle N^A \rangle\!\rangle |\!) & = \\
\mu \gamma^{\langle C \rangle}.\langle\!\langle M_1^{\perp} \rangle\!\rangle (\!| [\alpha^{\langle A \supset B \rangle}](\bullet) \leftarrow [\beta^{\langle B \rangle}](\bullet) \langle\!\langle N^A \rangle\!\rangle |\!) & =_{\mathrm{IH}} \\
\mu \gamma^{\langle C \rangle}.\langle\!\langle M_1^{\perp} (\!| [\alpha^{A \supset B}](\bullet) \leftarrow [\beta^B](\bullet) N^A |\!) \rangle\!\rangle & = \\
\langle\!\langle (\mu \gamma^C.M_1^{\perp} (\!| [\alpha^{A \supset B}](\bullet) \leftarrow [\beta^B](\bullet) N^A |\!))^C \rangle\!\rangle & = \\
\langle\!\langle (\mu \gamma^C.M_1^{\perp})^C (\!| [\alpha^{A \supset B}](\bullet) \leftarrow [\beta^B](\bullet) N^A |\!) \rangle\!\rangle. &
\end{array}
$$

- $M = (M_1^C +_{\mathrm{L}} s)^C$:

$$\begin{array}{ll}
\langle\!| M |\!\rangle (\!| [\alpha^{\langle A \supset B \rangle}](\bullet) \leftarrow [\beta^{\langle B \rangle}](\bullet) \langle\!| N^A |\!\rangle |\!) & = \\
\langle\!| M_1^C |\!\rangle (\!| [\alpha^{\langle A \supset B \rangle}](\bullet) \leftarrow [\beta^{\langle B \rangle}](\bullet) \langle\!| N^A |\!\rangle |\!) & =_{\mathrm{IH}} \\
\langle\!| M_1^C (\!| [\alpha^{A \supset B}](\bullet) \leftarrow [\beta^B](\bullet) N^A |\!) |\!\rangle & = \\
\langle\!| (M_1^C (\!| [\alpha^{A \supset B}](\bullet) \leftarrow [\beta^B](\bullet) N^A |\!) +_{\mathrm{L}} s (\!| [\alpha^{A \supset B}](\bullet) \leftarrow [\beta^B](\bullet) \mathsf{w}(N^A) |\!))^C |\!\rangle & = \\
\langle\!| (M_1^C +_{\mathrm{L}} s)^C (\!| [\alpha^{A \supset B}](\bullet) \leftarrow [\beta^B](\bullet) N^A |\!) |\!\rangle. &
\end{array}$$

- $M = (s +_{\mathrm{R}} M_1^C)^C$: this case is analogous to the previous one.

□

In order to prove that the mapping preserves strong normalization, we need to distinguish between the two kinds of λ^{LP}-reduction: principal reduction (using the rules β, γ, μ, ζ and θ) and permutative reduction (rules ψ_L, ψ_R, ϕ_L, ϕ_R, χ_L, χ_R, ι_L and ι_L). We will show that a principal reduction step maps to one or more reduction steps in $\lambda\mu^{\mathbf{1}}$, while permutative reduction steps are dropped by the mapping (i.e. they translate to 0 reduction steps in $\lambda\mu^{\mathbf{1}}$).

Lemma 4.5.11 For all λ^{LP}-terms M and N, if $M \to N$ in λ^{LP} *without* the use of permutative rules, then $\langle\!| M |\!\rangle \to^+ \langle\!| N |\!\rangle$ in $\lambda\mu^{\mathbf{1}}$. That is, $\langle\!| M |\!\rangle$ reduces to $\langle\!| N |\!\rangle$ in 1 or more steps.

Proof.- By induction on the derivation of M. The base cases are trivial, since no reductions may originate from x^A or v^A. The inductive cases follow from the fact that reduction is closed under all constructors in $\lambda\mu^{\mathbf{1}}$. If the reduction takes place at the root of M, we must consider which reduction rule was used.

- $\beta : M = (\lambda x^A.M_1^B)M_2^A$ and $N = M_1^B\{x^A \leftarrow M_2^A\}$.

$$\begin{array}{ll}
\langle\!| M |\!\rangle & = \\
(\lambda x^{\langle A \rangle}.\langle\!| M_1^B |\!\rangle)\langle\!| M_2^A |\!\rangle & \to_{R_1} \\
\langle\!| M_1^B |\!\rangle\{x^{\langle A \rangle} \leftarrow \langle\!| M_2^A |\!\rangle\} & =_{\text{L. 4.5.7}} \\
\langle\!| M_1^B\{x^A \leftarrow M_2^A\} |\!\rangle & = \\
\langle\!| N |\!\rangle. &
\end{array}$$

- $\gamma : M = (M_1^B\langle v^A{:=}r, !M_2^A\rangle)^{B\{v^A \leftarrow r\}}$ and $N = (M_1^B\{v^A \leftarrow M_2^A, r\})^{B\{v^A \leftarrow r\}}$.

$$\begin{array}{ll}
\langle\!| M |\!\rangle & = \\
(\lambda x_v^{\mathbf{1} \supset \langle A \rangle}.\langle\!| M_1^B |\!\rangle)\lambda y^{\mathbf{1}}.\langle\!| M_2^A |\!\rangle & \to_{R_1} \\
\langle\!| M_1^B |\!\rangle\{x_v^{\mathbf{1} \supset \langle A \rangle} \leftarrow \lambda y^{\mathbf{1}}.\langle\!| M_2^A |\!\rangle\} & \overset{\text{L. 4.5.8}}{\twoheadrightarrow}_{R_1} \\
\langle\!| M_1^B\{v^A \leftarrow M_2^A, r\} |\!\rangle & = \\
\langle\!| N |\!\rangle. &
\end{array}$$

- $\mu : M = [\beta^A]\mu\alpha^A.M_1^{\perp}$ and $N = M_1^{\perp}\{\alpha^A \leftarrow \beta^A\}$.

$$\begin{array}{ll}
\langle\!| M |\!\rangle & = \\
[\beta^{\langle A \rangle}]\mu\alpha^{\langle A \rangle}.\langle\!| M_1^{\perp} |\!\rangle & \to_{S_1} \\
\langle\!| M_1^{\perp} |\!\rangle\{\alpha^{\langle A \rangle} \leftarrow \beta^{\langle A \rangle}\} & =_{\text{L. 4.5.9}} \\
\langle\!| M_1^{\perp}\{\alpha^A \leftarrow \beta^A\} |\!\rangle & = \\
\langle\!| N |\!\rangle. &
\end{array}$$

- $\zeta : M = (\mu\alpha^{A\supset B}.M_1^{\perp})^{A\supset B}M_2^A$ and $N = \mu\beta^B.M_1^{\perp}(\!|\, [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)M_2^A \,|\!)$.

$$\begin{array}{ll}
\langle\!| M |\!\rangle & = \\
(\mu\alpha^{\langle A\supset B\rangle}.\langle\!| M_1^{\perp} |\!\rangle)\langle\!| M_2^A |\!\rangle \to_{R_2} \mu\beta^{\langle B\rangle}.\langle\!| M_1^{\perp} |\!\rangle(\!|\, [\alpha^{\langle A\supset B\rangle}](\bullet) \leftarrow [\beta^{\langle B\rangle}](\bullet)\langle\!| M_2^A |\!\rangle \,|\!) & =_{\text{L. 4.5.10}} \\
\mu\beta^{\langle B\rangle}.\langle\!| M_1^{\perp}(\!|\, [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)M_2^A \,|\!) |\!\rangle & = \\
\langle\!| N |\!\rangle. &
\end{array}$$

- $\theta : M = \mu\alpha^A.[\alpha^A]M_1^A$, $N = M_1^A$ and $\alpha^A \notin \mathsf{FVF}(M^A)$. Note that, by L. 4.5.6, $\alpha^{\langle A\rangle} \notin \mathsf{FVF}(\langle\!| M^A |\!\rangle)$.

$$\begin{array}{ll}
\langle\!| M |\!\rangle & = \\
\mu\alpha^{\langle A\rangle}.[\alpha^{\langle A\rangle}]\langle\!| M_1^A |\!\rangle & \to_{S_2} \\
\langle\!| M_1^A |\!\rangle & = \\
\langle\!| N |\!\rangle. &
\end{array}$$

□

Lemma 4.5.12 For all λ^{LP}-terms M and N, if $M \to N$ in λ^{LP} using *only* permutative rules, then $\langle\!| M |\!\rangle = \langle\!| N |\!\rangle$.

Proof.- By induction on M. As in the previous lemma, no reductions originate from variables. The inductive cases are immediate; we focus on the permutative reductions at the root of M.

- $\psi_L : M = (M_1^{A\supset B} +_{\mathrm{L}} s)^{A\supset B}M_2^A$ and $N = (M_1^{A\supset B}M_2^A)^B +_{\mathrm{L}} s$. Then $\langle\!| M |\!\rangle = \langle\!| M_1^{A\supset B} |\!\rangle\langle\!| M_2^A |\!\rangle = \langle\!| N |\!\rangle$.
- ψ_R : this case is analogous to the previous one.
- $\phi_L : M = M_2^B\langle v^A{:=}r, (M_1^{[\![r]\!]A} +_{\mathrm{L}} s)\rangle$ and $N = (M_2^B\langle v^A{:=}r, M_1\rangle)^{B\{v^A \leftarrow r^A\}} +_{\mathrm{L}} s$. Then $\langle\!| M |\!\rangle = (\lambda x_v^{\mathbf{1}\supset\langle A\rangle}.\langle\!| M_2^B |\!\rangle)\langle\!| M_1^{[\![r]\!]A} |\!\rangle = \langle\!| N |\!\rangle$.
- ϕ_R : this case is analogous to the previous one.
- $\chi_L : M = [\beta^A](M_1^A +_{\mathrm{L}} s)^A$ and $N = ([\beta^A]M_1^A)^{\perp} +_{\mathrm{L}} s$. $\langle\!| M |\!\rangle = [\beta^{\langle A\rangle}]\langle\!| M_1^A |\!\rangle = \langle\!| N |\!\rangle$.
- χ_R : this case is analogous to the previous one.
- $\iota_L : M = \mu\alpha^A.(M_1^{\perp} +_{\mathrm{L}} s)^{\perp}$ and $N = (\mu\alpha^A.M_1^{\perp})^A +_{\mathrm{L}} s$. $\langle\!| M |\!\rangle = \mu\alpha^{\langle A\rangle}.\langle\!| M_1^{\perp} |\!\rangle = \langle\!| N |\!\rangle$.
- ι_R : this case is analogous to the previous one.

□

Lemma 4.5.13 Permutative reduction is SN.

Proof.- We exhibit a polynomial interpretation $(\cdot)_{\mathcal{A}}$ in $\mathbb{N}_{\geq 2}$, using the standard order for natural numbers (cf. Chapter 6 of [BKdV03]). Terms are interpreted as follows:

$$
\begin{array}{rcl}
x^B_{\mathcal{A}} & \triangleq & 2 \\
v^B_{\mathcal{A}} & \triangleq & 2 \\
(M^{A\supset B}N^A)^B_{\mathcal{A}} & \triangleq & M^{A\supset B}_{\mathcal{A}} \times N^A_{\mathcal{A}} \\
(\lambda x^A.M^B)^{A\supset B}_{\mathcal{A}} & \triangleq & 2 \times M^B_{\mathcal{A}} \\
([\alpha^B]M^B)^{\perp}_{\mathcal{A}} & \triangleq & 2 \times M^B_{\mathcal{A}} \\
(\mu\alpha^B.M^{\perp})^B_{\mathcal{A}} & \triangleq & 2 \times M^{\perp}_{\mathcal{A}} \\
(!M^B)^{[\![s]\!]B}_{\mathcal{A}} & \triangleq & 1 + M^B_{\mathcal{A}} \\
(N^B\langle v^C{:=}r, M\rangle)^{B\{v^C\leftarrow r\}}_{\mathcal{A}} & \triangleq & N^B_{\mathcal{A}} \times M^{[\![r]\!]B}_{\mathcal{A}} + 1 \\
(M^B +_{\mathrm{L}} t)^B_{\mathcal{A}} & \triangleq & 2 \times M^B_{\mathcal{A}} + 2 \\
(s +_{\mathrm{R}} M^B)^B_{\mathcal{A}} & \triangleq & 2 \times M^B_{\mathcal{A}} + 2
\end{array}
$$

We now show that the permutative rules are compatible with this interpretation:

$$
\begin{array}{lrcl}
\psi_L : & (M^{A\supset B} +_{\mathrm{L}} t)^{A\supset B}N^A & \to & (M^{A\supset B}N^A)^B +_{\mathrm{L}} t \\
 & (2 \times M^{A\supset B}_{\mathcal{A}} + 2) \times N^A_{\mathcal{A}} & > & 2 \times M^{A\supset B}_{\mathcal{A}} \times N^A_{\mathcal{A}} + 2 \text{ since } N^A_{\mathcal{A}} > 1.
\end{array}
$$

$$
\begin{array}{lrcl}
\psi_R : & (s +_{\mathrm{R}} M^{A\supset B})^{A\supset B}N^A & \to & s +_{\mathrm{R}} (M^{A\supset B}N^A)^B \\
 & & & \text{Same as } \psi_L.
\end{array}
$$

$$
\begin{array}{lrcl}
\phi_L : & N^B\langle v^A{:=}r, (M^{[\![r]\!]A} +_{\mathrm{L}} t)\rangle & \to & (N^B\langle v^A{:=}r, M\rangle)^{B\{v^A\leftarrow r\}} +_{\mathrm{L}} t \\
 & N^B_{\mathcal{A}} \times (2 \times M^{[\![r]\!]A}_{\mathcal{A}} + 2) + 1 & > & 2 \times (N^B_{\mathcal{A}} \times M^{[\![r]\!]A}_{\mathcal{A}} + 1) + 2 \text{ since } N^B_{\mathcal{A}} > 1.5.
\end{array}
$$

$$
\begin{array}{lrcl}
\phi_R : & N^B\langle v^A{:=}r, (s +_{\mathrm{R}} M^{[\![r]\!]A})\rangle & \to & s +_{\mathrm{R}} (N^B\langle v^A{:=}r, M\rangle)^{B\{v^A\leftarrow r\}} \\
 & & & \text{Same as } \phi_L.
\end{array}
$$

$$
\begin{array}{lrcl}
\chi_L : & [\beta^A](M^A +_{\mathrm{L}} t)^A & \to & ([\beta^A]M^A)^{\perp} +_{\mathrm{L}} t \\
 & 2 \times (2 \times M^A_{\mathcal{A}} + 2) & > & 2 \times 2 \times M^A_{\mathcal{A}} + 2
\end{array}
$$

$$
\begin{array}{lrcl}
\chi_R : & [\beta^B](s +_{\mathrm{R}} N^B)^B & \to & s +_{\mathrm{R}} ([\beta^B]N^B)^{\perp} \\
 & & & \text{Same as } \chi_L.
\end{array}
$$

$$
\begin{array}{lrcl}
\iota_L : & \mu\alpha^A.(M^{\perp} +_{\mathrm{L}} t)^{\perp} & \to & (\mu\alpha^A.M^{\perp}) +_{\mathrm{L}} N^B \text{ if } \alpha^A \notin \mathsf{FVF}(t) \\
 & 2 \times (2 \times M^A_{\mathcal{A}} + 2) & > & 2 \times 2 \times M^A_{\mathcal{A}} + 2
\end{array}
$$

$$
\begin{array}{lrcl}
\iota_R : & \mu\alpha^A.(s +_{\mathrm{R}} N^{\perp})^{\perp} & \to & M^B +_{\mathrm{R}} (\mu\alpha^A.N^{\perp}) \text{ if } \alpha^A \notin \mathsf{FVF}(s) \\
 & & & \text{Same as } \iota_L.
\end{array}
$$

Thus, permutative reduction is polynomially terminating, and therefore SN. □

Proposition 4.5.14 Every typable λ^{LP}-term is SN.

Proof.- By contradiction. Assume that there is an infinite reduction sequence starting from a typable λ^{LP}-term M_0. We distinguish between principal reductions ($\overset{\mathrm{B}}{\to}$) and permutative reductions ($\overset{\mathrm{P}}{\to}$) within this sequence.

Since, by Lemma 4.5.13, permutative reduction is SN, our sequence must contain an infinite number of principal reduction steps. Between any two principal steps, there may be 0 or more permutative steps (always a finite number). Therefore, the reduction sequence has the form:

$$M_0 \xrightarrow{\mathrm{P}}\!\!\!\!\twoheadrightarrow M_0' \xrightarrow{\mathrm{B}} M_1 \xrightarrow{\mathrm{P}}\!\!\!\!\twoheadrightarrow M_1' \xrightarrow{\mathrm{B}} M_2 \xrightarrow{\mathrm{P}}\!\!\!\!\twoheadrightarrow M_2' \xrightarrow{\mathrm{B}} \cdots$$

Additionally, by Lemma 4.5.12, $\langle\!| M_i |\!\rangle = \langle\!| M_i' |\!\rangle$ for every i. Also, by Lemma 4.5.11, we know that for every i, $\langle\!| M_i |\!\rangle \to^+ \langle\!| M_{i+1} |\!\rangle$ in $\lambda\mu^1$. We can therefore construct an infinite $\lambda\mu^1$-reduction sequence:

$$\langle\!| M_0 |\!\rangle \to^+ \langle\!| M_1 |\!\rangle \to^+ \langle\!| M_2 |\!\rangle \to^+ \cdots$$

However, M_0 is typable in λ^{LP} and, by Lemma 4.4.20, so is every M_i. Since the mapping preserves typability (Lemma 4.5.5), then we have an infinite reduction sequence of typable $\lambda\mu^1$-terms. This is an absurd, since reduction of typable $\lambda\mu^1$-terms is SN. Therefore, there cannot be an infinite reduction sequence starting from a typable λ^{LP}-term. □

4.6 Confluence

This section addresses confluence of λ^{LP}, namely that if $M_0 \twoheadrightarrow M_1$ and $M_0 \twoheadrightarrow M_2$, then there exists M_3 s.t. $M_1 \twoheadrightarrow M_3$ and $M_2 \twoheadrightarrow M_3$. Confluence is an immediate consequence (via Newman's Lemma) of the fact that λ^{LP} is strongly normalising and that all critical pairs are joinable. Regarding the latter point, note that λ^{LP} has the following critical pairs:

- $\mu - \theta :\ [\beta^A]\mu\alpha^A.[\alpha^A]M^A$ with $\alpha^A \notin \mathsf{FVF}(M^A)$
- $\mu - \iota_L :\ [\beta^A]\mu\alpha^A.M^{\perp} +_{\mathrm{L}} t$ with $\alpha^A \notin \mathsf{FVF}(t)$
- $\mu - \iota_R :\ [\beta^A]\mu\alpha^A.s +_{\mathrm{R}} M^{\perp}$ with $\alpha^A \notin \mathsf{FVF}(s)$
- $\zeta - \theta :\ (\mu\alpha^{A\supset B}.[\alpha^{A\supset B}]M^{A\supset B})N^A$ with $\alpha^{A\supset B} \notin \mathsf{FVF}(M^{A\supset B})$
- $\zeta - \iota_L :\ (\mu\alpha^{A\supset B}.M^{\perp} +_{\mathrm{L}} t)N^A$ with $\alpha^A \notin \mathsf{FVF}(t)$
- $\zeta - \iota_R :\ (\mu\alpha^{A\supset B}.s +_{\mathrm{R}} M^{\perp})N^A$ with $\alpha^A \notin \mathsf{FVF}(s)$
- $\theta - \mu :\ \mu\beta^A.[\beta^A]\mu\alpha^A.M^A$ with $\beta^A \notin \mathsf{FVF}(M^A)$
- $\theta - \chi_L :\ \mu\alpha^A.[\alpha^A]M^A +_{\mathrm{L}} t$ with $\alpha^A \notin \mathsf{FVF}(t)$
- $\theta - \chi_R :\ \mu\alpha^A.[\alpha^A]s +_{\mathrm{R}} M^A$ with $\alpha^A \notin \mathsf{FVF}(s)$

The critical pairs involving ι_R, ϕ_R or χ_R are analogous to those involving ι_L, ϕ_L or χ_L respectively. Thus, we will only show the diagrams for the latter.

μ-θ Critical pair:

$$\begin{tikzcd}
{[\beta^A]\mu\alpha^A.[\alpha^A]M^A} \arrow[d, bend right] \arrow[d, bend left] \\
{[\beta^A]M^A}
\end{tikzcd}$$

μ-ι_L Critical pair:

$$\begin{tikzcd}
& {[\beta^A]\mu\alpha^A.M^\perp +_{\mathrm{L}} t} \arrow[dl] \arrow[dr] & \\
{[\beta^A]M^\perp\{\alpha^A \leftarrow \beta^A\} +_{\mathrm{L}} t} \arrow[d] & & {[\beta^A](\mu\alpha^A.M^\perp) +_{\mathrm{L}} t} \arrow[d] \\
{([\beta^A]M^\perp\{\alpha^A \leftarrow \beta^A\}) +_{\mathrm{L}} t} & & {([\beta^A]\mu\alpha^A.M^\perp) +_{\mathrm{L}} t} \arrow[ll]
\end{tikzcd}$$

ζ-θ Critical pair:

$$\begin{tikzcd}
& {(\mu\alpha^{A\supset B}.[\alpha^{A\supset B}]M^{A\supset B})N^A} \arrow[dl] \arrow[dr] & \\
{\mu\beta^B.[\beta^B]M^{A\supset B}N^A} \arrow[rr] & & {M^{A\supset B}N^A}
\end{tikzcd}$$

ζ-ι_L Critical pair:

$$\begin{tikzcd}
& {(\mu\alpha^{A\supset B}.M^\perp +_{\mathrm{L}} t)N^A} \arrow[dl] \arrow[dr] & \\
{\mu\beta^B.M' +_{\mathrm{L}} t} \arrow[dr] & & {(\mu\alpha^{A\supset B}.M^\perp)N^A +_{\mathrm{L}} t} \arrow[dl] \\
& {(\mu\beta^B.M') +_{\mathrm{L}} t} &
\end{tikzcd}$$

where $M' = M^\perp(\!| {}_{[\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)N^A} |\!)$

θ-μ Critical pair:

$$\begin{tikzcd}
& {\mu\beta^A.[\beta^A]\mu\alpha^A.M^A} \arrow[dl] \arrow[dr] & \\
{\mu\alpha^A.M^A} \arrow[rr, dotted, "=_\alpha"] & & {\mu\beta^A.M^A\{\alpha^A \leftarrow \beta^A\}}
\end{tikzcd}$$

θ-χ_L Critical pair:

$$\begin{tikzcd}
& {\mu\alpha^A.[\alpha^A]M^A +_{\mathrm{L}} t} \arrow[dl] \arrow[dr] & \\
{M^A +_{\mathrm{L}} t} & & {\mu\alpha^A.([\alpha^A]M^A) +_{\mathrm{L}} t} \arrow[dl] \\
& {(\mu\alpha^A.[\alpha^A]M^A) +_{\mathrm{L}} t} \arrow[ul] &
\end{tikzcd}$$

Proposition 4.6.1 (Confluence) Reduction in λ^{LP} is confluent.

Proof.- We have already proved (Proposition 4.5.14) that normalisation of derivations in HLP is terminating. Therefore, by Newman's Lemma ([New42]), λ^{LP} is confluent. $\square$

We now address our hitherto pending proof of consistency of the $\equiv$ theory.

Lemma 4.6.2 If $\rhd_{\lambda^{\mathsf{LP}}} \Theta; \Gamma; \Delta \vdash s \equiv t : B$, then there are terms M_1, M_2, M_3 s.t. :

1. $\rhd_{\lambda^{\mathsf{LP}}} \Theta; \Gamma; \Delta \vdash M_1^B \mid s$;
2. $\rhd_{\lambda^{\mathsf{LP}}} \Theta; \Gamma; \Delta \vdash M_2^B \mid t$;
3. $M_1 \twoheadrightarrow M_3$; and
4. $M_2 \twoheadrightarrow M_3$.

Proof.- By induction on the derivation of $\Theta; \Gamma; \Delta \vdash s \equiv t : B$. Since there may be more than one candidate for M_1 and M_2, we will assume that the derivations we are working with are **canonical** in the following sense:

- Whenever either (T-)PlusL or (T-)PlusR can be used to prove the same formula, we use (T-)PlusL. This eliminates the possibility of two different terms encoding a proof with a witness of the form $s' + t'$. (T-)PlusR may still be used to derive judgements of the form $\Theta'; \Gamma'; \Delta' \vdash (s' +_{\mathrm{R}} M^{A'})^{A'} \,|\, s' + t'$ when $\Theta'; \Gamma'; \Delta' \vdash A' \,|\, s'$ is not derivable.
- We use (T-)□I' instead of (T-)□I, in order to avoid the possibility of multiple (in fact infinite) terms encoding a proof with a witness of the form $!s'$.

This way, if $\rhd_{\mathsf{HLP}} \Theta'; \Gamma'; \Delta' \vdash A \,|\, r$, there exactly one term M s.t. $\Theta'; \Gamma'; \Delta' \vdash M^A \,|\, r$ has a canonical derivation (this is straightforward by induction on s, since canonical derivations are syntax-driven).

In order to preserve this invariant, we will ensure that the derivations we construct are also canonical in this sense (by using T-□I' instead of T-□I, and not using T-PlusR unless said rule was used in the original derivation).

We analyze the last rule used.

- Eq-β: $s = (\lambda x^C.s') \cdot t'$, $t = s'\{x^C \leftarrow t'\}$, and both $\Theta; \Gamma, x^C; \Delta \vdash M^B \,|\, s'$ and $\Theta; \Gamma; \Delta \vdash N^C \,|\, t'$ are derivable by hypothesis (for some M^B and N^C which encode the respective derivations). Using T-⊃I on the former, we can derive $\Theta; \Gamma; \Delta \vdash (\lambda x^C.M^B)^{C \supset B} \,|\, \lambda x^C.s'$, and then obtain $\Theta; \Gamma; \Delta \vdash (\lambda x^C.M^B)^{C \supset B} \cdot N^B \,|\, (\lambda x^C.s') \cdot t'$ by T-⊃E.

  $\Theta; \Gamma; \Delta \vdash (M^B\{x^C \leftarrow N^C\})^B \,|\, t$ can be derived from the premises by Lemma 4.4.12. Take $M_1 = (\lambda x^C.M^B)^{C \supset B} N^C$, $M_2 = M_3 = M^B\{x^C \leftarrow N^C\}$. $M_1 \rightarrow_\beta M_3$.
- Eq-γ: $B = C\{v^A \leftarrow s'\}$, $s = t'\langle v^A{:=}s', !s'\rangle$, $t = t'\{v^A \leftarrow s'\}$ and both $\Theta; \cdot; \cdot \vdash N^A \,|\, s'$ and $\Theta, v^A; \Gamma; \Delta \vdash M^C \,|\, t'$ are derivable by hypothesis (for some M^C and N^A). Take $M_1 = (M^C\langle v^A{:=}s', (!N^A)^{[\![s']\!]A}\rangle)^{C\{v^D \leftarrow s'\}}$ and $M_2 = M_3 = M^C\{v^A \leftarrow N^A, s'\}$. We can obtain $\Theta; \Gamma; \Delta \vdash M_1^B \,|\, s$ by using T-□I' and T-□E.

  $\Theta; \Gamma; \Delta \vdash B \,|\, t$ can be derived from the premises by Lemma 4.4.13, and $M_1 \rightarrow_\gamma M_2$.
- Eq-μ: $B = \bot$, $s = [\beta^C]\mu\alpha^C.s'$, $t = s'\{\alpha^C \leftarrow \beta^C\}$, $\Delta = \Delta', \beta^C$ and $\Theta; \Gamma; \Delta', \alpha^C, \beta^C \vdash \bot^M \,|\, s'$ is derivable by hypothesis for some $M^\perp$. By T-NAbs, we can derive $\Theta; \Gamma; \Delta', \beta^C \vdash (\mu\alpha^C.M^\perp)^C \,|\, \mu\alpha^C.s'$, and then get $\Theta; \Gamma; \Delta', \beta^C \vdash ([\beta^C]\mu\alpha^C.M^\perp)^\perp \,|\, [\beta^C]\mu\alpha^C.s'$ by T-Name. Take $M_1 = [\beta^C]\mu\alpha^C.M^\perp$ and $M_2 = M_3 = M^\perp\{\alpha^C \leftarrow \beta^C\}$.

  $\Theta; \Gamma; \Delta \vdash M_2^B \,|\, t$ can be derived from the premises by Lemma 4.4.15, and $M_1 \rightarrow_\mu M_2$.
- Eq-ζ: $s = (\mu\alpha^{A \supset B}.s') \cdot t'$, $t = \mu\beta^B.s'(\!|\, [\alpha^{A \supset B}](\bullet) \leftarrow [\beta^B](\bullet)t' \,|\!)$ and both $\Theta; \Gamma; \Delta, \alpha^{A \supset B} \vdash M^\perp \,|\, s'$ and $\Theta; \Gamma; \Delta \vdash N^A \,|\, t'$ are derivable by hypothesis for some $M^\perp$ and N^A. Take $M_1 = (\mu\alpha^{A \supset B}.M^\perp)^{A \supset B} \cdot N^A$, and $M_2 = M_3 = \mu\beta^B.M^\perp(\!|\, [\alpha^{A \supset B}](\bullet) \leftarrow [\beta^B](\bullet)N^A \,|\!)$. By T-NAbs, we can derive $\Theta; \Gamma; \Delta \vdash (\mu\alpha^{A \supset B}.M^\perp)^{A \supset B} \,|\, \mu\alpha^{A \supset B}.s'$, and then obtain $\Theta; \Gamma; ((\mu\alpha^{A \supset B}.M^\perp)^{A \supset B} \cdot N^A) \vdash B^\Delta \,|\, (\mu\alpha^{A \supset B}.s') \cdot t'$ by T-⊃E.

  $\Theta; \Gamma; \Delta \vdash M_2^B \,|\, t$ can be derived from the premises by Lemma 4.4.16, and $M_1 \rightarrow_\zeta M_2$.
- Eq-θ: $s = \mu\alpha^B.[\alpha^B]t$ and $\Theta; \Gamma; \Delta \vdash M^B \,|\, t$ is derivable by hypothesis for some M^B. By Weakening, $\Theta; \Gamma; \Delta, \alpha^B \vdash M^B \,|\, t$ is also derivable. Take $M_1 = \mu\alpha^B.([\alpha^B]M^B)^\perp$ and $M_2 = M_3 = M^B$. We can use T-Name and T-NAbs to derive $\Theta; \Gamma; B \vdash M_1 \,|\, \Delta\mu\alpha^B.[\alpha^B]t$. Note that we already know $\Theta; \Gamma; B \vdash M \,|\, \Delta t$ is derivable, and $M_1 \rightarrow_\theta M^B$.

- Eq-ψ_L: $s = (s' + r) \cdot t'$, $t = (s' \cdot t') +_{\mathsf{L}} r$ and both $\Theta; \Gamma; \Delta \vdash M^{C \supset B} \mid s'$ and $\Theta; \Gamma; \Delta \vdash N^C \mid t'$ are derivable by hypothesis for some $M^{C \supset B}$ and N^C. Take $M_1 = (M^{C \supset B} +_{\mathsf{L}} r)^{C \supset B} N^C$ and $M_2 = M_3 = (M^{C \supset B} N^C)^B +_{\mathsf{L}} r$. By T-PlusL, we can derive $\Theta; \Gamma; \Delta \vdash M^{C \supset B} +_{\mathsf{L}} r^{C \supset B} \mid s' + r$. And then, by T-$\supset$E, we obtain $\Theta; \Gamma; \Delta \vdash M_1^B \mid s$. Conversely, using T-$\supset$E first, we can derive $\Theta; \Gamma; \Delta \vdash (M^{C \supset B} N^C)^B \mid s' \cdot t'$. And, by T-PlusL, $\Theta; \Gamma; \Delta \vdash M_2^B \mid t$. And $M_1 \to_{\psi_L} M_2$.
- Eq-ψ_R: this case is analogous to the previous one, using T-PlusR instead of T-PlusL, and ψ_R instead of ψ_L.
- Eq-ϕ_L: $B = C\{v^A \leftarrow r\}$, $s = u\langle v^A{:=}r, (s' + t')\rangle$, $t = u\langle v^A{:=}r, s'\rangle + t'$ and both $\Theta; \Gamma; [\![r]\!]A \vdash M \mid \Delta$s' and $\Theta, v^A; \Gamma; C \vdash N \mid \Delta$u are derivable by hypothesis for some $M^{[\![r]\!]A}$ and N^C. Take $M_1 = N^C\langle v^A{:=}r, (M^{[\![r]\!]A} +_{\mathsf{L}} t')\rangle$ and $M_2 = M_3 = N^C\langle v^A{:=}r, M^{[\![r]\!]A}\rangle +_{\mathsf{L}} t'$. $\Theta; \Gamma; \Delta \vdash M_1^B \mid s$ is obtained by using T-PlusL and then T-$\Box$E, while $\Theta; \Gamma; \Delta \vdash M_2^B \mid t$ is obtained by using T-$\Box$E and then T-PlusL, and $M_1 \to_{\phi_L} M_2$.
- Eq-ϕ_R: this case is analogous to the previous one, using T-PlusR instead of T-PlusL, and ϕ_R instead of ϕ_L.
- Eq-χ_L: analogous to Eq-ϕ_L and Eq-ψ_L, using T-PlusL and then T-Name to obtain $\Theta; \Gamma; B \vdash M_1 \mid \Delta$s, and conversely T-Name and then T-PlusL to obtain $\Theta; \Gamma; B \vdash M_2 \mid \Delta$t. $M_1 \to_{\chi_L} M_2 = M_3$.
- Eq-χ_R: this case is analogous to the previous one, using T-PlusR instead of T-PlusL, and χ_R instead of χ_L.
- Eq-ι_L: analogous to the previous cases, using T-PlusL, T-NAbs and ι_L.
- Eq-ι_R: this case is analogous to the previous one, using T-PlusR instead of T-PlusL, and ι_R instead of ι_L.
- Eq-Refl: $t = s$ and $\Theta; \Gamma; \Delta \vdash M^B \mid s$ is already derivable by hypothesis for some M^B. Take $M_1 = M_2 = M_3 = M^B$.
- Eq-Symm: here $\Theta; \Gamma; \Delta \vdash (M_1')^B \mid s$ and $\Theta; \Gamma; \Delta \vdash (M'2)^B \mid t$ are derivable by IH for some M_1' and M_2'. Take $M_1 = M_1'$ and $M_2 = M_2'$. By IH, there is a term M_3' s.t. $M_1 \twoheadrightarrow M_3'$ and $M_2 \twoheadrightarrow M_3'$. Take $M_3 = M_3'$.
- Eq-Trans: since $\Theta; \Gamma; \Delta \vdash s \equiv r : B$ and $\Theta; \Gamma; \Delta \vdash r \equiv t : B$ are derivable by hypothesis, we know by IH that there are M_1', M_2', M_3', M_4' and M_5' s.t. $\Theta; \Gamma; \Delta \vdash (M_1')^B \mid s$, $\Theta; \Gamma; \Delta \vdash (M_2')^B \mid r$ $\Theta; \Gamma; \Delta \vdash (M_4')^B \mid t$, $M_1' \twoheadrightarrow M_3'$, $M_2' \twoheadrightarrow M_3'$, $M_2' \twoheadrightarrow M_5'$ and $M_4' \twoheadrightarrow M_5'$. (we know that M_1' is the same on both sides of the IH since we are working with canonical derivations).

  Since $M_2' \twoheadrightarrow M_3'$ and $M_2' \twoheadrightarrow M_5'$, by confluence there is a term M_6' s.t. $M_3' \twoheadrightarrow M_6'$ and $M_5' \twoheadrightarrow M_6'$. Take $M_1 = M_1'$, $M_2 = M_4'$ and $M_3 = M_6'$. $M_1 \twoheadrightarrow M_3' \twoheadrightarrow M_3$ and $M_2 \twoheadrightarrow M_5' \twoheadrightarrow M_3$.
- Eq-λ: $B = A \supset C$, $s = \lambda x^A.s'$, $t = \lambda x^A.t'$ and, by IH, both $\Theta; \Gamma, x^A; \Delta \vdash (M_1')^C \mid s'$ and $\Theta; \Gamma, x^A; C \vdash (M_2') \mid \Delta t'$ are derivable for some M_1' and M_2', and there is a term M_3' s.t. $M_1' \twoheadrightarrow M_3'$ and $M_2' \twoheadrightarrow M_3'$. Take $M_1 = \lambda x^A.M_1'$, $M_2 = \lambda x^A.M_2'$, $M_3 = \lambda x^A.M_3'$. We obtain $\Theta; \Gamma; \Delta \vdash M_1^{A \supset C} \mid \lambda x^A.s'$ and $\Theta; \Gamma; \Delta \vdash M_2^{A \supset C} \mid \lambda x^A.t'$ by T-$\supset$I, and $M_1 \twoheadrightarrow M_3$ and $M_2 \twoheadrightarrow M_3$ by means of internal reductions.

- All other cases (Eq-$\cdot$, Eq-$\langle\rangle$, Eq-$[\alpha]$, Eq-$\mu\alpha$, Eq-$+_{\mathrm{L}}$ and Eq-$+_{\mathrm{R}}$) are analogous to Eq-λ, using the IH and the corresponding inference rule on both sides.

□

Corollary 4.6.3 The $\equiv$ theory is consistent.

Proof.- Suppose that we can derive $\cdot;\{x^A,y^A\};\cdot \vdash x^A \equiv y^A : A$. By Lemma 4.6.2, there are M_1, M_2 and M_3 s.t. both $\cdot;\{x^A,y^A\};\cdot \vdash M_1^A \,|\, x^A$ and $\cdot;\{x^A,y^A\};\cdot \vdash M_2^A \,|\, y^A$ are derivable, and $M_1 \twoheadrightarrow M_3$ and $M_2 \twoheadrightarrow M_3$. By Lemma 4.4.5, $x^A = \mathsf{w}(M_1)$ and $y^A = \mathsf{w}(M_2)$. By Definition 4.4.3, M_1 can only be x^A and M_2 can only be y^A. But both x^A and M_2 and y^A are normal forms, so M_3 cannot exist. Therefore, $\cdot;\{x^A,y^A\};\cdot \vdash x^A \equiv y^A : A$ is not derivable.

Since $\cdot;\{x^A,y^A\};\cdot \vdash A \,|\, x^A$ and $\cdot;\{x^A,y^A\};\cdot \vdash A \,|\, y^A$ are both derivable (by Var), then the $\equiv$ theory is consistent. □

4.7 Additional Considerations

4.7.1 Additional Permutative Rules

We briefly comment on the λ_p^{LP}-calculus, resulting from adding the following permutative reduction rules to λ^{LP}, where υ_θ is subject to the condition that $\alpha^A \notin \mathsf{FVF}(N^{[\![r]\!]B})$:

$$\begin{array}{lcl}
\upsilon_\beta : M_1^{A\supset B}\langle v^C{:=}r, M_2^{[\![r]\!]C}\rangle N^A & \to & (M_1^{A\supset B} N^A)\langle v^C{:=}r, M_2^{[\![r]\!]C}\rangle \\
\upsilon_\gamma : M^A\langle v^B{:=}r, N_1^{[\![r]\!]B}\langle u^C{:=}s, N_2^{[\![s]\!]C}\rangle\rangle & \to & M^A\langle v^B{:=}r, N_1^{[\![r]\!]B}\rangle\langle u^C{:=}s, N_2^{[\![s]\!]C}\rangle \\
\upsilon_\mu : [\beta^B](M^B\langle v^A{:=}r, N^{[\![r]\!]A}\rangle) & \to & ([\beta^B]M^B)\langle v^A{:=}r, N^{[\![r]\!]A}\rangle \\
\upsilon_\theta : \mu\alpha^A.(M^\perp\langle v^B{:=}r, N^{[\![r]\!]B}\rangle) & \to & (\mu\alpha^A.M^\perp)\langle v^B{:=}r, N^{[\![r]\!]B}\rangle
\end{array}$$

The restriction to rule υ_θ prevents the creation of free variables upon reduction.

Proof witness equivalence must also be augmented with the inference schemes of Fig. 4.9. Type Preservation is then seen to hold. Also, the following new critical pairs appear:

- $\mu - \upsilon_\theta :\ [\beta^A]\mu\alpha^A.(M^\perp\langle v^B{:=}r, N^{[\![r]\!]B}\rangle)$, with $\alpha^A \notin \mathsf{FVF}(N^{[\![r]\!]B})$
- $\zeta - \upsilon_\theta :\ (\mu\alpha^{A\supset B}.M_1^\perp\langle v^C{:=}r, N^{[\![r]\!]C}\rangle)M_2^A$, with $\alpha^A \notin \mathsf{FVF}(N^{[\![r]\!]C})$
- $\upsilon_\beta - \gamma :\ (M_1^{A\supset B}\langle v^C{:=}r, !N^C\rangle)M_2^A$, with $v^C \notin \mathsf{FVV}(M_2^A)$, $\mathsf{FVT}(N^C) = \mathsf{FVT}(N^C) = \emptyset$
- $\upsilon_\beta - \phi_L :\ (M_1^{A\supset B}\langle v^C{:=}r, N^{[\![r]\!]C} +_{\mathrm{L}} t\rangle)M_2^A$
- $\upsilon_\beta - \phi_R :\ (M_1^{A\supset B}\langle v^C{:=}r, s +_{\mathrm{R}} N^{[\![r]\!]C}\rangle)M_2^A$
- $\upsilon_\beta - \upsilon_\gamma :\ (M_1^{A\supset B}\langle v^C{:=}r, M_2^{[\![r]\!]C}\langle u^F{:=}s, M_3^{[\![s]\!]F}\rangle\rangle)N^A$
- $\upsilon_\gamma - \gamma :\ M_1^A\langle v^B{:=}r, M_2^{[\![r]\!]B}\langle u^C{:=}s, !N^C\rangle\rangle$, with $\mathsf{FVT}(N^C) = \mathsf{FVT}(N^C) = \emptyset$
- $\upsilon_\gamma - \phi_L :\ M_1^A\langle v^B{:=}r, M_2^{[\![r]\!]B}\langle u^C{:=}s, M_3^{[\![s]\!]C} +_{\mathrm{L}} t\rangle\rangle$

$$\frac{\mathcal{H}, v^C \vdash A \supset B \,|\, s_1 \quad \mathcal{H} \vdash [\![r]\!]C \,|\, s_2 \quad \mathcal{H} \vdash A \,|\, t}{\mathcal{H} \vdash\ s_1\langle v^C{:=}r, s_2\rangle \cdot t \equiv (s_1 \cdot t)\langle v^C{:=}r, s_2\rangle : B\{v^C \leftarrow r\}}\ \mathsf{Eq}\text{-}\upsilon_\beta$$

$$\frac{\mathcal{H}, v^B, u^C \vdash A \,|\, t_1 \quad \mathcal{H}, u^C \vdash [\![r]\!]B \,|\, t_2 \quad \mathcal{H} \vdash [\![s]\!]C \,|\, t_3}{\mathcal{H} \vdash\ t_1\langle v^B{:=}r, t_2\langle u^C{:=}s, t_3\rangle\rangle \equiv t_1\langle v^B{:=}r, t_2\rangle\langle u^C{:=}s, t_3\rangle : A\{v^B \leftarrow r\}}\ \mathsf{Eq}\text{-}\upsilon_\gamma$$

$$\frac{\mathcal{H}, v^A, \beta^B \vdash B \,|\, s \quad \mathcal{H}, \beta^B \vdash [\![r]\!]A \,|\, t}{\mathcal{H}, \beta^B \vdash\ [\beta^B](s\langle v^A{:=}r, t\rangle) \equiv ([\beta^B]s)\langle v^A{:=}r, t\rangle : \bot}\ \mathsf{Eq}\text{-}\upsilon_\mu$$

$$\frac{\mathcal{H}, v^B, \alpha^A \vdash \bot \,|\, s \quad \mathcal{H} \vdash [\![r]\!]B \,|\, t \quad \alpha^A \notin \mathsf{FVF}(N^{[\![r]\!]B})}{\mathcal{H} \vdash\ \mu\alpha^A.(s\langle v^B{:=}r, t\rangle) \equiv (\mu\alpha^A.s)\langle v^B{:=}r, t\rangle : A}\ \mathsf{Eq}\text{-}\upsilon_\theta$$

Figure 4.9: Additional proof witness equivalence schemes.

- $\upsilon_\gamma - \phi_R :\ M_1^A\langle v^B{:=}r, M_2^{[\![r]\!]B}\langle u^C{:=}s, t +_{\mathrm{R}} M_3^{[\![s]\!]C}\rangle\rangle$
- $\upsilon_\gamma - \upsilon_\gamma :\ M_1^A\langle v^B{:=}r, M_2^{[\![r]\!]B}\langle u^C{:=}s, M_3^{[\![s]\!]C}\langle w^F{:=}t, M_4^{[\![w]\!]F}\rangle\rangle\rangle$
- $\upsilon_\mu - \gamma :\ [\alpha^A](M^A\langle v^B{:=}r, !N^B\rangle)$, with $\mathsf{FVT}(N^B) = \mathsf{FVT}(N^B) = \emptyset$
- $\upsilon_\mu - \phi_L :\ [\alpha^A](M^A\langle v^B{:=}r, N^{[\![r]\!]B} +_{\mathrm{L}} t\rangle)$
- $\upsilon_\mu - \phi_R :\ [\alpha^A](M^A\langle v^B{:=}r, s +_{\mathrm{R}} N^{[\![r]\!]B}\rangle)$
- $\upsilon_\mu - \upsilon_\gamma :\ [\alpha^A](M_1^A\langle v^B{:=}r, M_2^{[\![r]\!]B}\langle u^C{:=}s, M_3^{[\![s]\!]C}\rangle\rangle)$
- $\upsilon_\theta - \gamma :\ \mu\alpha^A.(M^A\langle v^B{:=}r, !N^B\rangle)$, with $\mathsf{FVT}(N^B) = \mathsf{FVT}(N^B) = \emptyset$
- $\upsilon_\theta - \phi_L :\ \mu\alpha^A.(M^A\langle v^B{:=}r, N^{[\![r]\!]B} +_{\mathrm{L}} t\rangle)$, with $\alpha^A \notin \mathsf{FVF}(N^{[\![r]\!]B} +_{\mathrm{L}} t)$
- $\upsilon_\theta - \phi_R :\ \mu\alpha^A.(M^A\langle v^B{:=}r, s +_{\mathrm{R}} N^{[\![r]\!]B}\rangle)$, with $\alpha^A \notin \mathsf{FVF}(s +_{\mathrm{R}} N^{[\![r]\!]B})$
- $\upsilon_\theta - \upsilon_\gamma :\ \mu\alpha^A.(M_1^A\langle v^B{:=}r, M_2^{[\![r]\!]B}\langle u^C{:=}s, M_3^{[\![s]\!]C}\rangle\rangle)$, with $\alpha^A \notin \mathsf{FVF}(M_2^{[\![r]\!]B}) \cup \mathsf{FVF}(M_3^{[\![s]\!]C})$

These are all joinable (Sec. B.3.1). Regarding SN, $\lambda_p^{\mathsf{LP}} \backslash \{\upsilon_\mu, \upsilon_\theta\}$ may be proved SN by translating it into the $\lambda\mu^{\to\wedge\vee\bot}$-calculus of [dG01] adapting the translation of [NPP08]. The translation is the same as the one shown in Sec. 4.5, except for the following clauses:

$$\begin{aligned}
\langle\!| v^A |\!\rangle &\triangleq v^{\langle\!| A |\!\rangle} \\
\langle\!| (!M^A)^{[\![s]\!]A} |\!\rangle &\triangleq \iota_1(\langle\!| M^A |\!\rangle) \\
\langle\!| (N^B\langle v^A{:=}r, M^{[\![r]\!]A}\rangle)^{B\{v^A \leftarrow r\}} |\!\rangle &\triangleq \delta(\langle\!| N^B |\!\rangle, v^{\langle\!| A |\!\rangle}.\langle\!| M^A |\!\rangle, v^{\langle\!| A |\!\rangle}.\langle\!| M^A |\!\rangle)
\end{aligned}$$

where ι_1 is the term denoting a left injection into a disjoint union and δ is the case elimination construct of the disjoint union. Note that the translated versions of rules υ_β and υ_γ are already present in $\lambda\mu^{\to\wedge\vee\bot}$, since this calculus already includes the two permutation rules:

$$\frac{}{\Theta;\Gamma;\Delta \vdash 0^{\mathbb{N}} \mid 0}\ \mathsf{Zero} \qquad \frac{\Theta;\Gamma;\Delta \vdash M^{\mathbb{N}} \mid s}{\Theta;\Gamma;\Delta \vdash (\mathtt{succ}\ M^{\mathbb{N}})^{\mathbb{N}} \mid \mathtt{succ}\ s}\ \mathsf{Succ}$$

$$\frac{\Theta;\Gamma;\Delta \vdash N^{\mathbb{N}} \mid r \quad \Theta;\Gamma;\Delta \vdash L^{A} \mid s \quad \Theta;\Gamma,x^{\mathbb{N}};\Delta \vdash M^{A} \mid t}{\Theta;\Gamma;\Delta \vdash (\mathtt{case}\ N^{\mathbb{N}}\ \mathtt{of}\ 0\colon L^{A}, \mathtt{succ}\ x^{\mathbb{N}}\colon M^{A})^{A} \mid \mathtt{case}\ r\colon s, x^{\mathbb{N}}\colon t}\ \mathsf{CaseNat}$$

$$\frac{\Theta;\Gamma;\Delta \vdash M^{A\supset A} \mid s}{\Theta;\Gamma;\Delta \vdash (\mathtt{fix}\ M^{A\supset A})^{A} \mid \mathtt{fix}\ s}\ \mathsf{Fix}$$

Figure 4.10: Natural numbers and recursion

$$\begin{array}{rcl} \delta(M, x.N, y.O)\ P & \to & \delta(M, x.N\ P, y.O\ P) \\ \delta(\delta(M, x.N, y.O), u.P, v.Q) & \to & \delta(M, x.\delta(N, u.P, v.Q), y.\delta(O, u.P, v.Q)) \end{array}$$

For $\lambda_p^{\mathsf{LP}}\backslash\{\upsilon_\mu, \upsilon_\theta\}$ we thus obtain confluence from Newman's Lemma. As regards SN for λ_p^{LP}, we conjecture that it should be easily obtainable by adapting the approach mentioned above for $\lambda_p^{\mathsf{LP}}\backslash\{\upsilon_\mu, \upsilon_\theta\}$.

Remark 4.7.1 All the metatheoretical results shown in Section 4.4 hold for this extension of HLP. The only results which need to be extended with the new reduction rules are Lemmas 4.4.19 and 4.2.15. The proofs of the additional cases for these lemmas can be found in Appendix B.3.

4.7.2 Extending the Language with Natural Numbers

Data structures are an important feature in programming languages. Like CL_P (see Section 3.5.2), HLP can also be extended with constructors and basic data types. For example, we can extend our calculus with natural numbers by introducing the primitive type `Nat`, and the constructors "`0`" and "`succ`", as well as the "`case`" observer, and the fix point operator "`fix`" to introduce recursion. The associated typing rules are given in Fig. 4.10.

Suppose we have a product operator "*" of type $\mathbb{N}\supset\mathbb{N}\supset\mathbb{N}$. We can define a certified power function as follows:

$$\mathtt{cpow} \ \triangleq\ !\mathtt{fix}\ \lambda p^{\mathbb{N}\supset\mathbb{N}\supset\mathbb{N}}.\lambda n^{\mathbb{N}}.\mathtt{case}\ n\ \mathtt{of}\ 0\colon (\lambda x^{\mathbb{N}}.\underline{1}), \mathtt{succ}\ m^{\mathbb{N}}\colon \lambda y^{\mathbb{N}}.x * (p\, m\, y)$$

where $\underline{1} \ \triangleq\ \mathtt{succ}\ 0$. The following judgement can be derived:

$$\begin{array}{l} \cdot;\cdot;\cdot \vdash \mathtt{cpow}^{[\![\mathtt{fix}\ \lambda p^{\mathbb{N}\supset\mathbb{N}\supset\mathbb{N}}.\lambda n^{\mathbb{N}}.\mathtt{case}\ n:(\lambda x^{\mathbb{N}}.\underline{1}), m^{\mathbb{N}}:\lambda y^{\mathbb{N}}.y*(p\cdot m\cdot y)]\!](\mathbb{N}\supset\mathbb{N}\supset\mathbb{N})} \mid \\ !\mathtt{fix}\ \lambda p^{\mathbb{N}\supset\mathbb{N}\supset\mathbb{N}}.\lambda n^{\mathbb{N}}.\mathtt{case}\ n\colon (\lambda x^{\mathbb{N}}.\underline{1}), m^{\mathbb{N}}\colon \lambda y^{\mathbb{N}}.y * (p \cdot m \cdot y) \end{array}$$

However, defining a power function of type $\mathbb{N} \supset [\![s]\!](\mathbb{N} \supset \mathbb{N})$ requires the witness s to be infinite. Consider for example the following expression:

$$\begin{array}{rcl} \mathtt{pow} & \triangleq & \mathtt{fix}\ \lambda p^{\mathbb{N}\supset[\![s]\!](\mathbb{N}\supset\mathbb{N})}.\lambda n^{\mathbb{N}}. \\ & & \mathtt{case}\ n\ \mathtt{of}\ 0\colon !(\lambda x^{\mathbb{N}}.\underline{1})+_{\mathrm{L}} t, \mathtt{succ}\ m^{\mathbb{N}}\colon (!(r +_{\mathrm{R}} \lambda y^{\mathbb{N}}.y * (vy)))\langle v^{\mathbb{N}\supset\mathbb{N}} := s, pm\rangle \end{array}$$

We still need to define the proof witnesses r, s and t. Keep in mind that both sides of the `case` must have the same type. We can derive the following judgement:

$$\cdot; p^{\mathbb{N}\supset[\![s]\!](\mathbb{N}\supset\mathbb{N})}; \cdot \vdash (!(\lambda x^{\mathbb{N}}.\underline{1})+_{\mathrm{L}} t)^{[\![(\lambda x^{\mathbb{N}}.\underline{1})+t]\!](\mathbb{N}\supset\mathbb{N})} \mid !(\lambda x^{\mathbb{N}}.\underline{1}) + t$$

Similarly, we can derive:

$$\begin{array}{c}\cdot; m^{\mathbb{N}}, p^{\mathbb{N}\supset[\![s]\!](\mathbb{N}\supset\mathbb{N})}; \cdot \vdash \\ (!(r +_{\mathrm{R}} \lambda y^{\mathbb{N}}.y * (vy)))\langle v^{\mathbb{N}\supset\mathbb{N}} := s, pm\rangle^{[\![r\{v^{\mathbb{N}\supset\mathbb{N}} \leftarrow s\} + \lambda y^{\mathbb{N}}.y*(s\cdot y)]\!]\mathbb{N}\supset\mathbb{N}} \mid (!(r +_{\mathrm{R}} \lambda y^{\mathbb{N}}.y * (v \cdot y)))\langle v^{\mathbb{N}\supset\mathbb{N}} := s, p \cdot m\rangle\end{array}$$

Since both sides of the `case` must have the same type, then $[\![(\lambda x^{\mathbb{N}}.\underline{1}) + t]\!](\mathbb{N} \supset \mathbb{N})$ must be the same as $[\![r\{v^{\mathbb{N}\supset\mathbb{N}} \leftarrow s\} + \lambda y^{\mathbb{N}}.y * (s \cdot y)]\!](\mathbb{N} \supset \mathbb{N})$. This equation can be easily solved with $r = \lambda x^{\mathbb{N}}.\underline{1}$ and $t = \lambda y^{\mathbb{N}}.y * (s \cdot y)$.

Therefore, the type of the `case` expression will be $[\![\lambda x^{\mathbb{N}}.\underline{1} + \lambda y^{\mathbb{N}}.y * (s \cdot y)]\!](\mathbb{N}\supset\mathbb{N})$.

Now let's look at s. Since the argument of `fix` is of type $(\mathbb{N} \supset [\![s]\!](\mathbb{N} \supset \mathbb{N})) \supset \mathbb{N} \supset [\![\lambda x^{\mathbb{N}}.\underline{1} + \lambda y^{\mathbb{N}}.y * (s \cdot y)]\!](\mathbb{N}\supset\mathbb{N})$, and the typing rule for `fix` requires its argument to be of type $A \supset A$ for some A, it then follows that

$$A = \mathbb{N} \supset [\![s]\!](\mathbb{N} \supset \mathbb{N}) = \mathbb{N} \supset [\![\lambda x^{\mathbb{N}}.\underline{1} + \lambda y^{\mathbb{N}}.y * (s \cdot y)]\!](\mathbb{N}\supset\mathbb{N})$$

and therefore $s = \lambda x^{\mathbb{N}}.\underline{1} + \lambda y^{\mathbb{N}}.y * (s \cdot y)$.

We can define this infinite (recursive) witness as $\mu s.\lambda x^{\mathbb{N}}.\underline{1} + \lambda y^{\mathbb{N}}.y * (s \cdot y)$. As a rule, $\mu s.C[s]$ will denote an infinite witness (with $C[s]$ being a proof witness which contains s in some position).

The need for infinite types generally arises when using ! within different subterms which must be of the same type. While the syntax of proof witnesses is extended to accommodate for recursive witnesses, there shall be no rules for introducing them. There are only three ways in which a recursive witness may appear in a derivation: if present within the type of a variable, as (or within) a witness introduced by PlusL or PlusR, or as a result of equivalence rules applied while using $\Box$I (for instance, if a β-expansion is used).

It would be tempting to avoid this issue by defining a power function the following way:

$$\texttt{pow'} \triangleq \lambda n^{\mathbb{N}}.!(\texttt{fix}\ (\lambda p^{\mathbb{N}\supset\mathbb{N}\supset\mathbb{N}}.\lambda x^{\mathbb{N}}.\texttt{case}\, x\, \texttt{of}\ \texttt{0}\colon (\lambda y^{\mathbb{N}}.\underline{1}), \texttt{succ}\ m^{\mathbb{N}}\colon \lambda z^{\mathbb{N}}.z * (p\, m\, z))n)$$

where $s = \texttt{fix}\ (\lambda p^{\mathbb{N}\supset\mathbb{N}\supset\mathbb{N}}.\lambda x^{\mathbb{N}}.\texttt{case}\, x\, \texttt{of}\ \texttt{0}\colon (\lambda y^{\mathbb{N}}.\underline{1}), \texttt{succ}\ m^{\mathbb{N}}\colon \lambda z^{\mathbb{N}}.z * (p\, m\, z))n$. However, the above term is not well-formed (it is not typable), since n appears free within the scope of "!".

4.7.3 Variations on Plus

As seen in the previous example, the "+" operator is required when two or more subterms must have the same type $[\![s]\!]A$ for some s and A. This is a common occurrence in computer programs (conditional branching, split function definitions). We can use a variant of the "+" rules specifically tailored for these purposes:

$$\frac{\Theta; \Gamma; \Delta \vdash M^A \mid s \quad \Theta; \Gamma; \Delta \vdash N^A \mid t}{\Theta; \Gamma; \Delta \vdash M^A \mid s + t}\,\text{T-PlusL}' \qquad \frac{\Theta; \Gamma; \Delta \vdash M^A \mid s \quad \Theta; \Gamma; \Delta \vdash N^A \mid t}{\Theta; \Gamma; \Delta \vdash N^A \mid s + t}\,\text{T-PlusR}'$$

This variant is meant to use for programming rather than as a stand-alone logic, since the above restriction to the "+" rules loses information about the derivation – the term no longer encodes a unique derivation modulo witness equivalence – and is not expressive enough to capture all LP-theorems (or even axioms like $[\![s]\!]A \supset [\![s+t]\!]A$). However, it can be useful when writing certified code which makes use of the "+" operator to extend its certificates. The use of this restriction ensures that both certificates (proof witnesses) are well-formed, and also that both branches of the program will indeed have the same type. Additionally, removing the extra witness from the term results in simpler code (it also eliminates the need for permutative reduction, since the "$+_L$" and "$+_R$" operators no longer appear in the terms). For example, the power function defined above could be rewritten as:

$$\begin{array}{rcl} \texttt{pow} & \triangleq & \texttt{fix}\ \lambda p^{\mathbb{N}\supset[\![s]\!](\mathbb{N}\supset\mathbb{N})}.\lambda n^{\mathbb{N}}. \\ & & \texttt{case}\ n\ \texttt{of}\ \texttt{0}\colon (!\lambda x^{\mathbb{N}}.\underline{1}), \texttt{succ}\ m^{\mathbb{N}}\colon (!\lambda y^{\mathbb{N}}.y*(vy))\langle v^{\mathbb{N}\supset\mathbb{N}}{:=}\underline{s}, pm\rangle \end{array}$$

where $s = \mu s'.\lambda x^{\mathbb{N}}.\underline{1} + \lambda y^{\mathbb{N}}.y*(s'\cdot y)$. Note that the witnesses $r = \lambda x^{\mathbb{N}}.\underline{1}$ and $t = \lambda y^{\mathbb{N}}.y*(s\cdot y)$ are no longer written on the complementary branches of the `case` construction.

The following variant is less restrictive than the above, and may be used in a setting where all witnesses involved in a derivation are required to be inhabited and all their variables accounted for, without imposing further restrictions on the corresponding formulas. In this case, we can modify the "+" rules as follows:

$$\frac{\Theta;\Gamma;\Delta\vdash A|\,s\quad \Theta;\Gamma;\Delta\vdash B|\,t}{\Theta;\Gamma;\Delta\vdash A\,|\,s+t}\mathsf{PlusL}^{-}\qquad \frac{\Theta;\Gamma;\Delta\vdash A|\,s\quad \Theta;\Gamma;\Delta\vdash B|\,t}{\Theta;\Gamma;\Delta\vdash B\,|\,s+t}\mathsf{PlusR}^{-}$$

We shall call this variant HLP^{-}. Again, it does not capture LP in its entirety (the LP-axioms $[\![s]\!]A \supset [\![s+t]\!]A$ and $[\![t]\!]A \supset [\![s]\!]B \supset [\![s+t]\!]A$ are not valid in HLP^{-} for arbitrary s and t), but it does capture a variant of LP which shall be referred to as LP^{-}. LP^{-} is a variant of LP where the axioms **A4** and **A5** are replaced by the following axioms:

A4. $[\![s]\!]A \supset [\![t]\!]B \supset [\![s+t]\!]A$

A5. $[\![s]\!]A \supset [\![t]\!]B \supset [\![s+t]\!]B$

The translation from LP^{-} to HLP^{-} is the same as the translation from LP to HLP, except for the constants corresponding to axioms **A4** and **A5**:

$$\underline{c^{\mathbf{A4}}_{s,t,A,B}} = \underline{c^{\mathbf{A5}}_{s,t,A,B}} \quad\triangleq\quad \lambda x^{\underline{[\![s]\!]A}}.\lambda y^{\underline{[\![t]\!]B}}.!(v^{\underline{A}}+w^{\underline{B}})\langle v^{\underline{A}}{:=}\underline{s}, x^{\underline{[\![s]\!]A}}\rangle\langle w^{\underline{B}}{:=}\underline{t}, y^{\underline{[\![t]\!]B}}\rangle$$

The translation from HLP^{-} to LP^{-} is the same as the translation from HLP to LP.

Proposition 4.7.2 If $\Gamma\vdash D$ is derivable in LP^{-}, then $\cdot;\underline{\Gamma};\cdot\vdash \underline{D}\,|\,s$ is derivable in HLP^{-} for some proof witness s.

Proof.- For axioms **A0**-**A3**, as well as for the inference rules, the proof is the same as in the translation from LP to HLP. The only differences lie in axioms **A4** and **A5**.

A4. $[\![s]\!]A \supset [\![t]\!]B \supset [\![s+t]\!]A$. Let $\Theta_1 = v^A, w^B$, $\Gamma_1 = x^{[\![s]\!]A}, y^{[\![t]\!]B}$

Part 1:

$$\dfrac{\dfrac{}{w^B;\Gamma_1;\cdot \vdash [\![s]\!]A \,|\, x^{[\![s]\!]A}}\,\mathsf{Var} \qquad \dfrac{\dfrac{\dfrac{}{\Theta_1;\cdot;\cdot \vdash A \,|\, v^A}\,\mathsf{VarM} \quad \dfrac{}{\Theta_1;\cdot;\cdot \vdash B \,|\, w^B}\,\mathsf{VarM}}{\Theta_1;\cdot;\cdot \vdash A \,|\, v^A + w^B}\,\mathsf{PlusL}^-}{\Theta_1;\Gamma_1;\cdot \vdash [\![v^A + w^B]\!]A \,|\, !(v^A + w^B)}\,\Box\mathsf{I}}{w^B;\Gamma_1;\cdot \vdash [\![s+w^B]\!]A \,|\, !(v^A+w^B)\langle v^A{:=}s, x^{[\![s]\!]A}\rangle}\,\Box\mathsf{E}$$

Part 2:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\cdot;\Gamma_1;\cdot \vdash [\![t]\!]B \,|\, y^{[\![t]\!]B}}\,\mathsf{Var} \quad \dfrac{\text{Part 1}}{w^B;\Gamma_1;\cdot \vdash [\![s+w^B]\!]A \,|\, !(v^A+w^B)\langle v^A{:=}s, x^{[\![s]\!]A}\rangle}}{\cdot;\Gamma_1;\cdot \vdash [\![s+t]\!]A \,|\, !(v^A+w^B)\langle v^A{:=}s, x^{[\![s]\!]A}\rangle\langle w^B{:=}t, y^{[\![t]\!]B}\rangle}\,\Box\mathsf{E}}{\cdot;x^{[\![s]\!]A};\cdot \vdash [\![t]\!]B \supset [\![s+t]\!]A \,|\, \lambda y^{[\![t]\!]B}.!(v^A+w^B)\langle v^A{:=}s, x^{[\![s]\!]A}\rangle\langle w^B{:=}t, y^{[\![t]\!]B}\rangle}\,\supset\mathsf{I}}{\cdot;\cdot;\cdot \vdash [\![s]\!]A \supset [\![t]\!]B \supset [\![s+t]\!]A \,|\, \lambda x^{[\![s]\!]A}.\lambda y^{[\![t]\!]B}.!(v^A+w^B)\langle v^A{:=}s, x^{[\![s]\!]A}\rangle\langle w^B{:=}t, y^{[\![t]\!]B}\rangle}\,\supset\mathsf{I}$$

A5. $[\![s]\!]A \supset [\![t]\!]B \supset [\![s+t]\!]B$. The proof is analogous to the previous one, except that PlusR^- is used instead of PlusL^-.

□

Proposition 4.7.3 If $\Theta;\Gamma;\Delta \vdash D \,|\, s$ is derivable in HLP^-, the LP^--judgement $\Theta^\star \cup \Gamma^\star \cup \Delta^\star \vdash [\![s^\star]\!]D^\star$ is derivable in LP.

Proof.- The proof is analogous to the proof of Proposition 4.3.6, except for the following cases:

- Case PlusL: the derivation ends in

$$\dfrac{\Theta;\Gamma;\Delta \vdash A \,|\, s \quad \Theta;\Gamma;\Delta \vdash B \,|\, t}{\Theta;\Gamma;\Delta \vdash A \,|\, s+t}\,\mathsf{PlusL}^-$$

  By the IH, we know that both $\Theta^\star \cup \Gamma^\star \cup \Delta^\star \vdash [\![s^\star]\!]A^\star$ and $\Theta^\star \cup \Gamma^\star \cup \Delta^\star \vdash [\![t^\star]\!]B^\star$ are derivable in LP^-. Thus, by **A4** and **R1** (twice), $\Theta^\star \cup \Gamma^\star \cup \Delta^\star \vdash [\![s^\star + t^\star]\!]A^\star$ is also derivable.

- Case PlusR^-: this is analogous to the previous case, using **A5** instead of **A4**.

□

Chapter 5

First-Order Hypothetical Logic of Proofs

This chapter extends our proof theoretical analysis of HLP to the first-order case. The first task being formulating appropriate Natural Deduction inference scheme capturing FOLP. This is to be followed by an analysis of normalisation of derivations and a proof of strong normalisation of such a process. We begin with a precise definition of FOLP in Section 5.1. We formulate our presentation of hypothetical First-Order Logic of Proofs in Section 5.2 and study its correspondence with FOLP in that same section. The Curry-Howard isomorphism is put forward in Section 5.3, including a term assignment, normalisation of derivations and strong normalisation of normalisation.

5.1 First-Order Logic of Proofs

This section recalls from [AY11] the FOLP, our presentation including two minor differences. The first is that we assume given functional letters in $\mathcal{L}$. The reason is that not much effort is required to incorporate them in our Natural Deduction development of the next section. The second is that we assume all proof variables to be decorated with formulas. That is, variables take the form x^A rather than x, just as we did in Chapter 4.

Definition 5.1.1 (Language of FOLP) Given $\mathcal{L}$, the language of FOLP, $\mathcal{L}_{\mathsf{LP}}$, is the extension of $\mathcal{L}$ that includes:

- Proof variables $x_1, x_2, \ldots$
- Proof constants $c_1, c_2, \ldots$
- Functional symbols for operations on proofs:
  - Those from LP: $+$ (binary), $\cdot$ (binary), $!$ (unary).
  - Unary $\mathsf{gen}_i()$ for each individual variable i.

The proof polynomials of propositional LP now take the name of "proof terms" [AY11].

Definition 5.1.2 (Proof terms and formulas) The set of proof terms, first-order expressions and formulas of FOLP is defined as follows:

$$\begin{array}{rcl} s,t & ::= & x^A \mid c \mid s \cdot t \mid !s \mid s + t \mid \mathsf{gen}_i(s) \\ E & ::= & i \mid f(E_1, \ldots, E_n) \\ A, B & ::= & P(E_1, \ldots, E_n) \mid \bot \mid A \supset B \mid [\![s]\!]_\Xi A \mid \forall i.A \end{array}$$

where Ξ is a set of individual variables. We often abbreviate $[\![s]\!]_\emptyset A$ with $[\![s]\!]A$.

Free individual variables in a first order term and formula of FOHLP is defined next. A proof term has a free individual variable i only if it occurs in the formula that decorates a proof variable and does not occur in an expression of the form $\mathsf{gen}_i(s)$. As stated earlier, the individual variables which are free in $[\![t]\!]_\Xi A$ are exactly those contained in Ξ. All other individual variables are assumed to be bound.

Definition 5.1.3 Free individual variables in a first-order expression and formula are defined as follows:

$$\begin{array}{rcl} \mathsf{FIV}(i) & \triangleq & \{i\} \\ \mathsf{FIV}(f(E_1, \ldots, E_n)) & \triangleq & \bigcup_{i \in 1..n} \mathsf{FIV}(E_i) \end{array}$$

$$\begin{array}{rcl} \mathsf{FIV}(P(E_1, \ldots, E_n)) & \triangleq & \bigcup_{i \in 1..n} \mathsf{FIV}(E_i) \\ \mathsf{FIV}(\bot) & \triangleq & \emptyset \\ \mathsf{FIV}(A \supset B) & \triangleq & \mathsf{FIV}(A) \cup \mathsf{FIV}(B) \\ \mathsf{FIV}([\![t]\!]_\Xi A) & \triangleq & \Xi \\ \mathsf{FIV}(\forall i.A) & \triangleq & \mathsf{FIV}(A) \backslash \{i\} \end{array}$$

For instance, in the formula $[\![(c^{\mathbf{A1a}}_{P(i),Q(j)})]\!]_{\{j\}}(P(i) \supset Q(j) \supset P(i))$, the variable j is free, while i is bound. We assume the following variable convention: the names of all bound individual variables are different each other, as well as from those of all free individual variables.

Remark 5.1.4 For every formula A, proof variable x and proof term s, $\mathsf{FIV}(A\{x \leftarrow s\}) = \mathsf{FIV}(A)$.

Definition 5.1.5 The axioms and inference rules of FOLP are the following (axioms **A1** and **B3** are separated into sub-axioms, one for each clause, in order to facilitate their treatment):

A1.	Axioms of first-order logic in the language of FOLP	
A2.	$([\![t]\!]_{\Xi,i}A) \supset [\![t]\!]_{\Xi}A$,	if $i \notin \mathsf{FIV}(A)$
A3.	$([\![t]\!]_{\Xi}A) \supset [\![t]\!]_{\Xi,i}A$	
B1.	$([\![t]\!]_{\Xi}A) \supset A$	
B2.	$([\![s]\!]_{\Xi}(A \supset B)) \supset ([\![t]\!]_{\Xi}A) \supset [\![(s \cdot t)]\!]_{\Xi}B$	
B3a.	$([\![s]\!]_{\Xi}A) \supset [\![(s + t)]\!]_{\Xi}A$	
B3b.	$([\![t]\!]_{\Xi}A) \supset [\![(s + t)]\!]_{\Xi}A$	
B4.	$([\![t]\!]_{\Xi}A) \supset [\![!t]\!]_{\Xi}[\![t]\!]_{\Xi}A$	
B5.	$([\![t]\!]_{\Xi}A) \supset [\![\mathsf{gen}_i(t)]\!]_{\Xi}\forall i.A$,	if $i \notin \Xi$
MP.	$\vdash A \supset B$ and $\vdash A \;\Rightarrow \vdash B$	
Gen.	$\vdash A \;\Rightarrow \vdash \forall i.A$	
Nec.	A an axiom $\;\Rightarrow \vdash [\![c]\!]A$	

The axioms of first-order logic we shall work with are:

A1a.	$A \supset B \supset A$	
A1b.	$(A \supset B \supset C) \supset (A \supset B) \supset A \supset C$	
A1c.	$\neg\neg A \supset A$	
A1d.	$(\forall i.A) \supset A\{i \leftarrow E\}$	
A1e.	$(\forall i.(A \supset B)) \supset (\forall i.A) \supset \forall i.B$	
A1f.	$A \supset \forall i.A$,	if $i \notin \mathsf{FIV}(A)$

Remark 5.1.6 While all LP axioms are still axioms in FOLP, their names differ in both presentations. LP axioms **A0**, **A1**, **A2**, **A2**, **A4**, **A5** have their first-order counterparts in axioms **A1a,b,c**, **B1**, **B2**, **B4**, **B3a** and **B3b** respectively. Additionally, inference rules **MP** and **Nec** in LP correspond to rules **MP** and **Nec** in FOLP.

5.1.1 Metatheoretical results

We now develop some metatheoretical results that we shall use in subsequent sections.

Definition 5.1.7 Let $\Gamma = A_1, \ldots A_n$, a context of the form $[\![u_1]\!]_{\Xi_1}A_1, \ldots, [\![u_n]\!]_{\Xi_n}A_n$ is referred to as $[\![\vec{u}]\!]_{\vec{\Xi}}\Gamma$.

Lemma 5.1.8 (Internalization) If $[\![\vec{u}]\!]_{\vec{\Xi}}\Gamma \vdash D$ is derivable in FOLP, then there exists a proof term r such that $[\![\vec{u}]\!]_{\vec{\Xi}}\Gamma \vdash [\![r]\!]_{\vec{\Xi}}D$ is derivable in FOLP.

Proof.- By induction on the derivation of $[\![\vec{u}]\!]_{\vec{\Xi}}\Gamma \vdash D$.

Note that below we make use (several times) of th fact that, for every $\Xi' \subseteq \vec{\Xi}$, $[\![r]\!]_{\vec{\Xi}}D$ can be obtained from $[\![r]\!]_{\Xi'}D$ by using **A3** and **MP** as many times as required.

- If there is some hypothesis $x^{[\![u]\!]_{\Xi'}A} \in [\![\vec{u}]\!]_{\vec{\Xi}}\Gamma$ and the derivation is obtained by using this hypothesis, then $D = [\![u]\!]_{\Xi'}A$ and $r = !u$ (note that $[\![\vec{u}]\!]_{\vec{\Xi}}\Gamma$ can only contain hypotheses of the form $x^{[\![u]\!]_{\Xi'}A}$ for some u and A, and that $\Xi' \subseteq \vec{\Xi}$).

- If the derivation is an instance of an axiom **Ai** with $i \in \{1a, \cdots, 3\}$, then $r = c^{\mathbf{Ai}}_{\cdots}$ (with the corresponding arguments for D).
- If the derivation is an instance of an axiom **Bi** with $i \in \{1, \cdots, 5\}$, then $r = c^{\mathbf{Bi}}_{\cdots}$ (with the corresponding arguments for D).
- If the derivation is obtained by using **MP** from $[\![\vec{u}]\!]_{\vec{\Xi}}\Gamma \vdash A \supset D$ and $[\![\vec{u}]\!]_{\vec{\Xi}}\Gamma \vdash A$, then by IH there exist s and t s.t. $[\![\vec{u}]\!]_{\vec{\Xi}}\Gamma \vdash [\![s]\!]_{\vec{\Xi}}(A \supset D)$ and $[\![\vec{u}]\!]_{\vec{\Xi}}\Gamma \vdash [\![t]\!]_{\vec{\Xi}}A$ are both derivable. Take $r = s \cdot t$.
- If the derivation is obtained by using **Gen** from $[\![\vec{u}]\!]_{\vec{\Xi}}\Gamma \vdash A$, then D is of the form $\forall i.A$ with $i \notin \mathsf{FIV}([\![\vec{u}]\!]_{\vec{\Xi}}\Gamma)$. By IH, there exists s s.t. $[\![\vec{u}]\!]_{\vec{\Xi}}\Gamma \vdash [\![s]\!]_{\vec{\Xi}}A$ is derivable. Take $r = \mathsf{gen}_i(s)$.
- If the derivation is obtained by using **Nec**, then D is of the form $[\![c]\!]A$ and $r = !c$.

□

Corollary 5.1.9 If $[\![\vec{u}]\!]_{\vec{\Xi}}\Gamma \vdash D$ is derivable in FOLP, then there exists a proof term $r^{\vec{\Xi},D}$ such that $[\![\vec{u}]\!]_{\vec{\Xi}}\Gamma \vdash [\![r^{\vec{\Xi},D}]\!]_{\vec{\Xi}\cap\mathsf{FIV}(D)}D$ is derivable in FOLP.

Proof.- Since $[\![\vec{u}]\!]_{\vec{\Xi}}\Gamma \vdash [\![r]\!]_{\vec{\Xi}}D$ is derivable for some r by Internalization, we can obtain $[\![\vec{u}]\!]_{\vec{\Xi}}\Gamma \vdash [\![r]\!]_{\vec{\Xi}\cap\mathsf{FIV}(D)}D$ by using **A2** and **MP** as many times as necessary. Take $r^{\vec{\Xi},D} = r$. □

Lemma 5.1.10 (Stripping) Suppose π is a FOLP-derivation of $\Gamma, x^{[\![y]\!]_{\Xi}A} \vdash B$, $y^A \notin \Gamma$ and, for every subformula of B of the form $[\![t]\!]_{\Xi'}C$ where t contains y^A, $\mathsf{FIV}(C) \subseteq \Xi'$. Then there is a derivation π' of $\Gamma, y^A \vdash B'$, where B' results from B, by replacing all occurrences of $[\![t]\!]_{\Xi'}C$ by C for every set of individual variables Ξ', every formula C, and every proof term t containing y^A (including constants for instances of axioms containing y^A).

Proof.- By induction on π.

- If $B = [\![y^A]\!]_{\Xi}A$ and π is obtained by using the hypothesis $x^{[\![y^A]\!]_{\Xi}A}$, then $B' = A$ and π' is the derivation of $\Gamma, y^A \vdash A$ obtained by using the hypothesis y^A.
- If π is obtained by using a hypothesis $z^B \in \Gamma$, then there is a derivation of $\Gamma \vdash B$ which uses neither $x^{[\![y^A]\!]A}$ nor y^A. We obtain π' from this derivation by Weakening, and $B' = B$.
- If π is obtained by using an axiom, there are three possibilities:
  - B has no proof term containing y^A: then $B' = B$, and the derivation of $\Gamma \vdash B$ can be obtained by Weakening of the axiom.
  - B has one or more proof terms containing y^A, but B' is still an instance of the same axiom. Since axioms can be derived in any context, then $\Gamma, y^A \vdash B'$ is derivable.
  - B has at least one proof term containing y^A in a way that B' is no longer an instance of the same axiom as B: in this case, B' is an instance of one of the following schemes:
    1. $C \supset C$ from axioms **A2**, **A3**, **B1**, **B3a**, **B3b**, **B4**.
    2. $[\![s]\!]_{\Xi}C \supset C$ from axioms **B3a**, **B3b**.
    3. $(A' \supset C') \supset ([\![t']\!]_{\Xi}C' \supset C')$, 4. $[\![s]\!]_{\Xi}(A' \supset C') \supset (A' \supset C')$ or 5. $(A' \supset C') \supset (A' \supset C')$ from axiom **B2**.

6. $C \supset \forall i.C$ from **B5**.

Cases 1-5 can be derived in FOLP in any context. Case 6 is obtained by **A1f** if $i \notin \mathsf{FIV}(C)$. This restriction holds, since $\mathsf{FIV}(C) \subseteq \Xi'$ and $i \notin \Xi'$.

- If π is obtained by applying **MP**:

$$\dfrac{\dfrac{\cdots}{\Gamma, x^{[\![y^A]\!]_{\Xi} A} \vdash D \supset B} \qquad \dfrac{\cdots}{\Gamma, x^{[\![y^A]\!]_{\Xi} A} \vdash D}}{\Gamma, x^{[\![y^A]\!]_{\Xi} A} \vdash B}\, MP$$

  By induction hypothesis, we have derivations of $\Gamma, y^A \vdash D' \supset B'$ and $\Gamma, y^A \vdash D'$. Therefore, by **MP**, we obtain a derivation of $\Gamma, y^A \vdash B'$.

- If π is obtained by applying **Gen**, then B is of the form $\forall i.D$ with $i \notin \mathsf{FIV}(\Gamma, x^{[\![y^A]\!]_{\Xi} A})$, and there is a derivation of $\Gamma, x^{[\![y^A]\!]_{\Xi} A} \vdash D$. By induction hypothesis we can derive $\Gamma, y^A \vdash D'$. And, since $\mathsf{FIV}(A) \subseteq \Xi$, then $i \notin \mathsf{FIV}(\Gamma, y^A)$. The result is obtained by **Gen**.

- If π is obtained by applying **Nec**, then B is of the form $[\![c]\!]D$ with c a proof constant and D an instance of an axiom. If $y^A \in D$, then $B' = D'$ and we resort to the third item of this proof. Otherwise, $B' = B$.

□

Lemma 5.1.11 (λ-Abstraction) If $\rhd_{\mathsf{FOLP}} [\![\vec{u}]\!]_{\vec{\Xi}} \Gamma, y^{[\![x^A]\!]_{\mathsf{FIV}(A)} A} \vdash [\![s(\vec{u}, x^A)]\!]_{\mathsf{FIV}(B)} B$ with $x^A \notin \Gamma, B$, then there exists a proof term $t_{\lambda}^{A \supset B}$ such that $\rhd_{\mathsf{FOLP}} [\![\vec{u}]\!]_{\vec{\Xi}} \Gamma \vdash [\![t_{\lambda}^{A \supset B}]\!]_{\vec{\Xi} \cap \mathsf{FIV}(A \supset B)} (A \supset B)$.

Proof.- W.l.o.g. we may assume that $x^A \in s(\vec{u}, x^A)$. Indeed, if this were not the case, then we could add it as follows:

$$\begin{array}{ll}
(a) & [\![\vec{u}]\!]_{\vec{\Xi}} \Gamma, y^{[\![x^A]\!]_{\mathsf{FIV}(A)} A} \vdash [\![c_{B,A}^{\mathbf{A1a}}]\!] (B \supset A \supset B) \\
(b) & [\![\vec{u}]\!]_{\vec{\Xi}} \Gamma, y^{[\![x^A]\!]_{\mathsf{FIV}(A)} A} \vdash [\![c_{B,A}^{\mathbf{A1a}}]\!]_{\mathsf{FIV}(A) \cup \mathsf{FIV}(B)} (B \supset A \supset B) \\
(c) & [\![\vec{u}]\!]_{\vec{\Xi}} \Gamma, y^{[\![x^A]\!]_{\mathsf{FIV}(A)} A} \vdash [\![s(\vec{u}, x^A)]\!]_{\mathsf{FIV}(B)} B \\
(d) & [\![\vec{u}]\!]_{\vec{\Xi}} \Gamma, y^{[\![x^A]\!]_{\mathsf{FIV}(A)} A} \vdash [\![s(\vec{u}, x^A)]\!]_{\mathsf{FIV}(A) \cup \mathsf{FIV}(B)} B \\
(e) & [\![\vec{u}]\!]_{\vec{\Xi}} \Gamma, y^{[\![x^A]\!]_{\mathsf{FIV}(A)} A} \vdash [\![c_{B,A}^{\mathbf{A1a}} \cdot s(\vec{u}, x^A)]\!]_{\mathsf{FIV}(A) \cup \mathsf{FIV}(B)} (A \supset B) \\
(f) & [\![\vec{u}]\!]_{\vec{\Xi}} \Gamma, y^{[\![x^A]\!]_{\mathsf{FIV}(A)} A} \vdash [\![x^A]\!]_{\mathsf{FIV}(A)} A \\
(g) & [\![\vec{u}]\!]_{\vec{\Xi}} \Gamma, y^{[\![x^A]\!]_{\mathsf{FIV}(A)} A} \vdash [\![x^A]\!]_{\mathsf{FIV}(A) \cup \mathsf{FIV}(B)} A \\
(h) & [\![\vec{u}]\!]_{\vec{\Xi}} \Gamma, y^{[\![x^A]\!]_{\mathsf{FIV}(A)} A} \vdash [\![c_{B,A}^{\mathbf{A1a}} \cdot s(\vec{u}, x^A) \cdot x^A]\!]_{\mathsf{FIV}(A) \cup \mathsf{FIV}(B)} B
\end{array}$$

We reason as follows:

$[\![\vec{u}]\!]_{\vec{\Xi}} \Gamma, y^{[\![x^A]\!]_{\mathsf{FIV}(A)} A} \vdash [\![s(\vec{u}, x^A)]\!]_{\mathsf{FIV}(B)} B$	(d)
$[\![\vec{u}]\!]_{\vec{\Xi}} \Gamma, x^A \vdash B$	(Stripping, $x \in s(\vec{u}, x^A)$, $\mathsf{FIV}(B) \subseteq \mathsf{FIV}(A) \cup \mathsf{FIV}(B)$)
$[\![\vec{u}]\!]_{\vec{\Xi}} \Gamma \vdash A \supset B$	(Deduction for LP)
$[\![\vec{u}]\!]_{\vec{\Xi}} \Gamma \vdash [\![r^{\vec{\Xi}, A \supset B}]\!]_{\vec{\Xi} \cap \mathsf{FIV}(A \supset B)} (A \supset B)$	(Cor. 5.1.9)

Take $t_{\lambda}^{A \supset B} = r^{\vec{\Xi}, A \supset B}$. □

Corollary 5.1.12 (μ-Abstraction) Suppose $[\![\vec{u}]\!]_{\vec{\Xi}}\Gamma, y^{[\![\alpha^{\neg A}]\!]_{\mathsf{FIV}(A)}\neg A} \vdash [\![s(\vec{u}, \alpha^{\neg A})]\!]\bot$ and $\alpha^{\neg A} \notin \Gamma$. Then $[\![\vec{u}]\!]_{\vec{\Xi}}\Gamma \vdash [\![t^A_\mu]\!]_{\vec{\Xi}\cap\mathsf{FIV}(A)}A$, where $t^A_\mu \triangleq c^{\mathbf{A1c}}_A \cdot t^{\neg\neg A}_\lambda$.

Proof.- We reason as follows:

$[\![\vec{u}]\!]_{\vec{\Xi}}\Gamma \vdash [\![t^{\neg\neg A}_\lambda([\![\vec{u}]\!]_{\vec{\Xi}}\Gamma)]\!]_{\vec{\Xi}\cap\mathsf{FIV}(A)}(\neg\neg A)$	(λ-Abstraction)
$[\![\vec{u}]\!]_{\vec{\Xi}}\Gamma \vdash [\![c^{\mathbf{A1}}_A]\!](\neg\neg A \supset A)$	(**A1** and **Nec**)
$[\![\vec{u}]\!]_{\vec{\Xi}}\Gamma \vdash [\![c^{\mathbf{A1}}_A]\!]_{\vec{\Xi}\cap\mathsf{FIV}(A)}(\neg\neg A \supset A)$	(**A3** and **MP** as many times as needed)
$[\![\vec{u}]\!]_{\vec{\Xi}}\Gamma \vdash [\![c^{\mathbf{A1}}_A \cdot t^{\neg\neg A}_\lambda([\![\vec{u}]\!]_{\vec{\Xi}}\Gamma)]\!]_{\vec{\Xi}\cap\mathsf{FIV}(A)}A$	(**B2**, and **MP** twice)

□

Lemma 5.1.13 (!-Abstraction) Suppose $\rhd_{\mathsf{FOLP}} [\![\vec{u}]\!]_{\vec{\Xi}}\Gamma \vdash [\![s]\!]_{\vec{\Xi}\cap\mathsf{FIV}(A)}A$ and $\vec{\Xi} \cap \mathsf{FIV}(A) \subseteq \Xi$, for some Ξ. Then there exists a proof term $t^{\Xi,A}_!$ such that $\rhd_{\mathsf{FOLP}} [\![\vec{u}]\!]_{\vec{\Xi}}\Gamma \vdash [\![t^{\Xi,A}_!]\!]_{\vec{\Xi}\cap\Xi}[\![s]\!]_\Xi A$.

Proof.- We reason as follows:

$[\![\vec{u}]\!]_{\vec{\Xi}}\Gamma \vdash [\![s]\!]_{\vec{\Xi}\cap\mathsf{FIV}(A)}A$	(hypothesis)
$[\![\vec{u}]\!]_{\vec{\Xi}}\Gamma \vdash [\![s]\!]_\Xi A$	(**A3***, **MP***, $\vec{\Xi} \cap \mathsf{FIV}(A) \subseteq \Xi$)
$[\![\vec{u}]\!]_{\vec{\Xi}}\Gamma \vdash [\![r^{\vec{\Xi},[\![s]\!]_\Xi A}]\!]_{\vec{\Xi}\cap\Xi}[\![s]\!]_\Xi A$	(Cor. 5.1.9)

* As many times as required. Take $t^{\Xi,A}_! = r^{\vec{\Xi},[\![s]\!]_\Xi A}$.

Note: if $\Xi \subseteq \vec{\Xi}$, then we can take $t^{\Xi,A}_! = !s$ and the result holds by **B4** instead of Cor. 5.1.9. □

Lemma 5.1.14 If C is an instance of a FOLP-axiom, then $C\{x^A \leftarrow s\}$ is also an instance of the same axiom.

Proof.- By definition of the substitution and the axioms. Note that, for any subformula C of B, if $i \notin \mathsf{FIV}(C)$, then $i \notin \mathsf{FIV}(C\{x^A \leftarrow s\})$ by Remark 5.1.4. Also note that proof variable substitution does not affect the sets of individual variables used as subindices, since, by definition, $([\![t]\!]_{\Xi_1}C)\{x^A \leftarrow s\} = [\![t\{x^A \leftarrow s\}]\!]_{\Xi_1}C\{x^A \leftarrow s\}$ for any t, C and Ξ_1. □

Lemma 5.1.15 (Substitution) $\Gamma \vdash [\![s]\!]_\Xi A$, $\Gamma, y^{[\![x^A]\!]_{\Xi'}A} \vdash B$, $\Xi \cap \mathsf{FIV}(A) \subseteq \Xi'$ and $x^A \notin \Gamma$ implies $\Gamma \vdash B\{x^A \leftarrow s\}$.

Proof.- By induction on the derivation of $\Gamma, y^{[\![x^A]\!]_{\Xi'}A} \vdash B$.

- If there is some hypothesis $z^B \in \Gamma$, $z^B \neq y^{[\![x^A]\!]_{\Xi'}A}$, and the derivation is obtained by using this hypothesis, then, by variable convention, $x^A \notin \mathsf{FV}(B)$ and $y^{[\![x^A]\!]_{\Xi'}A} \notin \mathsf{FV}(B)$. Therefore $B\{x^A \leftarrow s\} = B$, and $\Gamma \vdash B$ is derivable by Strengthening.
- If $B = [\![x^A]\!]_{\Xi'}A$ and the derivation is obtained by using the hypothesis $y^{[\![x^A]\!]_{\Xi'}A}$, then $B\{x^A \leftarrow s\} = [\![s]\!]_{\Xi'}A$. Since $\Gamma \vdash [\![s]\!]_\Xi A$ is derivable by hypothesis and $\Xi \cap \mathsf{FIV}(A) \subseteq \Xi'$, we can derive $\Gamma \vdash [\![s]\!]_{\Xi'}A$ by using **A2** and **A3** as necessary.

- If the derivation is an instance of an axiom, then $B\{x^A \leftarrow s\}$ is also an instance of the same axiom by Lemma 5.1.14.
- If the derivation is obtained by using **MP** from $\Gamma \vdash C \supset B$ and $\Gamma \vdash C$, then by IH $\Gamma \vdash C\{x^A \leftarrow s\} \supset B\{x^A \leftarrow s\}$ and $\Gamma \vdash C\{x^A \leftarrow s\}$ are both derivable. We obtain the result by **MP**.
- If the derivation is obtained by using **Gen** from $\Gamma \vdash C$, then B is of the form $\forall i.C$ with $i \notin \mathsf{FIV}(\Gamma)$. By IH, we can derive $\Gamma \vdash C\{x^A \leftarrow s\}$, and then obtain the result by **Gen**.
- If the derivation is obtained by using **Nec**, then B is of the form $[\![c]\!]C$ with C an instance of an axiom. Since $C\{x^A \leftarrow s\}$ is also an instance of the same axiom (by Lemma 5.1.14), then $\Gamma \vdash [\![c]\!]C\{x^A \leftarrow s\}$ is derivable by **Nec**.

□

5.1.2 Kripke Semantics

Fitting [Fit14] extended his possible worlds semantics of LP to FOLP. In the propositional case, "$[\![t]\!]A$" was considered to be true at a possible world if A was true at all accessible worlds, and t served as meaningful evidence for A at that world. In FOLP, this notion becomes more complex because of the two roles individual variables can play in a formula. Remember that the formula like $[\![t]\!]_{\{i,j\}}P(i,j,k)$, the variables i and j can be affected by substitutions, while the variable k can be subject to universal generalization. In Fitting's interpretation, the concept of truth at a possible world Γ applies to instances of formulas like $[\![t]\!]_{\{i,j\}}P(a,b,k)$, with a and b in the domain of Γ, where $P(a,b,c)$ is true at every possible world Γ' accessible from Γ for every c in the quantificational domain of Γ'.

An **FOLP-skeleton** is a structure $\langle \mathcal{G}, \mathcal{R}, \mathcal{D} \rangle$ where $\mathcal{G}$ is a non-empty set of possible worlds, $\mathcal{R}$ is a binary reflexive and transitive accessibility relation on $\mathcal{G}$, and $\mathcal{D}$ is a domain function mapping each member of $\mathcal{G}$ to a non-empty set, and subject to a monotonicity condition: if $\Gamma\mathcal{R}\Gamma'$, then $\mathcal{D}(\Gamma) \subseteq \mathcal{D}(\Gamma')$. The **domain** of $\langle \mathcal{G}, \mathcal{R}, \mathcal{D} \rangle$, denoted as $\mathcal{D}^*$, is defined as $\{\mathcal{D}(\Gamma) \mid \Gamma \in \mathcal{G}\}$.

Given a non-empty set D, a D**-formula** is the result of replacing some (possibly all) free occurrences of individual variables in an FOLP-formula with members of D. The elements of D act as constants within that formula, and are referred to as **domain constants**.

An **FOLP-model** based on a skeleton $\langle \mathcal{G}, \mathcal{R}, \mathcal{D} \rangle$ is a structure $\langle \mathcal{G}, \mathcal{R}, \mathcal{D}, \mathcal{I}, \mathcal{E} \rangle$ where:

- $\mathcal{I}$ is an **interpretation function**: for each n-ary predicate P and each $\Gamma \in \mathcal{G}$, $\mathcal{I}(P, \Gamma)$ is an n-ary relation on $\mathcal{D}(\Gamma)$.
- $\mathcal{E}$ is an **evidence function**: for each proof term t and each $\mathcal{D}^*$-formula A, $\mathcal{E}(t, A)$ is some set of possible worlds such that if $\Gamma \in \mathcal{E}(t, A)$, then all domain constants in A are from $\mathcal{D}(\Gamma)$.

Additionally, every FOLP-model $\langle \mathcal{G}, \mathcal{R}, \mathcal{D}, \mathcal{I}, \mathcal{E} \rangle$ must meet the following conditions:

- $\cdot$: $\mathcal{E}(s, A \supset B) \cap \mathcal{E}(t, A) \subseteq \mathcal{E}(s \cdot t, B)$.
- $\mathcal{R}$ Closure: if $\Gamma \in \mathcal{E}(t, A)$ and $\Gamma\,\mathcal{R}\,\Gamma'$, then $\Gamma' \in \mathcal{E}(t, A)$.

- !: $\mathcal{E}(t, A) \subseteq \mathcal{E}(!t, [\![t]\!]_X A)$ where X is the set of domain constants in A.
- +: $\mathcal{E}(s, A) \cup \mathcal{E}(t, A) \subseteq \mathcal{E}(s + t, A)$.
- Instantiation: if $\Gamma \in \mathcal{E}(t, A)$ and $a \in \mathcal{D}(\Gamma)$, then $\Gamma \in \mathcal{E}(t, A\{i \leftarrow a\})$.
- gen_i: $\mathcal{E}(t, A) \subseteq \mathcal{E}(\mathsf{gen}_i(t), \forall i.A)$.

Let $\langle \mathcal{G}, \mathcal{R}, \mathcal{D}, \mathcal{I}, \mathcal{E} \rangle$ be an FOLP-model, and let A be a $\mathcal{D}^*$-formula, A is said to **live in** a world $\Gamma \in \mathcal{G}$ if all members of $\mathcal{D}^*$ that occur in A are in $\mathcal{D}(\Gamma)$. An evidence function **meets** a constant specification $\mathcal{CS}$ if, for every c s.t. $[\![c]\!]_\emptyset A \in \mathcal{CS}$ and each $\Gamma \in \mathcal{G}$ s.t. A lives in Γ, $\Gamma \in \mathcal{E}(c, A)$. A model meets a constant specification if its evidence function does.

The concept of **truth** at a world is defined as follows: given an FOLP-model $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, \mathcal{D} \rangle$, the closed (i.e. with no free individual variables) $\mathcal{D}^*$-formula A is *true* at world Γ in $\mathcal{M}$ (denoted as $\mathcal{M}, \Gamma \Vdash A$) if one of the following conditions is met:

- $A = P(a_1, \ldots, a_n)$ and $a_1, \ldots, a_n \in \mathcal{I}(\Gamma, P)$.
- $A = B \supset C$ and either $\mathcal{M}, \Gamma \nVdash B$ or $\mathcal{M}, \Gamma \Vdash C$ (and similarly for other propositional connectives).
- $A = \forall i.B$ and $\mathcal{M}, \Gamma \Vdash B\{i \leftarrow a\}$ for every $a \in \mathcal{D}(\Gamma)$.
- $A = [\![t]\!]_X B$ with X a set of individual constants, $\mathsf{FIV}(B) = \{i_1, \ldots, i_n\}$, $\Gamma \in \mathcal{E}(t, A)$ and $\mathcal{M}, \Gamma' \Vdash B\{i_1 \leftarrow a_1, \ldots, i_n \leftarrow a_n\}$ for every Γ' s.t. $\Gamma \mathcal{R} \Gamma'$ and all $a_1, \ldots, a_n \in \mathcal{D}(\Gamma')$.

As usual, a closed FOLP-formula is considered **valid** in a model $\langle \mathcal{G}, \mathcal{R}, \mathcal{D}, \mathcal{I}, \mathcal{E} \rangle$ if it is true at every possible world $\Gamma \in \mathcal{G}$. An FOLP-formula with free individual variables is valid if its universal closure is.

Fitting proved soundness and completeness by showing that each of the FOLP-axioms is valid in all FOLP-models, and the rules preserve validity, hence every theorem is valid.

5.2 First-Order Hypothetical Logic of Proofs

Definition 5.2.1 (Proof witnesses and formulas) The set of first order expressions, formulas and proof witnesses of FOHLP is defined by the following syntax:

$$\begin{array}{rcl} r, s, t & ::= & x^A \mid v^A_\Xi \mid \lambda x^A.s \mid s \cdot t \\ & \mid & !s \mid t\langle v^A_{\Xi'} := r, s\rangle \\ & \mid & [\alpha^A]s \mid \mu\alpha^A.s \\ & \mid & s + t \\ & \mid & \mathsf{gen}_i(s) \mid \mathsf{ins}^E_i(s) \\ E_1, \ldots, E_n & ::= & i \mid f(E_1, \ldots, E_n) \\ A, B & ::= & P(E_1, \ldots, E_n) \mid \bot \mid A \supset B \mid [\![s]\!]_\Xi A \mid \forall i.A \end{array}$$

Additional proof witnesses are required in order to deal with $\forall$ introduction and elimination. These new witnesses take the forms $\mathsf{gen}_i(s)$ (*generalization*) and $\mathsf{ins}^E_i(s)$ (*instantiation*).

Definition 5.2.2 (Contexts and judgements for FOHLP) As in HLP, we introduce three types of **contexts** for defining judgements. Contexts of truth hypothesis Γ and negated truth hypothesis Δ are as before (Def. 4.2.2). Contexts of validity assumptions are also as before except that now validity variables are those of Def. 5.2.1: $(v_1)^{A_1}_{\Xi_1}, \ldots, (v_n)^{A_n}_{\Xi_n}$. Judgements are also as for HLP.

Definition 5.2.3 (Free individual variables of formulas, proof witnesses and contexts) The set of free individual variables of a formula is A, denoted $\mathsf{FIV}(A)$, is defined as:

$$\begin{array}{rcl}
\mathsf{FIV}(i) & \triangleq & \{i\} \\
\mathsf{FIV}(P(E_1, \ldots, E_n)) & \triangleq & \mathsf{FIV}(E_1) \cup \ldots \cup \mathsf{FIV}(E_n) \\
\mathsf{FIV}(\bot) & \triangleq & \emptyset \\
\mathsf{FIV}(A \supset B) & \triangleq & \mathsf{FIV}(A) \cup \mathsf{FIV}(B) \\
\mathsf{FIV}([\![t]\!]_{\Xi} A) & \triangleq & \Xi \\
\mathsf{FIV}(\forall i.A) & \triangleq & \mathsf{FIV}(A) \backslash \{i\}
\end{array}$$

The set of free individual variables of a proof witness s, denoted $\mathsf{FIV}(s)$, is defined as follows:

$$\begin{array}{rcrcl}
 & & \mathsf{FIV}(x^A) & \triangleq & \mathsf{FIV}(A) \\
 & & \mathsf{FIV}(v^A_{\Xi}) & \triangleq & \Xi \\
\mathsf{FIV}(s \cdot t) & \triangleq & \mathsf{FIV}(s + t) & \triangleq & \mathsf{FIV}(s) \cup \mathsf{FIV}(t) \\
 & & \mathsf{FIV}(!s) & \triangleq & \mathsf{FIV}(s) \\
 & & \mathsf{FIV}(t\langle v^A_{\Xi} := r, s\rangle) & \triangleq & \mathsf{FIV}(t) \cup \mathsf{FIV}(r) \cup \mathsf{FIV}(s) \cup \Xi \\
\mathsf{FIV}(\lambda x^A.s) & \triangleq & \mathsf{FIV}([\alpha^A]s) & \triangleq & \mathsf{FIV}(\mu\alpha^A.s) \quad \triangleq \quad \mathsf{FIV}(A) \cup \mathsf{FIV}(s) \\
 & & \mathsf{FIV}(\mathsf{gen}_i(s)) & \triangleq & \mathsf{FIV}(s) \backslash \{i\} \\
 & & \mathsf{FIV}(\mathsf{ins}^E_i(s)) & \triangleq & \mathsf{FIV}(s) \backslash \{i\} \cup \mathsf{FIV}(E)
\end{array}$$

For contexts we have:

$$\begin{array}{rcl}
\mathsf{FIV}(\Theta) & \triangleq & \bigcup\limits_{v^A_{\Xi} \in \Theta} \Xi \\
\mathsf{FIV}(\Gamma) & \triangleq & \{\mathsf{FIV}(A) \mid x^A \in \Gamma\} \\
\mathsf{FIV}(\Delta) & \triangleq & \{\mathsf{FIV}(A) \mid \alpha^A \in \Delta\}
\end{array}$$

Note the clause defining $\mathsf{FIV}(v^A_{\Xi})$ in Def. 5.2.3. It denotes Ξ reflecting that all free individual variables that are not in Ξ are considered bound whereas those that are in Ξ are considered free (disregarding whether they occur in A or not). Also, all free individual variables in the type decoration A in x^A are free in x^A since substitution may affect them.

We will work modulo α-equivalence over individual variables, renaming where appropriate so that the following notational convention is upheld: the names of the free individual are assumed distinct and also different from the names of the free individual variables, in any proof witness, formula, statement or proof. For example, we do not allow formulas of the form $[\![s]\!]_{\Xi} A$ where Ξ contains one or more individual variables which are bound in either s or A.

Definition 5.2.4 (α-equivalence for individual variables) α-equivalence for individual variables is defined as the least congruence $=_\alpha$ that includes:

$$\begin{array}{lll} \forall i.A & =_\alpha \forall j.A\{i \leftarrow j\}, & \text{if } j \notin \mathsf{FIV}(A) \\ [\![t]\!]_\Xi A & =_\alpha [\![t\{i \leftarrow j\}]\!]_\Xi A\{i \leftarrow j\}, & \text{if } i \notin \Xi \text{ and } j \text{ fresh.} \end{array}$$

Lemma 5.2.5 For every formula A, truth variable x^B and proof witness t, $\mathsf{FIV}(A\{x^B \leftarrow t\}) = \mathsf{FIV}(A)$.

Proof.- By induction on A, using the definition of free individual variables. □

With individual variable substitution defined as follows.

Definition 5.2.6 (Individual variable substitution) Substitution of individual variable i in a first-order expression E' by E, written $E'\{i \leftarrow E\}$, is defined as:

$$\begin{array}{rcll} i\{i \leftarrow E\} & \triangleq & E & \\ j\{i \leftarrow E\} & \triangleq & j, & \text{if } j \neq i \\ f(E_1, \ldots, E_n)\{i \leftarrow E\} & \triangleq & f(E_1\{i \leftarrow E\}, \ldots, E_n\{i \leftarrow E\}) & \end{array}$$

Substitution of individual variable i in a formula is defined as:

$$\begin{array}{rcll} P(E_1, \ldots, E_n)\{i \leftarrow E\} & \triangleq & P(E_1\{i \leftarrow E\}, \ldots, E_n\{i \leftarrow E\}) & \\ \bot\{i \leftarrow E\} & \triangleq & \bot & \\ (A \supset B)\{i \leftarrow E\} & \triangleq & A\{i \leftarrow E\} \supset B\{i \leftarrow E\} & \\ (\forall j.A)\{i \leftarrow E\} & \triangleq & \forall j.A\{i \leftarrow E\}, & \text{if } i \neq j \\ (\forall i.A)\{i \leftarrow E\} & \triangleq & \forall i.A & \\ ([\![s]\!]_\Xi A)\{i \leftarrow E\} & \triangleq & [\![s\{i \leftarrow E\}]\!]_{(\Xi \setminus \{i\}) \cup \mathsf{FIV}(E)} A\{i \leftarrow E\}, & \text{if } i \in \Xi \\ ([\![s]\!]_\Xi A)\{i \leftarrow E\} & \triangleq & [\![s]\!]_\Xi A, & \text{if } i \notin \Xi \end{array}$$

Finally, substitution of individual variable i in a proof witness is defined as:

$$\begin{array}{rcll} x^A\{i \leftarrow E\} & \triangleq & x^{A\{\cancel{i} \leftarrow E\}} & \\ v^A_\Xi\{i \leftarrow E\} & \triangleq & v^{A\{\cancel{i} \leftarrow E\}}_{(\Xi \setminus \{i\}) \cup \mathsf{FIV}(E)}, & \text{if } i \in \Xi \\ v^A_\Xi\{i \leftarrow E\} & \triangleq & v^A_\Xi, & \text{if } i \notin \Xi \\ (s \cdot t)\{i \leftarrow E\} & \triangleq & s\{i \leftarrow E\} \cdot t\{i \leftarrow E\} & \\ (s + t)\{i \leftarrow E\} & \triangleq & s\{i \leftarrow E\} + t\{i \leftarrow E\} & \\ (!s)\{i \leftarrow E\} & \triangleq & !s\{i \leftarrow E\} & \\ (t\langle v^A_\Xi := r, s\rangle)\{i \leftarrow E\} & \triangleq & t\{i \leftarrow E\}\langle v^A_\Xi\{i \leftarrow E\} := r\{i \leftarrow E\}, s\{i \leftarrow E\}\rangle & \\ (\lambda x^A.s)\{i \leftarrow E\} & \triangleq & \lambda x^{A\{\cancel{i} \leftarrow E\}}.s\{i \leftarrow E\} & \\ ([\alpha^A]s)\{i \leftarrow E\} & \triangleq & [\alpha^{A\{\cancel{i} \leftarrow E\}}]s\{i \leftarrow E\} & \\ (\mu\alpha^A.s)\{i \leftarrow E\} & \triangleq & \mu\alpha^{A\{\cancel{i} \leftarrow E\}}.s\{i \leftarrow E\} & \\ \mathsf{gen}_i(s)\{i \leftarrow E\} & \triangleq & \mathsf{gen}_i(s) & \\ \mathsf{gen}_j(s)\{i \leftarrow E\} & \triangleq & \mathsf{gen}_j(s\{i \leftarrow E\}), & \text{if } j \neq i \\ \mathsf{ins}^{E'}_i(s)\{i \leftarrow E\} & \triangleq & \mathsf{ins}^{E'}_i(s) & i \notin \mathsf{FIV}(E') \\ \mathsf{ins}^{E'}_j(s)\{i \leftarrow E\} & \triangleq & \mathsf{ins}^{E'\{\cancel{i} \leftarrow E\}}_j(s\{i \leftarrow E\}), & \text{if } j \neq i \end{array}$$

$$\frac{}{\mathcal{H}, x^A \vdash A \,|\, x^A}\ \mathsf{Var}$$

$$\frac{\mathcal{H}, x^A \vdash B \,|\, s}{\mathcal{H} \vdash A \supset B \,|\, \lambda x^A.s}\ \supset\mathsf{I} \qquad \frac{\mathcal{H} \vdash A \supset B \,|\, s \qquad \mathcal{H} \vdash A \,|\, t}{\mathcal{H} \vdash B \,|\, s \cdot t}\ \supset\mathsf{E}$$

$$\frac{}{\mathcal{H}, v^A_\Xi \vdash A \,|\, v^A_\Xi}\ \mathsf{VarM}$$

$$\frac{\Theta; \cdot; \cdot \vdash A \,|\, s \quad \Theta; \cdot; \cdot \vdash s \equiv t : A \quad \mathsf{FIV}(\Theta) \cap \mathsf{FIV}(A) \subseteq \Xi}{\Theta; \Gamma; \Delta \vdash [\![t]\!]_\Xi A \,|\, !t}\ \Box\mathsf{I}$$

$$\frac{\mathcal{H} \vdash [\![r]\!]_\Xi A \,|\, s \quad \mathcal{H}, v^A_{\Xi'} \vdash C \,|\, t \quad \Xi \cap \mathsf{FIV}(A) \subseteq \Xi'}{\mathcal{H} \vdash C\{v^A_{\Xi'} \leftarrow r\} \,|\, t\langle v^A_{\Xi'} := r, s\rangle}\ \Box\mathsf{E}$$

$$\frac{\mathcal{H} \vdash A \,|\, s}{\mathcal{H} \vdash A \,|\, s + t}\ \mathsf{PlusL} \qquad \frac{\mathcal{H} \vdash A \,|\, t}{\mathcal{H} \vdash A \,|\, s + t}\ \mathsf{PlusR}$$

$$\frac{\mathcal{H}, \alpha^A \vdash A \,|\, s}{\mathcal{H}, \alpha^A \vdash \bot \,|\, [\alpha^A]s}\ \mathsf{Name} \qquad \frac{\mathcal{H}, \alpha^A \vdash \bot \,|\, s}{\mathcal{H} \vdash A \,|\, \mu\alpha^A.s}\ \mathsf{NAbs}$$

$$\frac{\mathcal{H} \vdash A \,|\, s \quad i \notin \mathsf{FIV}(\mathcal{H})}{\mathcal{H} \vdash \forall i.A \,|\, \mathsf{gen}_i(s)}\ \forall\mathsf{I} \qquad \frac{\mathcal{H} \vdash \forall i.A \,|\, s}{\mathcal{H} \vdash A\{i \leftarrow E\} \,|\, \mathsf{ins}^E_i(s)}\ \forall\mathsf{E}$$

Figure 5.1: Axiom and inference schemes of FOHLP

In the clause for $\mathsf{ins}^{E'}_i(s)\{i \leftarrow E\}$, we may assume $i \notin \mathsf{FIV}(E')$ by the variable convention.

Note that, unlike in HLP, the formula which decorates a truth, validity or falsehood variable may change after a substitution. For example $x^A\{i \leftarrow E\} = x^{A\{i \leftarrow E\}}$.

Definition 5.2.7 (Inference schemes) The inference schemes of FOHLP are given in Fig. 5.1. They infer judgements. We say a judgement $\Theta; \Gamma; \Delta \vdash A \,|\, s$ is derivable if there is a derivation of it using the inference schemes of Fig. 5.1 and write $\rhd_{\mathsf{FOHLP}} \Theta; \Gamma; \Delta \vdash A \,|\, s$ in that case. As for HLP, we write $\mathcal{H} \vdash A \,|\, s$ as an abbreviation for $\Theta; \Gamma; \Delta \vdash A \,|\, s$.

Note that the only inference schemes that change w.r.t. those of HLP (Fig. 4.1) are VarM, $\Box$I, $\Box$E, and the new schemes $\forall$I and $\forall$E. For the benefit of the reader, we include them all in Fig. 5.1. We now comment on the new schemes.

- $\Box$I. $\mathsf{FIV}(\Theta) \cap \mathsf{FIV}(A) \subseteq \Xi$ is necessary to avoid binding individual variables which are used as free variables in the premises, much like the restriction that $i \notin \mathsf{FIV}(\Theta, \Gamma, \Delta)$ in $\forall$I. Without that restriction, we would be able to derive judgements which should not be derivable, like $v_i^{P(i)}; \cdot; \cdot \vdash [\![v_i^{P(i)}]\!]_\emptyset P(i) \,|\, !v_i^{P(i)}$, which would mean that $\forall i.P(i)$ is true – and

even valid – if we assume that $P(i)$ is valid for a given i. That is, we would be able to prove $[\![w_i^{P(i)}]\!]_i P(i) \supset \forall i.P(i)$ (see Proposition 5.2.19 for details on why $[\![v_i^{P(i)}]\!]_\emptyset P(i)$ implies $\forall i.P(i)$, particularly axioms **B5** and **B1**.

- $\Box$E. The restriction for $\Box$E prevents a proof of a formula with free individual variables to be used as proof of a formula where those variables are bound. The converse can be done safely (just like a proof of $\forall i.P(i)$ can be used to prove $P(i)$), which is why the inclusion is oriented in only one direction.

Remark 5.2.8 As in HLP, we may also introduce a less general variant of $\Box$I:

$$\frac{\Theta; \cdot; \cdot \vdash A \,|\, t \quad \mathsf{FIV}(\Theta) \cap \mathsf{FIV}(A) \subseteq \Xi}{\Theta; \Gamma; \Delta \vdash [\![t]\!]_\Xi A \,|\, !t} \Box\mathsf{I}'$$

This variant presents the same problem as its propositional counterpart, as equivalence is still required for proof normalization. However, it may be used as a shortcut for $\Box$I when the change of proof witness is not required.

Lemma 5.2.9 (Weakening and Strengthening) Suppose $\rhd_{\mathsf{FOHLP}} \Theta; \Gamma; A \vdash \Delta \,|\, s$. Then:

1. $\rhd_{\mathsf{FOHLP}} \Theta \cup \Theta'; \Gamma \cup \Gamma'; A \vdash \Delta \cup \Delta' \,|\, s$; and
2. $\rhd_{\mathsf{FOHLP}} \Theta \cap \mathsf{FVV}(s); \Gamma \cap \mathsf{FVT}(s); A \vdash \Delta \cap \mathsf{FVF}(s) \,|\, s$.

Proof.- By induction on the derivation of $\Theta; \Gamma; \Delta \vdash A \,|\, s$. For the first item, note that in the case of $\forall$I, if $i \in \mathsf{FIV}(\Theta', \Gamma', \Delta')$, then the variable bound by the $\forall$ can be renamed and the result still holds. In the case of $\Box$I, if there is an individual variable $i \in (\mathsf{FIV}(\Theta') \cap \mathsf{FIV}(A)) \backslash \Xi$, then i is bound in $[\![t]\!]_A \Xi$ and can thus be renamed to some individual variable not in $\mathsf{FIV}(\Theta')$.

The proof of the second item is presented in in Section 5.2.1. $\Box$

Remark 5.2.10 If $\Theta; \Gamma; \Delta \vdash A \,|\, s$ is derivable in HLP, then $\Theta; \Gamma; \Delta \vdash A \,|\, s$ is derivable in FOHLP. Since HLP deals only with propositions (0-ary predicates), we can replicate the original derivation, using the empty set whenever a set of individual variables is required. Whenever $\Box$I is used, the restriction will hold since $\mathsf{FIV}(\Theta) = \emptyset$. The restriction for $\Box$E also holds trivially, since $\Xi = \Xi' = \mathsf{FIV}(A) = \emptyset$.

5.2.1 Proof Witness Equivalence

The following new equivalence rules are introduced: Fig. 5.2 and Fig. 5.3. All other equivalence rules are identical to those of HLP.

Lemma 5.2.11 (Weakening for proof witness equivalence) If the judgement $\Theta; \Gamma; \Delta \vdash s \equiv t : A$ is derivable, then so is $\Theta \cup \Theta'; \Gamma \cup \Gamma'; \Delta \cup \Delta' \vdash s \equiv t : A$.

Proof.- By induction on the derivation of $\Theta; \Gamma; \Delta \vdash s \equiv t : A$. Note that, on the Eq-$\gamma$ rule, the restriction that $\mathsf{FIV}(\Theta) \cap \mathsf{FIV}(A) \subseteq \Xi$ is not an obstacle. This is because any individual variables in $\mathsf{FIV}(A) \backslash \Xi$ are bound in v_Ξ^A, and can thus be renamed if necessary. $\Box$

$$\frac{\Theta;\cdot;\cdot \vdash A \,|\, s \qquad \mathsf{FIV}(\Theta) \cap \mathsf{FIV}(A) \subseteq \Xi \qquad \Theta, v^A_\Xi; \Gamma; \Delta \vdash C \,|\, t}{\Theta; \Gamma; \Delta \vdash\ t\langle v^A_\Xi{:=}s, !s\rangle \equiv t\{v^A_\Xi \leftarrow s\} : C\{v^A_\Xi \leftarrow s\}}\ \mathsf{Eq}\text{-}\gamma$$

$$\frac{\mathcal{H} \vdash A \,|\, s \qquad i \notin \mathsf{FIV}(\Theta, \Gamma, \Delta)}{\mathcal{H} \vdash\ \mathsf{ins}^E_i(\mathsf{gen}_i(s)) \equiv s\{i \leftarrow E\} : A\{i \leftarrow E\}}\ \mathsf{Eq}\text{-}\xi$$

$$\frac{\mathcal{H} \vdash [\![r]\!]_\Xi A \,|\, s \qquad \mathcal{H}, v^A_{\Xi'} \vdash C \,|\, u \qquad \Xi \cap \mathsf{FIV}(A) \subseteq \Xi'}{\mathcal{H} \vdash\ u\langle v^A_{\Xi'}{:=}r, (s+t)\rangle \equiv u\langle v^A_{\Xi'}{:=}r, s\rangle + t : C\{v^A_{\Xi'} \leftarrow r\}}\ \mathsf{Eq}\text{-}\phi_L$$

$$\frac{\mathcal{H} \vdash [\![r]\!]_\Xi A \,|\, t \qquad \mathcal{H}, v^A_{\Xi'} \vdash C \,|\, u \qquad \Xi \cap \mathsf{FIV}(A) \subseteq \Xi'}{\mathcal{H} \vdash\ u\langle v^A_{\Xi'}{:=}r, (s+t)\rangle \equiv s + u\langle v^A_{\Xi'}{:=}r, t\rangle : C\{v^A_{\Xi'} \leftarrow r\}}\ \mathsf{Eq}\text{-}\phi_R$$

$$\frac{\mathcal{H} \vdash \forall i.A \,|\, s}{\mathcal{H} \vdash\ \mathsf{ins}^E_i(s+t) \equiv \mathsf{ins}^E_i(s) + t : A\{i \leftarrow E\}}\ \mathsf{Eq}\text{-}\epsilon_L$$

$$\frac{\mathcal{H} \vdash \forall i.A \,|\, t}{\mathcal{H} \vdash\ \mathsf{ins}^E_i(s+t) \equiv s + \mathsf{ins}^E_i(t) : A\{i \leftarrow E\}}\ \mathsf{Eq}\text{-}\epsilon_R$$

Figure 5.2: New principal rules

Consistency of the equational theory is proved in Section 5.3 (Corollary 5.3.22).

All substitution lemmas hold, as will be proved on section 5.3 once terms have been introduced. For now, we will mention two of these, which shall be used in this section.

Lemma 5.2.12 (Validity Variable Substitution)

1. If $\rhd_{\mathsf{FOHLP}} \Theta, v^A_\Xi; \Gamma; \Delta \vdash B \,|\, s$ and $\rhd_{\mathsf{FOHLP}} \Theta; \cdot; \cdot \vdash A \,|\, t$, then $\rhd_{\mathsf{FOHLP}} \Theta; \Gamma; \Delta \vdash B\{v^A_\Xi \leftarrow t\} \,|\, s\{v^A_\Xi \leftarrow t\}$.

2. If $\rhd_{\mathsf{FOHLP}} \Theta, v^A_\Xi; \Gamma; \Delta \vdash s \equiv r : B$ and $\rhd_{\mathsf{FOHLP}} \Theta; \cdot; \cdot \vdash A \,|\, t$, then $\rhd_{\mathsf{FOHLP}} \Theta; \Gamma; \Delta \vdash s\{v^A_\Xi \leftarrow t\} \equiv r\{v^A_\Xi \leftarrow t\} : B\{v^A_\Xi \leftarrow t\}$.

Proof.- This is a corollary of Lemma 5.3.6, shown on Section 5.3. □

Lemma 5.2.13 (Individual Variable Substitution)

1. If $\rhd_{\mathsf{FOHLP}} \Theta; \Gamma; \Delta \vdash D \,|\, r$, then $\rhd_{\mathsf{FOHLP}} \Theta\{i \leftarrow E\}; \Gamma\{i \leftarrow E\}; \Delta\{i \leftarrow E\} \vdash D\{i \leftarrow E\} \,|\, r\{i \leftarrow E\}$.

2. If $\rhd_{\mathsf{FOHLP}} \Theta; \Gamma; \Delta \vdash r_1 \equiv r_2 : D$, then $\rhd_{\mathsf{FOHLP}} \Theta\{i \leftarrow E\}; \Gamma\{i \leftarrow E\}; \Delta\{i \leftarrow E\} \vdash r_1\{i \leftarrow E\} \equiv r_2\{i \leftarrow E\} : D\{i \leftarrow E\}$.

Proof.- This is a corollary of Lemma 5.3.12, shown on Section 5.3. □

$$\frac{\mathcal{H} \vdash\ s \equiv s' : [\![r]\!]_{\Xi} A \qquad \mathcal{H}, v^A_{\Xi'} \vdash\ t \equiv t' : C \qquad \Xi \cap \mathsf{FIV}(A) \subseteq \Xi'}{\mathcal{H} \vdash\ t\langle v^A_{\Xi'} := r, s\rangle \equiv t'\langle v^A_{\Xi'} := r, s'\rangle : C\{v^A_{\Xi'} \leftarrow r\}}\ \mathsf{Eq}\text{-}\langle\rangle$$

$$\frac{\mathcal{H} \vdash\ s \equiv t : A \qquad i \notin \mathsf{FIV}(\Theta, \Gamma, \Delta)}{\mathcal{H} \vdash\ \mathsf{gen}_i(s) \equiv \mathsf{gen}_i(t) : \forall i.A}\ \mathsf{Eq}\text{-}\mathsf{gen}$$

$$\frac{\mathcal{H} \vdash\ s \equiv t : \forall i.A}{\mathcal{H} \vdash\ \mathsf{ins}^E_i(s) \equiv \mathsf{ins}^E_i(t) : A\{i \leftarrow E\}}\ \mathsf{Eq}\text{-}\mathsf{ins}$$

Figure 5.3: New compatibility rules

Lemma 5.2.14 If $\rhd_{\mathsf{FOHLP}}\ \Theta; \Gamma; \Delta \vdash\ s \equiv t : D$, both $\Theta; \Gamma; \Delta \vdash D \,|\, s$ and $\Theta; \Gamma; \Delta \vdash D \,|\, t$ are derivable.

Proof.- By induction on the derivation of $\Theta; \Gamma; \Delta \vdash\ s \equiv t : D$. We analyze the last rule used. For all rules not listed here, the proof is analogous to the one shown for Lemma 4.2.15.

- $\mathsf{Eq}\text{-}\gamma$: $s = t'\langle v^A_{\Xi} := s', !s'\rangle$, $t = t'\{v^A_{\Xi} \leftarrow s'\}$, $D = C\{v^A_{\Xi} \leftarrow s'\}$, both $\Theta; \cdot; \cdot \vdash A \,|\, s'$ and $\Theta, v^A_{\Xi}; \Gamma; \Delta \vdash C \,|\, t'$ are derivable by hypothesis, and $\mathsf{FIV}(\Theta) \cap \mathsf{FIV}(A) \subseteq \Xi$.

  Let $\Xi_1 = \mathsf{FIV}(\Theta)$. Since $\Xi_1 \cap \mathsf{FIV}(A) \subseteq \Xi_1$, then by $\Box\mathsf{I}$, we can derive $\Theta; \Gamma; \Delta \vdash [\![s']\!]_{\Xi_1} A \,|\, !s'$ – note that we use the same proof witness s' on both sides of the equivalence, which is the same as using an alternate rule $\Box\mathsf{I}$' analogous to the one discussed for HLP. Now, by $\Box\mathsf{E}$, we can derive $\Theta; \Gamma; \Delta \vdash C\{v^A_{\Xi} \leftarrow s'\} \,|\, t'\langle v^A_{\Xi} := s', !s'\rangle$.

  The judgement $\Theta; \Gamma; \Delta \vdash C\{v^A_{\Xi} \leftarrow s'\} \,|\, t'\{v^A_{\Xi} \leftarrow s'\}$ can be obtained from $\Theta, v^A_{\Xi}; \Gamma; \Delta \vdash C \,|\, t'$ and $\Theta; \cdot; \cdot \vdash A \,|\, s'$ by Lemma 5.2.12.

- $\mathsf{Eq}\text{-}\xi$: $s = \mathsf{ins}^E_i(\mathsf{gen}_i(s'))$, $t = s'\{i \leftarrow E\}$, $D = A\{i \leftarrow E\}$, $\Theta; \Gamma; \Delta \vdash A \,|\, s'$ is derivable by hypothesis, and $i \notin \mathsf{FIV}(\Theta, \Gamma, \Delta)$.

  By $\forall\mathsf{I}$, we can derive $\Theta; \Gamma; \Delta \vdash \forall i.A \,|\, \mathsf{gen}_i(s')$. And then, by $\forall\mathsf{E}$, we obtain $\Theta; \Gamma; \Delta \vdash A\{i \leftarrow E\} \,|\, \mathsf{ins}^E_i(\mathsf{gen}_i(s'))$.

  $\Theta; \Gamma; \Delta \vdash A\{i \leftarrow E\} \,|\, s'\{i \leftarrow E\}$ can be obtained from $\Theta; \Gamma; \Delta \vdash A \,|\, s'$ by Lemma 5.2.13.

- $\mathsf{Eq}\text{-}\phi_L$: $s = u\langle v^A_{\Xi'} := r, (s' + t')\rangle$, $t = u\langle v^A_{\Xi'} := r, s'\rangle + t'$, $D = C\{v^A_{\Xi'} \leftarrow r\}$, both $\Theta; \Gamma; \Delta \vdash [\![r]\!]_{\Xi} A \,|\, s'$ and $\Theta, v^A_{\Xi'}; \Gamma; \Delta \vdash C \,|\, u$ are derivable by hypothesis and $\Xi \cap \mathsf{FIV}(A) \subseteq \Xi'$. $\Theta; \Gamma; \Delta \vdash D \,|\, s$ is obtained by applying PlusL and then $\Box\mathsf{E}$; and, conversely, $\Theta; \Gamma; \Delta \vdash D \,|\, t$ results from applying $\Box\mathsf{E}$ and then PlusL.

- $\mathsf{Eq}\text{-}\phi_R$: this case is analogous to the previous one.

- $\mathsf{Eq}\text{-}\epsilon_L$: $s = \mathsf{ins}^E_i(s' + t')$, $t = \mathsf{ins}^E_i(s') + t'$, $D = A\{i \leftarrow E\}$ and $\Theta; \Gamma; \Delta \vdash \forall i.A \,|\, s'$ is derivable by hypothesis.

  $\Theta; \Gamma; \Delta \vdash D \,|\, s$ is obtained by applying PlusL and then $\forall\mathsf{E}$; and, conversely, $\Theta; \Gamma; \Delta \vdash D \,|\, t$ results from applying $\forall\mathsf{E}$ and then PlusL.

- $\mathsf{Eq}\text{-}\epsilon_R$: this case is analogous to the previous one.

- Eq-$\langle\rangle$: $s = t_1\langle v^A_{\Xi'}{:=}r, s_1\rangle$, $t = t_2\langle v^A_{\Xi'}{:=}r, s_2\rangle$, $D = C\{v^A_{\Xi'}{\leftarrow}\ r\}$, both $\Theta;\Gamma;\Delta \vdash\ s_1 \equiv s_2 : [\![r]\!]_{\Xi}A$ and $\Theta, v^A_{\Xi'};\Gamma;\Delta \vdash\ t_1 \equiv t_2 : C$ are derivable by hypothesis, and $\Xi \cap \mathsf{FIV}(A) \subseteq \Xi'$. By induction hypothesis, we can derive:
  - $\Theta;\Gamma;\Delta \vdash [\![r]\!]_{\Xi}A \,|\, s_1$
  - $\Theta, v^A_{\Xi'};\Gamma;\Delta \vdash C \,|\, t_1$
  - $\Theta;\Gamma;\Delta \vdash [\![r]\!]_{\Xi}A \,|\, s_2$
  - $\Theta, v^A_{\Xi'};\Gamma;\Delta \vdash C \,|\, t_2$

  The results are obtained from these by $\Box$E.
- Eq-gen: the results are obtained by induction hypothesis and $\forall$I.
- Eq-ins: the results are obtained by induction hypothesis and $\forall$E.

$\Box$

Remark 5.2.15 If $\Theta;\Gamma;\Delta \vdash A \,|\, s$ is derivable in FOHLP, then there is a derivation of $\Theta;\Gamma;\Delta \vdash A \,|\, s$ which does not make use of the equivalence rules. That is, there is a derivation which uses $\Box\mathsf{I}'$ instead of $\Box$I. This follows from Lemma 5.2.14 (whose proof does not introduce applications of equivalence rules).

Now we can finally complete the proof of Lemma 5.2.9 (part 2: Strengthening).

Proof.- By induction on the derivation of $\Theta;\Gamma;\Delta \vdash A \,|\, s$. We assume that the derivation does not make use of equivalence rules (i.e., it uses $\Box$I' instead of $\Box$I). This is always possible by Remark 5.2.15.

We will show the proof for the rules which differ from those of HLP.

- Case VarM: $s = v^A_{\Xi}$. $v^A_{\Xi};\cdot;\cdot \vdash A \,|\, v^A_{\Xi}$ is derivable by VarM.
- Case $\Box$I': $s = !t$, $A = [\![t]\!]_{\Xi}B$, $\mathsf{FIV}(\Theta) \cap \mathsf{FIV}(B) \subseteq \Xi$, and $\Theta;\cdot;\cdot \vdash B \,|\, t$ is derivable by hypothesis. The judgement $\Theta \cap \mathsf{FVV}(t);\cdot;\cdot \vdash B \,|\, t$ is derivable by IH. Since $\mathsf{FIV}(\Theta) \cap \mathsf{FIV}(B) \subseteq \Xi$, then $\mathsf{FIV}(\Theta \cap \mathsf{FVV}(t) \cap \mathsf{FIV}(B)) \subseteq \Xi$. Therefore $\Theta \cap \mathsf{FVV}(t);\cdot;\cdot \vdash [\![t]\!]_{\Xi}B \,|\, !t$ is obtained by $\Box$I'.
- Case $\Box$E: $A = C\{v^B_{\Xi'} \leftarrow r\}$ and $s = t\langle v^B_{\Xi'}{:=}r, s'\rangle$ By induction hypothesis, both $\Theta \cap \mathsf{FVV}(s');\Gamma \cap \mathsf{FVT}(s');\Delta \cap \mathsf{FVF}(s') \vdash [\![r]\!]_{\Xi}B \,|\, s'$ and $\Theta, v^B_{\Xi'} \cap \mathsf{FVV}(t);\Gamma \cap \mathsf{FVT}(t);\Delta \cap \mathsf{FVF}(t) \vdash C \,|\, t$ are derivable. By Weakening, both $\Theta \cap \mathsf{FVV}(t\langle v^B_{\Xi'}{:=}r, s'\rangle);\Gamma \cap \mathsf{FVT}(s', t);\Delta \cap \mathsf{FVF}(s', t) \vdash [\![r]\!]_{\Xi}B \,|\, s'$ and $\Theta, v^B_{\Xi'} \cap \mathsf{FVV}(t\langle v^B_{\Xi'}{:=}r, s'\rangle);\Gamma \cap \mathsf{FVT}(s', t);\Delta \cap \mathsf{FVF}(s', t) \vdash C \,|\, t$ are also derivable. Since Ξ, Ξ' and B do not change, the restriction still holds. The result is obtained by $\Box$E.
- Case $\forall$I: $A = \forall i.B$ and $s = \mathsf{gen}_i(t)$. $\mathsf{FVV}(s) = \mathsf{FVV}(t)$, $\mathsf{FVT}(s) = \mathsf{FVT}(t)$ and $\mathsf{FVF}(s) = \mathsf{FVF}(t)$. $\Theta \cap \mathsf{FVV}(t);\Gamma \cap \mathsf{FVT}(t);\Delta \cap \mathsf{FVF}(t) \vdash B \,|\, t$ is derivable by IH, and since $i \notin \mathsf{FIV}(\Theta, \Gamma, \Delta)$, then $i \notin \mathsf{FIV}(\Theta \cap \mathsf{FVV}(t), \Gamma \cap \mathsf{FVT}(t), \Delta \cap \mathsf{FVF}(t))$. The result holds by $\forall$I.
- Case $\forall$E: $A = B\{i \leftarrow E\}$, $s = \mathsf{ins}^E_i(t)$, $\mathsf{FVV}(s) = \mathsf{FVV}(t)$, $\mathsf{FVT}(s) = \mathsf{FVT}(t)$ and $\mathsf{FVF}(s) = \mathsf{FVF}(t)$. $\Theta \cap \mathsf{FVV}(t);\Gamma \cap \mathsf{FVT}(t);\Delta \cap \mathsf{FVF}(t) \vdash \forall i.B \,|\, t$ is derivable by IH, thus the result follows by $\forall$E.

$\Box$

5.2.2 Translation from FOLP to FOHLP

All theorems of FOLP can be proved in FOHLP. However, we require some minimum assumptions on the use of proof constants in the Necessitation inference rule **Nec**.

Definition 5.2.16 (Constant Specification) A *constant specification* is a set $\mathcal{C}$ of formulas of FOLP of the form $[\![c]\!]_\emptyset A$. It is assumed that A is an axiom. A proof meets a constant specification $\mathcal{C}$ if whenever the rule **Nec** is used to introduce $[\![c]\!]_\emptyset A$, then $[\![c]\!]_\emptyset A$ is in $\mathcal{C}$. A constant specification is *injective* if $[\![c]\!]_\emptyset A_1 \in \mathcal{C}$ and $[\![c]\!]_\emptyset A_2 \in \mathcal{C}$, implies $A_1 = A_2$.

Given an injective constant specification $\mathcal{C}$ we can associate a derivation in FOHLP to the corresponding constant. For that it is convenient to decorate a constant c such that $[\![c]\!]_\emptyset A \in \mathcal{C}$, with information describing which axiom A is an instance of and also which instance is denoted. We put the name of the axiom as superindex and the instances of its metavariables as its subindices. For example, we write $c^{\mathbf{A1}}_{\Xi,x^P,P}$ for the constant corresponding to the instance $[\![x]\!]_\Xi P \supset P$ of the axiom **B1** (and similarly for the other axioms).

$$
\begin{array}{rcl}
\underline{c^{\mathbf{A1a}}_{A,B}} & \triangleq & (\lambda x^{\underline{A}}.\lambda y^{\underline{B}}.x) \\
\underline{c^{\mathbf{A1b}}_{A,B,C}} & \triangleq & (\lambda x^{\underline{A\supset B\supset C}}.\lambda y^{\underline{A\supset B}}.\lambda z^{\underline{A}}.x \cdot z \cdot (y \cdot z)) \\
\underline{c^{\mathbf{A1c}}_{A}} & \triangleq & (\lambda y^{\underline{\neg\neg A}}.\mu\alpha^{\underline{A}}.y \cdot \lambda x^{\underline{A}}.[\alpha^{\underline{A}}]x) \\
\underline{c^{\mathbf{A1d}}_{A,i,E}} & \triangleq & \lambda x^{\underline{\forall i.A}}.\mathsf{ins}^E_i(x^{\underline{\forall i.A}}) \\
\underline{c^{\mathbf{A1e}}_{A,B,i}} & \triangleq & \lambda x^{\underline{\forall i.A\supset B}}.\lambda y^{\underline{\forall i.A}}.\mathsf{gen}_j(\mathsf{ins}^j_i(x^{\underline{\forall i.(A\supset B)}})\cdot\mathsf{ins}^j_i(y^{\underline{\forall i.A}})) \\
\underline{c^{\mathbf{A1f}}_{A,i}} & \triangleq & \lambda x^{\underline{A}}.\mathsf{gen}_i(x^{\underline{A}}) \\
\underline{c^{\mathbf{A2}}_{\Xi,i,t,A}} & \triangleq & \lambda x^{[\![\underline{t}]\!]_{\Xi,i\underline{A}}}.!v^{\underline{A}}_{\Xi}\langle v^{\underline{A}}_{\Xi}{:=}\underline{t}, x^{[\![\underline{t}]\!]_{\Xi,i\underline{A}}}\rangle \\
\underline{c^{\mathbf{A3}}_{\Xi,i,t,A}} & \triangleq & \lambda x^{[\![\underline{t}]\!]_{\Xi\underline{A}}}.!v^{\underline{A}}_{\Xi,i}\langle v^{\underline{A}}_{\Xi,i}{:=}\underline{t}, x^{[\![\underline{t}]\!]_{\Xi\underline{A}}}\rangle \\
\underline{c^{\mathbf{B1}}_{\Xi,t,A}} & \triangleq & \lambda x^{[\![\underline{t}]\!]_{\Xi\underline{A}}}.v^{\underline{A}}_{\Xi}\langle v^{\underline{A}}_{\Xi}{:=}\underline{t}, x^{[\![\underline{t}]\!]_{\Xi\underline{A}}}\rangle \\
\underline{c^{\mathbf{B2}}_{\Xi,s,t,A,B}} & \triangleq & \lambda x^{[\![\underline{s}]\!]_{\Xi\underline{A\supset B}}}.\lambda y^{[\![\underline{t}]\!]_{\Xi\underline{A}}}.!(w^{\underline{A\supset B}}_{\Xi}\cdot v^{\underline{A}}_{\Xi})\langle w^{\underline{A\supset B}}_{\Xi}{:=}\underline{s}, x\rangle\langle v^{\underline{A}}_{\Xi}{:=}\underline{t}, y\rangle \\
\underline{c^{\mathbf{B3a}}_{\Xi,s,t,A}} & \triangleq & \lambda x^{[\![\underline{s}]\!]_{\Xi\underline{A}}}.!(v^{\underline{A}}_{\Xi}+\underline{t})\langle v^{\underline{A}}_{\Xi}{:=}\underline{s}, x^{[\![\underline{s}]\!]_{\Xi\underline{A}}}\rangle \\
\underline{c^{\mathbf{B3b}}_{\Xi,s,t,A}} & \triangleq & \lambda x^{[\![\underline{t}]\!]_{\Xi\underline{A}}}.!(\underline{s}+v^{\underline{A}}_{\Xi})\langle v^{\underline{A}}_{\Xi}{:=}\underline{t}, x^{[\![\underline{s}]\!]_{\Xi\underline{A}}}\rangle \\
\underline{c^{\mathbf{B4}}_{\Xi,s,A}} & \triangleq & \lambda x^{[\![\underline{s}]\!]_{\Xi\underline{A}}}.!!v^{\underline{A}}_{\Xi}\langle v^{\underline{A}}_{\Xi}{:=}\underline{s}, x^{[\![\underline{s}]\!]_{\Xi\underline{A}}}\rangle \\
\underline{c^{\mathbf{B5}}_{\Xi,i,t,A}} & \triangleq & \lambda x^{[\![\underline{t}]\!]_{\Xi\underline{A}}}.!\mathsf{gen}_i(v^{\underline{A}}_{\Xi})\langle v^{\underline{A}}_{\Xi}{:=}\underline{t}, x^{[\![\underline{t}]\!]_{\Xi\underline{A}}}\rangle
\end{array}
$$

We introduce a simple translation from FOLP-proofs with injective constant specifications to FOHLP-proofs. To that effect, we introduce a translation $\underline{\bullet}$ from FOLP-formulas to FOHLP-formulas. It simply traverses the structure of formulas and proof terms, replacing all proof constants by appropriate FOHLP-proofs of the axioms they represent.

Definition 5.2.17 The translation $\underline{\bullet}$ from FOLP to FOHLP is defined as follows (individual variables and objects translate to themselves), by extending its domain to formulas:

$$
\begin{array}{rcl}
\underline{E} & \triangleq & E \\
\underline{P(E_1,\ldots,E_n)} & \triangleq & P(E_1,\ldots,E_n) \\
\underline{\bot} & \triangleq & \bot \\
\underline{A \supset B} & \triangleq & \underline{A} \supset \underline{B} \\
\underline{[\![s]\!]_\Xi A} & \triangleq & [\![\underline{s}]\!]_\Xi \underline{A} \\
\underline{\forall i.A} & \triangleq & \forall i.\underline{A}
\end{array}
$$

proof witnesses and contexts:

$$
\begin{array}{rcl}
\underline{x^A} & \triangleq & x^{\underline{A}} \\
\underline{s \cdot t} & \triangleq & \underline{s} \cdot \underline{t} \\
\underline{!s} & \triangleq & !\underline{s} \\
\underline{s + t} & \triangleq & \underline{s} + \underline{t} \\
\underline{\mathsf{gen}_i(s)} & \triangleq & \mathsf{gen}_i(\underline{s}) \\
\underline{\Gamma} & \triangleq & \{x^{\underline{A}} \mid x^A \in \Gamma\}
\end{array}
$$

Remark 5.2.18 The translations of $c^{\mathbf{A2}}_{\Xi,i,t,A}$ and $c^{\mathbf{A3}}_{\Xi,i,t,A}$ are almost identical, exchanging the occurrences of Ξ and Ξ, i.

Proposition 5.2.19 If $\Gamma \vdash D$ is derivable in FOLP, then $\cdot; \underline{\Gamma}; \cdot \vdash \underline{D} \mid s$ is derivable in FOHLP for some proof witness s.

Proof.- By induction on the derivation of $\Gamma \vdash D$. We show that the axioms and inference schemes of FOLP are derivable in FOHLP.

If $x^D \in \Gamma$, the result is immediate by Var, and $s = x^{\underline{D}}$.

For the cases where D is an axiom, we will show that $\underline{D}$ can be derived in the empty context. By *weakening*, this means it can also be derived in any context Γ.

We prove the translations of the FOLP-axioms in FOHLP. The formulas and proof witnesses shown in the proof are the translations of FOLP-formulas and proof terms.

- **A1a-c., B1-4.** The proofs are analogous to those for the translation from LP to HLP. Whenever a set of individual variables is required, use the same Ξ from the corresponding axiom instance.
- **A1d.** $(\forall i.A) \supset A\{i \leftarrow E\}$

$$
\dfrac{\dfrac{\dfrac{}{\cdot; x^{\forall i.A}; \cdot \vdash \forall i.A \mid x^{\forall i.A}}\,\mathsf{Var}}{\cdot; x^{\forall i.A}; \cdot \vdash A\{i \leftarrow E\} \mid \mathsf{ins}^E_i(x^{\forall i.A})}\,\forall\mathsf{E}}{\cdot; \cdot; \cdot \vdash (\forall i.A) \supset A\{i \leftarrow E\} \mid \lambda x^{\forall i.A}.\mathsf{ins}^E_i(x^{\forall i.A})}\,\supset\mathsf{I}
$$

- **A1e.** $(\forall i.(A \supset B)) \supset (\forall i.A) \supset \forall i.B$

Let $\Gamma_1 = x^{\forall i.(A \supset B)}, y^{\forall i.A}$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\cdot;\Gamma_1;\cdot \vdash \forall i.(A \supset B) \mid x^{\forall i.(A\supset B)}}\,\mathsf{Var}}{\cdot;\Gamma_1;\cdot \vdash A \supset B\{i \leftarrow j\} \mid \mathsf{ins}_i^j(x^{\forall i.(A\supset B)})}\,\forall\mathsf{E} \quad \dfrac{\dfrac{}{\cdot;\Gamma_1;\cdot \vdash \forall i.A \mid y^{\forall i.A}}\,\mathsf{Var}}{\cdot;\Gamma_1;\cdot \vdash A\{i \leftarrow j\} \mid \mathsf{ins}_i^j(y^{\forall i.A})}\,\forall\mathsf{E}}{\cdot;\Gamma_1;\cdot \vdash B\{i \leftarrow j\} \mid \mathsf{ins}_i^j(x^{\forall i.(A\supset B)}) \cdot \mathsf{ins}_i^j(y^{\forall i.A})}\,\supset\mathsf{E}}{\cdot;\Gamma_1;\cdot \vdash \forall j.B\{i \leftarrow j\} \mid \mathsf{gen}_j(\mathsf{ins}_i^j(x^{\forall i.(A\supset B)}) \cdot \mathsf{ins}_i^j(y^{\forall i.A}))}\,\forall\mathsf{I}}{\cdot;x^{\forall i.(A\supset B)};\cdot \vdash (\forall i.A) \supset \forall i.B \mid \lambda y^{\forall i.A}.\mathsf{gen}_j(\mathsf{ins}_i^j(x^{\forall i.(A\supset B)}) \cdot \mathsf{ins}_i^j(y^{\forall i.A}))}\,\supset\mathsf{I}}{\cdot;\cdot;\cdot \vdash (\forall i.(A \supset B)) \supset (\forall i.A) \supset \forall i.B \mid \lambda x^{\forall i.(A\supset B)}.\lambda y^{\forall i.A}.\mathsf{gen}_j(\mathsf{ins}_i^j(x^{\forall i.(A\supset B)}) \cdot \mathsf{ins}_i^j(y^{\forall i.A}))}\,\supset\mathsf{I}$$

Note: we choose a fresh j to ensure that $j \notin \mathsf{FIV}(\Gamma_1)$, and use the fact that $\forall j.B\{i \leftarrow j\} = \forall i.B$.

- **A1f.** $A \supset \forall i.A$ if $i \notin \mathsf{FIV}(A)$

$$\dfrac{\dfrac{\dfrac{}{\cdot;x^A;\cdot \vdash A \mid x^A}\,\mathsf{Var}}{\cdot;x^A;\cdot \vdash \forall i.A \mid \mathsf{gen}_i(x^A)}\,\forall\mathsf{I}}{\cdot;\cdot;\cdot \vdash A \supset \forall i.A \mid \lambda x^A.\mathsf{gen}_i(x^A)}\,\supset\mathsf{I}$$

Note that, since $i \notin \mathsf{FIV}(A)$, then $i \notin \mathsf{FIV}(\{x^A\})$.

- **A2.** $([\![t]\!]_{\Xi,i}A) \supset [\![t]\!]_{\Xi}A$ if $i \notin \mathsf{FIV}(A)$

$$\dfrac{\dfrac{\dfrac{}{\cdot;x^{[\![t]\!]_{\Xi,i}A};\cdot \vdash [\![t]\!]_{\Xi,i}A \mid x^{[\![t]\!]_{\Xi,i}A}}\,\mathsf{Var} \quad \dfrac{\dfrac{}{v_{\Xi}^A;\cdot;\cdot \vdash A \mid v_{\Xi}^A}\,\mathsf{VarM}}{v_{\Xi}^A;x^{[\![t]\!]_{\Xi,i}A};\cdot \vdash [\![v_{\Xi}^A]\!]_{\Xi}A \mid !v_{\Xi}^A}\,\Box\mathsf{I}}{\cdot;x^{[\![t]\!]_{\Xi,i}A};\cdot \vdash [\![t]\!]_{\Xi}A \mid !v_{\Xi}^A\langle v_{\Xi}^A{:=}t, x^{[\![t]\!]_{\Xi,i}A}\rangle}\,\Box\mathsf{E}}{\cdot;\cdot;\cdot \vdash ([\![t]\!]_{\Xi,i}A) \supset [\![t]\!]_{\Xi}A \mid \lambda x^{[\![t]\!]_{\Xi,i}A}.!v_{\Xi}^A\langle v_{\Xi}^A{:=}t, x^{[\![t]\!]_{\Xi,i}A}\rangle}\,\supset\mathsf{I}$$

The restriction for $\Box\mathsf{I}$ holds, since $\Xi \cap \mathsf{FIV}(A) \subseteq \Xi$. The restriction for $\Box\mathsf{E}$ holds, since $\Xi, i \cap \mathsf{FIV}(A) \subseteq \Xi$ because $i \notin \mathsf{FIV}(A)$.

- **A3.** $([\![t]\!]_{\Xi}A) \supset [\![t]\!]_{\Xi,i}A$

$$\dfrac{\dfrac{\dfrac{}{\cdot;x^{[\![t]\!]_{\Xi}A};\cdot \vdash [\![t]\!]_{\Xi}A \mid x^{[\![t]\!]_{\Xi}A}}\,\mathsf{Var} \quad \dfrac{\dfrac{}{v_{\Xi,i}^A;\cdot;\cdot \vdash A \mid v_{\Xi,i}^A}\,\mathsf{VarM}}{v_{\Xi,i}^A;x^{[\![t]\!]_{\Xi}A};\cdot \vdash [\![v_{\Xi,i}^A]\!]_{\Xi,i}A \mid !v_{\Xi,i}^A}\,\Box\mathsf{I}}{\cdot;x^{[\![t]\!]_{\Xi}A};\cdot \vdash [\![t]\!]_{\Xi,i}A \mid !v_{\Xi,i}^A\langle v_{\Xi,i}^A{:=}t, x^{[\![t]\!]_{\Xi}A}\rangle}\,\Box\mathsf{E}}{\cdot;\cdot;\cdot \vdash ([\![t]\!]_{\Xi}A) \supset [\![t]\!]_{\Xi,i}A \mid \lambda x^{[\![t]\!]_{\Xi}A}.!v_{\Xi,i}^A\langle v_{\Xi,i}^A{:=}t, x^{[\![t]\!]_{\Xi}A}\rangle}\,\supset\mathsf{I}$$

The derivation is almost identical to that for **A2**, exchanging the occurrences of Ξ and Ξ, i. The restriction for $\Box\mathsf{I}$ holds, since $\Xi, i \cap \mathsf{FIV}(A) \subseteq \Xi, i$. The restriction for $\Box\mathsf{E}$ holds, since $\Xi \cap \mathsf{FIV}(A) \subseteq \Xi, i$.

- **B5.** $([\![t]\!]_{\Xi}A) \supset [\![\mathsf{gen}_i(t)]\!]_{\Xi}\forall i.A$ if $i \notin \Xi$

$$\dfrac{\dfrac{\dfrac{}{\cdot; x^{[\![t]\!]_{\Xi}A}; \cdot \vdash [\![t]\!]_{\Xi}A \mid x^{[\![t]\!]_{\Xi}A}}\,\mathsf{Var} \quad \dfrac{\dfrac{\dfrac{}{v^A_{\Xi}; \cdot; \cdot \vdash A \mid v^A_{\Xi}}\,\mathsf{VarM}}{v^A_{\Xi}; \cdot; \cdot \vdash \forall i.A \mid \mathsf{gen}_i(v^A_{\Xi})}\,\forall\mathsf{I}}{v^A_{\Xi}; x^{[\![t]\!]_{\Xi}A}; \cdot \vdash [\![\mathsf{gen}_i(v^A_{\Xi})]\!]_{\Xi}\forall i.A \mid !\mathsf{gen}_i(v^A_{\Xi})}\,\Box\mathsf{I}}{\cdot; x^{[\![t]\!]_{\Xi}A}; \cdot \vdash [\![\mathsf{gen}_i(t)]\!]_{\Xi}\forall i.A \mid !\mathsf{gen}_i(v^A_{\Xi})\langle v^A_{\Xi}{:=}t, x^{[\![t]\!]_{\Xi}A}\rangle}\,\Box\mathsf{E}}{\cdot; \cdot; \cdot \vdash ([\![t]\!]_{\Xi}A) \supset [\![\mathsf{gen}_i(t)]\!]_{\Xi}\forall i.A \mid \lambda x^{[\![t]\!]_{\Xi}A}.!\mathsf{gen}_i(v^A_{\Xi})\langle v^A_{\Xi}{:=}t, x^{[\![t]\!]_{\Xi}A}\rangle}\,\supset\mathsf{I}$$

  The restriction for $\forall\mathsf{I}$ holds, since $i \notin \Xi$. The restriction for $\Box\mathsf{I}$ holds, since $\Xi \cap \mathsf{FIV}(\forall i.A) \subseteq \Xi$. And, of course, the restriction for $\Box\mathsf{E}$ holds, since $\Xi \cap \mathsf{FIV}(A) \subseteq \Xi$.

For inference rules **MP** and **Nec**, the proof is analogous to that of rules **MP** and **Gen** of LP.

Rule **Gen** of FOLP is an instance of $\forall\mathsf{I}$ plus Weakening.

$\Box$

5.2.3 Translation from FOHLP to FOLP

The translation $\bullet^{\star}$ from FOHLP to FOLP is defined as follows for formulas:

$$\begin{array}{rcl} E^{\star} & \triangleq & E \\ \bot^{\star} & \triangleq & \bot \\ (A \supset B)^{\star} & \triangleq & A^{\star} \supset B^{\star} \\ \forall i.A^{\star} & \triangleq & \forall i.A^{\star} \\ [\![s]\!]_{\Xi}A^{\star} & \triangleq & [\![s^{\star}]\!]_{\Xi}A^{\star} \end{array}$$

For contexts as:

$$\begin{array}{rcl} \cdot^{\star} & \triangleq & \cdot \\ (\Theta, v^A_{\Xi})^{\star} & \triangleq & \Theta^{\star}, [\![v^{A^{\star}}]\!]_{\Xi}A^{\star} \\ (\Gamma, x^A)^{\star} & \triangleq & \Gamma^{\star}, [\![x^{A^{\star}}]\!]_{\mathsf{FIV}(A)}A^{\star} \\ (\Delta, \alpha^A)^{\star} & \triangleq & \Delta^{\star}, [\![\alpha^{\neg A^{\star}}]\!]_{\mathsf{FIV}(A)}\neg A^{\star} \end{array}$$

Remark 5.2.20 For every formula A, $\mathsf{FIV}(A) = \mathsf{FIV}(A^{\star})$. Note that FOHLP-proof witnesses and FOLP-proof terms play no role in the definition of the free individual variables of a formula.

For proof witnesses (disregarding lambda and name abstractions, unbox and proof checker) we have:

$$\begin{array}{rcl} (x^A)^{\star} & \triangleq & x^{A^{\star}} \\ (v^A_{\Xi})^{\star} & \triangleq & v^{A^{\star}} \\ (s \cdot t)^{\star} & \triangleq & s^{\star} \cdot t^{\star} \\ ([\alpha^A]s)^{\star} & \triangleq & \alpha^{\neg A^{\star}} \cdot s^{\star} \end{array} \qquad \begin{array}{rcl} (s + t)^{\star} & \triangleq & s^{\star} + t^{\star} \\ \mathsf{gen}_i(s)^{\star} & \triangleq & \mathsf{gen}_i(s^{\star}) \\ \mathsf{ins}^E_i(s)^{\star} & \triangleq & c^{\mathbf{A1d}} \cdot s^{\star} \end{array}$$

Here **A1d** refers to the axiom scheme of classical logic $(\forall i.A) \supset A\{i \leftarrow E\}$.

Translating lambda and name abstraction. We now explain how we address lambda abtraction (name abstraction is addressed similarly). Bear in mind that the translation of a FOHLP-judgement $\Theta; \Gamma; \Delta \vdash A \,|\, s$ is defined as:

$$(\Theta; \Gamma; \Delta \vdash A \,|\, s)^\star \triangleq \Theta^\star, \Gamma^\star, \Delta^\star \vdash [\![s^\star]\!]_{\mathsf{FIV}(\Theta^\star \cup \Gamma^\star \cup \Delta^\star) \cap \mathsf{FIV}(A^\star)} A^\star$$

Suppose that the last scheme applied in the derivation π of a judgement $\Theta; \Gamma; \Delta \vdash C \,|\, s$ is:

$$\frac{\Theta; \Gamma, x^A; \Delta \vdash B \,|\, s}{\Theta; \Gamma; \Delta \vdash A \supset B \,|\, \lambda x^A.s} \supset \mathsf{I}$$

The IH of our forthcoming proof (Prop. 5.2.21) will yield derivability in FOLP of:

$$\Theta^\star \cup \Gamma^\star, [\![x^{A^\star}]\!]_{\mathsf{FIV}(A^\star)} A^\star \cup \Delta^\star \vdash [\![s^\star]\!]_{\mathsf{FIV}(\Theta^\star \cup \Gamma^\star \cup \Delta^\star) \cap \mathsf{FIV}(B^\star)} B^\star \tag{5.1}$$

However, we are after derivability of $\Theta^\star \cup \Gamma^\star \cup \Delta^\star \vdash [\![t]\!]_{\mathsf{FIV}(\Theta^\star \cup \Gamma^\star \cup \Delta^\star) \cap \mathsf{FIV}(A^\star \supset B^\star)} (A^\star \supset B^\star)$, for an appropriate proof term t. As in HLP, building a derivation of this judgement requires three steps:

1. We first need to "drop" the outermost modalities of the formulas $[\![x^{A^\star}]\!]_{\mathsf{FIV}(A^\star)} A^\star$ and $[\![s^\star]\!]_{\mathsf{FIV}(\Theta^\star \cup \Gamma^\star \cup \Delta^\star) \cap \mathsf{FIV}(B^\star)} B^\star$ from (5.1). This is achieved via Stripping (Lem. 4.1.6).

2. This allows us then to resort to the standard Deduction Theorem to deduce $A^\star \supset B^\star$.

3. Finally, we resort to the reflective capabilities of FOLP in order to deduce the appropriate proof term t. This is achieved via the Internalization Lemma (Lem. 5.1.8).

These three steps conform the content of the Abstraction Lemma (Lemma 5.1.11). Note that t is thus a function of the original FOLP derivation of (5.1) and, in turn, this is obtained from π. Since there may be multiple FOHLP derivations of an FOHLP judgement we shall assume in our proof of Proposition 5.2.21 (and Corollary 5.2.22) that π is *canonical* in the sense that multiple occurrences of a judgement in π all have the exact same proof. The clauses defining the translation of lambda $(\lambda x^A.s)^\star$ and name abstraction $(\mu\alpha^A.s)^\star$, are:

$$\begin{array}{lll}
(\lambda x^A.s)^\star & \triangleq & t_\lambda^{A^\star \supset B^\star}, \quad \text{if there exists a FOLP-context } [\![\vec{u}]\!]_{\vec{\Xi}}\Gamma \text{, a formula } B \text{ and a fresh } y \text{ s.t.} \\
 & & \rhd_{\mathsf{FOLP}} [\![\vec{u}]\!]_{\vec{\Xi}}\Gamma, y^{[\![x^{A^\star}]\!]_{\mathsf{FIV}(A)} A^\star} \vdash [\![s^\star]\!]_{\mathsf{FIV}([\![\vec{u}]\!]_{\vec{\Xi}}\Gamma) \cap \mathsf{FIV}(B^\star)} B^\star. \\
(\lambda x^A.s)^\star & \triangleq & c^{\mathbf{A1c}} \cdot c^{\mathbf{A1c}}, \quad \text{otherwise.} \\
(\mu\alpha^A.s)^\star & \triangleq & t_\mu^{A^\star}, \quad \text{if there exists a FOLP-context } [\![\vec{u}]\!]_{\vec{\Xi}}\Gamma \text{ and a fresh } y \text{ s.t.} \\
 & & \rhd_{\mathsf{FOLP}} [\![\vec{u}]\!]_{\vec{\Xi}}\Gamma, y^{[\![\alpha^{\neg A^\star}]\!]_{\mathsf{FIV}(A^\star)} A^\star} \vdash [\![s^\star]\!]_\emptyset \bot. \\
(\mu\alpha^A.s)^\star & \triangleq & c^{\mathbf{A1c}} \cdot c^{\mathbf{A1c}}, \quad \text{otherwise.}
\end{array}$$

Here **A1c** refers to the axiom scheme of classical logic $\neg\neg A \supset A$. In our use of this translation (Prop. 5.2.21) the conditions of the first clause and third clauses shall be met when dealing with modalities that are introduced using $\Box\mathsf{I}$ and in which the translated abstraction that occurs in the internalized proof witness is proved in π itself; the second and fourth other cases are used

when these abstractions that occur in modalities do not represent valid proofs[1]. The proof terms defined by the first and third clauses all depend on the form that the assumed FOLP derivation takes. Since there are non-linear constraints in our terms (*cf.* $\supset\mathsf{E}$ in Fig. 5.1 has A in positive and negative positions), hence the reason for the assumption that π be canonical.

Translating bang. Regarding the clause for $(!t)^\star$, defining it simply as $!t^\star$ presents technical difficulties when addressing the case $\Box\mathsf{I}'$. Indeed, the IH yields:

$$\Theta^\star \vdash [\![t^\star]\!]_{\mathsf{FIV}(\Theta^\star)\cap\mathsf{FIV}(A^\star)} A^\star$$

from which we can obtain the following, given the condition $\mathsf{FIV}(\Theta)\cap\mathsf{FIV}(A)\subseteq\Xi$ of $\Box\mathsf{I}'$:

$$\Theta^\star \vdash [\![t^\star]\!]_{\Xi} A^\star$$

But then **B4** yields:

$$\Theta^\star \vdash [\![!t^\star]\!]_{\Xi}[\![t^\star]\!]_{\Xi} A^\star$$

Here we are stuck since it is not sound to simply discard the variables in Ξ in order to obtain $\mathsf{FIV}(\mathcal{H}^\star)\cap\Xi$. A similar situation arises if we define $t\langle v^A_\Xi{:=}r,s\rangle$ as $t^\star\{v^{A^\star}\leftarrow r^\star\}$. We thus define $!t^\star$ and $t\langle v^A_\Xi{:=}r,s\rangle$ as follows:

$$\begin{array}{rcll}
(!t)^\star & \triangleq & t_!^{\Xi,A^\star}, & \text{if there exists a } \mathsf{FOLP}\text{-context } [\![\vec{u}]\!]_{\vec{\Xi}}\Gamma \text{ and a formula } A \text{ s.t.} \\
 & & & \rhd_{\mathsf{FOLP}} [\![\vec{u}]\!]_{\vec{\Xi}}\Gamma \vdash [\![t^\star]\!]_{\mathsf{FIV}([\![\vec{u}]\!]_{\vec{\Xi}}\Gamma)\cap\mathsf{FIV}(A^\star)} A^\star \text{ and} \\
 & & & \mathsf{FIV}([\![\vec{u}]\!]_{\vec{\Xi}}\Gamma)\cap\mathsf{FIV}(A^\star)\subseteq\Xi. \\
(!t)^\star & \triangleq & !t^\star, & \text{otherwise.} \\
(t\langle v^A_\Xi{:=}r,s\rangle)^\star & \triangleq & r^{\vec{\Xi},C^\star\{v^{A^\star}\leftarrow r^\star\}}, & \text{if there exists a } \mathsf{FOLP}\text{-context } [\![\vec{u}]\!]_{\vec{\Xi}}\Gamma \text{ and a formula } C \text{ s.t.} \\
 & & & \rhd_{\mathsf{FOLP}} [\![\vec{u}]\!]_{\vec{\Xi}}\Gamma \vdash [\![t^\star\{v^{A^\star}\leftarrow r^\star\}]\!]_{(\mathsf{FIV}([\![\vec{u}]\!]_{\vec{\Xi}}\Gamma)\cup\Xi)} C^\star\{v^{A^\star}\leftarrow r^\star\}. \\
(t\langle v^A_\Xi{:=}r,s\rangle)^\star & \triangleq & t^\star\{v^{A^\star}\leftarrow r^\star\}, & \text{otherwise.}
\end{array}$$

Proposition 5.2.21 If $\rhd_{\mathsf{FOHLP}}\mathcal{H}\vdash D\,|\,s$, then $\rhd_{\mathsf{FOLP}}\mathcal{H}^\star\vdash [\![s^\star]\!]_{\mathsf{FIV}(\mathcal{H}^\star)\cap\mathsf{FIV}(D^\star)} D^\star$, where $\mathcal{H}^\star$ is shorthand for $\Theta^\star\cup\Gamma^\star\cup\Delta^\star$.

Proof.- By induction on the derivation of $\mathcal{H}\vdash D\,|\,s$. We assume that the derivation does not resort to proof witness equivalence (*cf.* Rem. 5.2.15). We analyze the last rule used.

- Case Var. $\mathcal{H}\vdash D\,|\,s$ is $\Theta;\Gamma',x^A;\Delta\vdash A\,|\,x^A$. Trivially $\rhd_{\mathsf{FOLP}}\Theta^\star\cup\Gamma^\star\cup\Delta^\star\vdash [\![x^{A^\star}]\!]_{\mathsf{FIV}(A^\star)} A^\star$. Since $[\![x^{A^\star}]\!]_{\mathsf{FIV}(A^\star)} A^\star\in\Gamma^\star$, then $\mathsf{FIV}(\mathcal{H}^\star)\cap\mathsf{FIV}(A^\star)=\mathsf{FIV}(A^\star)$.
- Case VarM. In this case $s=v^D_\Xi$, $\Theta=\Theta',v^D_\Xi$ and hence $\mathcal{H}=\Theta',v^D_\Xi;\Gamma;\Delta$. $\mathcal{H}^\star\vdash [\![s^\star]\!]_{\Xi} D^\star$ is derivable since $v^{D^\star}\in\mathcal{H}^\star$. Note also that $\mathsf{FIV}(\mathcal{H}^\star)\cap\mathsf{FIV}(D^\star)=\mathsf{FIV}(D^\star)$. We can use **A2** and **A3** as necessary (along with **MP**) to derive $\mathcal{H}^\star\vdash [\![s^\star]\!]_{\mathsf{FIV}(D^\star)} D^\star$.
- Case $\supset\mathsf{I}$. The derivation ends in:

[1] Eg. $\cdot;x^{[\![\lambda z^A.z\cdot z]\!]_{\Xi} B};\cdot\vdash [\![\lambda z^A.z\cdot z]\!]_{\Xi} B\,|\,x^{[\![\lambda z^A.z\cdot z]\!]_{\Xi} B}$.

$$\frac{\Theta; \Gamma, x^A; \Delta \vdash B \mid t}{\Theta; \Gamma; \Delta \vdash A \supset B \mid \lambda x^A.t} \supset \mathsf{I}$$

By the IH we can derive:

$$\Theta^\star \cup \Gamma^\star, [\![x^{A^\star}]\!]_{\mathsf{FIV}(A)} A^\star \cup \Delta^\star \vdash [\![s^\star]\!]_{\mathsf{FIV}(\mathcal{H}^\star) \cap \mathsf{FIV}(B^\star)} B^\star \tag{5.2}$$

Thus, our translation requires that we prove:

$$\mathcal{H}^\star \vdash [\![t_\lambda^{A^\star \supset B^\star}]\!]_{\mathsf{FIV}(\mathcal{H}^\star) \cap \mathsf{FIV}(A^\star \supset B^\star)} (A^\star \supset B^\star) \tag{5.3}$$

in FOLP in order to conclude the case. For that we obtain $\Theta^\star \cup \Gamma^\star, [\![x^{A^\star}]\!]_{\mathsf{FIV}(A)} A^\star \cup \Delta^\star \vdash [\![s^\star]\!]_{\mathsf{FIV}(B^\star)} B^\star$ from (5.2) and possibly multiple uses of **A2**, **A3** and **MP**. We then resort to the λ-Abstraction Lemma (5.1.11) to obtain $t_\lambda^{A^\star \supset B^\star}$ s.t. $\mathcal{H}^\star \vdash [\![t_\lambda^{A^\star \supset B^\star}]\!]_{\mathsf{FIV}(\mathcal{H}^\star) \cap \mathsf{FIV}(A^\star \supset B^\star)} (A^\star \supset B^\star)$ is derivable in FOLP.

- Case $\supset\mathsf{E}$. The derivation ends in:

$$\frac{\Theta; \Gamma; \Delta \vdash A \supset B \mid s \quad \Theta; \Gamma; \Delta \vdash A \mid t}{\Theta; \Gamma; \Delta \vdash B \mid s \cdot t} \supset \mathsf{E}$$

By the IH both of the following judgements are derivable in FOLP:

1. $\mathcal{H}^\star \vdash [\![s^\star]\!]_{\mathsf{FIV}(\mathcal{H}^\star) \cap \mathsf{FIV}(A^\star \supset B^\star)} (A \supset B)^\star$ and
2. $\mathcal{H}^\star \vdash [\![t^\star]\!]_{\mathsf{FIV}(\mathcal{H}^\star) \cap \mathsf{FIV}(A^\star)} A^\star$

We can derive $\mathcal{H}^\star \vdash [\![t^\star]\!]_{\mathsf{FIV}(\mathcal{H}^\star) \cap \mathsf{FIV}(A^\star \supset B^\star)} A^\star$ by using **A3** and **MP** as many times as required (keep in mind that $\mathsf{FIV}(A) = \mathsf{FIV}(A^\star) \subseteq \mathsf{FIV}(A^\star \supset B^\star)$). Then, using **B2** and **MP** twice, we derive $\mathcal{H}^\star \vdash [\![(s^\star \cdot t^\star)]\!]_{\mathsf{FIV}(\mathcal{H}^\star) \cap \mathsf{FIV}(A^\star \supset B^\star)} B^\star$.

Note that $A^\star$ is the same on both sides, since we are assuming canonical derivations and thus translations are unique.

- Case $\Box\mathsf{I}'$. In this case $D = [\![t]\!]_\Xi A$ and the derivation ends in:

$$\frac{\Theta; \cdot; \cdot \vdash A \mid t \quad \mathsf{FIV}(\Theta) \cap \mathsf{FIV}(A) \subseteq \Xi}{\Theta; \Gamma; \Delta \vdash [\![t]\!]_\Xi A \mid !t} \Box\mathsf{I}'$$

We reason as follows, where * in step (c) means **A3** and **MP** are used possibly multiple times:

(a)	$\Theta^\star \vdash [\![t^\star]\!]_{\mathsf{FIV}(\Theta^\star) \cap \mathsf{FIV}(A^\star)} A^\star$	(IH)
(b)	$\Theta^\star \vdash [\![t_!^{\Xi, A^\star}]\!]_{\mathsf{FIV}(\Theta^\star) \cap \Xi} [\![t]\!]_\Xi A^\star$	(Lem. 5.1.13, $\mathsf{FIV}(\Theta^\star) \cap \mathsf{FIV}(A^\star) \subseteq \Xi$)
(c)	$\Theta^\star \cup \Gamma^\star \cup \Delta^\star \vdash [\![t_!^{\Xi, A}]\!]_{\mathsf{FIV}(\mathcal{H}^\star) \cap \Xi} [\![t^\star]\!]_\Xi A^\star$	(**A3** and **MP**)*

We know that $t_!^{\Xi, A}$ is the correct for translation for $!t$, since $\rhd_{\mathsf{FOLP}} \Theta^\star \vdash [\![t^\star]\!]_{\mathsf{FIV}(\Theta^\star) \cap \mathsf{FIV}(A)} A^\star$ and $\mathsf{FIV}(\Theta^\star) \cap \mathsf{FIV}(A) \subseteq \Xi$.

- Case $\Box\mathsf{E}$. The derivation ends in:

$$\frac{\Theta; \Gamma; \Delta \vdash [\![r]\!]_\Xi A \mid s' \quad \Theta, v_{\Xi'}^A; \Gamma; \Delta \vdash C \mid t \quad \Xi \cap \mathsf{FIV}(A) \subseteq \Xi'}{\Theta; \Gamma; \Delta \vdash C\{v_{\Xi'}^A \leftarrow r\} \mid t\langle v_{\Xi'}^A := r, s'\rangle} \Box\mathsf{E}$$

with $D = C\{v^A_{\Xi'} \leftarrow r\}$ and $s = t\langle v^A_{\Xi'} := r, s'\rangle$.

By the IH both of the following judgements are derivable in FOLP:

1. $\mathcal{H}^\star \vdash [\![s'^\star]\!]_{\mathsf{FIV}(\mathcal{H}^\star)\cap\Xi}[\![r^\star]\!]_\Xi A^\star$
2. $\mathcal{H}^\star, [\![v^{A^\star}]\!]_{\Xi'} A^\star \vdash [\![t^\star]\!]_{(\mathsf{FIV}(\mathcal{H}^\star)\cup\Xi')\cap\mathsf{FIV}(C^\star)} C^\star$

We now reason as follows:

(a)	$\mathcal{H}^\star \vdash ([\![s'^\star]\!]_{\mathsf{FIV}(\mathcal{H}^\star)\cap\Xi}[\![r^\star]\!]_\Xi A^\star) \supset ([\![r^\star]\!]_\Xi A^\star)$	(**B1**)
(b)	$\mathcal{H}^\star \vdash [\![r^\star]\!]_\Xi A^\star$	((a), **MP**)
(c)	$\mathcal{H}^\star \vdash [\![t^\star\{v^{A^\star} \leftarrow r^\star\}]\!]_{((\mathsf{FIV}(\mathcal{H}^\star)\cup\Xi')\cap\mathsf{FIV}(C^\star))} C^\star\{v^{A^\star} \leftarrow r^\star\}$	(Lem. 5.1.15, (b), (2))
(d)	$\mathcal{H}^\star \vdash C^\star\{v^{A^\star} \leftarrow r^\star\}$	(**B1**, **MP**)
(e)	$\mathcal{H}^\star \vdash [\![r^{\mathsf{FIV}(\mathcal{H}^\star), C^\star\{v^{A^\star} \leftarrow r^\star\}}]\!]_{(\mathsf{FIV}(\mathcal{H}^\star)\cap\mathsf{FIV}(C^\star))} C^\star\{v^{A^\star} \leftarrow r^\star\}$	(Cor. 5.1.9)

By Remark 5.1.4, $\mathsf{FIV}(C^\star) = \mathsf{FIV}(C^\star\{v^{A^\star} \leftarrow r^\star\})$.

We know that $r^{\mathsf{FIV}(\mathcal{H}^\star), C^\star\{v^{A^\star} \leftarrow r^\star\}}$ is the correct translation for $t\langle v^A_{\Xi'} := r, s'\rangle$, since $\mathcal{H}^\star \vdash [\![t^\star\{v^{A^\star} \leftarrow r^\star\}]\!]_{((\mathsf{FIV}(\mathcal{H}^\star)\cup\Xi')\cap\mathsf{FIV}(C^\star))} C^\star\{v^{A^\star} \leftarrow r^\star\}$ is derivable in FOLP by (c), and therefore so is $\mathcal{H}^\star \vdash [\![t^\star\{v^{A^\star} \leftarrow r^\star\}]\!]_{\mathsf{FIV}(\mathcal{H}^\star)\cup\Xi'} C^\star\{v^{A^\star} \leftarrow r^\star\}$.

- Case PlusL (PlusR is similar and hence ommitted). The derivation ends in:

$$\frac{\Theta; \Gamma; \Delta \vdash A \mid s}{\Theta; \Gamma; \Delta \vdash A \mid s + t}\ \mathsf{PlusL}$$

By IH $\rhd_{\mathsf{FOLP}} \mathcal{H}^\star \vdash [\![s^\star]\!]_{\mathsf{FIV}(\mathcal{H}^\star)\cap\mathsf{FIV}(A^\star)} A^\star$. Thus, by **B3a** and **MP**, also $\rhd_{\mathsf{FOLP}} \mathcal{H}^\star \vdash [\![s^\star + t^\star]\!]_{\mathsf{FIV}(\mathcal{H}^\star)\cap\mathsf{FIV}(A^\star)} A^\star$.

- Case NAbs. The derivation ends in:

$$\frac{\Theta; \Gamma; \Delta, \alpha^A \vdash \bot \mid s}{\Theta; \Gamma; \Delta \vdash A \mid \mu\alpha^A.s}\ \mathsf{NAbs}$$

By IH, $\rhd_{\mathsf{FOLP}} \mathcal{H}^\star, [\![\alpha^{\neg A^\star}]\!]_{\mathsf{FIV}(A^\star)} A^\star \vdash [\![s^\star]\!]_\emptyset \bot$, and thus $\mu\alpha^A.s^\star = t^{A^\star}_\mu = c^{\mathbf{A1c}}_{A^\star} \cdot t^{\neg\neg A}_\lambda$. By the μ-Abstraction Corollary (5.1.12), $\mathcal{H}^\star \vdash [\![t^{A^\star}_\mu]\!]_{\mathsf{FIV}(\mathcal{H}^\star)\cap\mathsf{FIV}(A^\star)} A^\star$ is derivable in FOLP.

- Case Name. The derivation ends in:

$$\frac{\Theta; \Gamma; \Delta, \alpha^A \vdash A \mid s}{\Theta; \Gamma; \Delta, \alpha^A \vdash \bot \mid [\alpha^A]s}\ \mathsf{Name}$$

By IH, $\rhd_{\mathsf{FOLP}} \mathcal{H}^\star, x^{[\![\alpha^{\neg A^\star}]\!]_{\mathsf{FIV}(A^\star)}\neg A^\star} \vdash [\![s^\star]\!]_{\mathsf{FIV}(\mathcal{H}^\star)\cap\mathsf{FIV}(A^\star)} A^\star$.

Note that $\mathsf{FIV}(\mathcal{H}^\star, x^{[\![\alpha^{\neg A^\star}]\!]_{\mathsf{FIV}(A^\star)}\neg A^\star}) \cap \mathsf{FIV}(A^\star) = \mathsf{FIV}(A^\star)$.

We reason as follows:

(1)	$\mathcal{H}^\star, x^{[\![\alpha^{\neg A^\star}]\!]_{\mathsf{FIV}(A^\star)}\neg A^\star} \vdash [\![\alpha^{\neg A^\star}]\!]_{\mathsf{FIV}(A^\star)}\neg A^\star$	(hypothesis $x^{[\![\alpha^{\neg A^\star}]\!]_{\mathsf{FIV}(A^\star)}\neg A^\star}$)
(2)	$\mathcal{H}^\star, x^{[\![\alpha^{\neg A^\star}]\!]_{\mathsf{FIV}(A^\star)}\neg A^\star} \vdash [\![s^\star]\!]_{\mathsf{FIV}(A^\star)} A^\star$	(IH)
(3)	$\mathcal{H}^\star, x^{[\![\alpha^{\neg A^\star}]\!]_{\mathsf{FIV}(A^\star)}\neg A^\star} \vdash [\![\alpha^{\neg A^\star} \cdot s^\star]\!]_{\mathsf{FIV}(A^\star)} \bot$	(**B2** and **MP** twice)
(4)	$\mathcal{H}^\star, x^{[\![\alpha^{\neg A^\star}]\!]_{\mathsf{FIV}(A^\star)}\neg A^\star} \vdash [\![\alpha^{\neg A^\star} \cdot s^\star]\!] \bot$	(**A2** and **MP** as required)

- Case $\forall\mathsf{I}$. The derivation ends in:

$$\frac{\Theta;\Gamma;\Delta \vdash A \,|\, t \quad i \notin \mathsf{FIV}(\Theta,\Gamma,\Delta)}{\Theta;\Gamma;\Delta \vdash \forall i.A \,|\, \mathsf{gen}_i(t)}\,\forall\mathsf{I}$$

with $D = \forall i.A$ and $s = \mathsf{gen}_i(t)$. By the IH, $\rhd_{\mathsf{FOLP}} \mathcal{H}^\star \vdash [\![t^\star]\!]_{\mathsf{FIV}(\mathcal{H}^\star)\cap\mathsf{FIV}(A^\star)} A^\star$. Since $i \notin \mathsf{FIV}(\mathcal{H})$, then $i \notin \mathsf{FIV}(\mathcal{H}^\star)$, and thus we can obtain the result by **B5**, **A2**, and **MP** (twice).

- Case $\forall\mathsf{E}$. The derivation ends in:

$$\frac{\Theta;\Gamma;\Delta \vdash \forall i.A \,|\, t}{\Theta;\Gamma;\Delta \vdash A\{i \leftarrow E\} \,|\, \mathsf{ins}_i^E(t)}\,\forall\mathsf{E}$$

with $D = A\{i \leftarrow E\}$ and $s = \mathsf{ins}_i^E(t)$. Let $\Xi = \mathsf{FIV}(\mathcal{H}^\star) \cap \mathsf{FIV}(\forall i.A^\star)$ and $\Xi' = \mathsf{FIV}(\mathcal{H}^\star) \cap \mathsf{FIV}(A^\star\{i \leftarrow E\})$. By the IH, $\rhd_{\mathsf{FOLP}} \mathcal{H}^\star \vdash [\![t^\star]\!]_\Xi \forall i.A^\star$.

Since $\mathsf{FIV}(\forall i.A^\star) \subseteq \mathsf{FIV}(A^\star\{i \leftarrow E\})$, then $\rhd_{\mathsf{FOLP}} \mathcal{H}^\star \vdash [\![t^\star]\!]_{\Xi'} \forall i.A^\star$ by using **A3** and **MP** as many times as required.

We reason as follows, where * indicates multiple uses of an axiom:

(a) $\mathcal{H}^\star \vdash \forall i.A^\star \supset A^\star\{i \leftarrow E\}$ (**A1d**)
(b) $\mathcal{H}^\star \vdash [\![c^{\mathbf{A1d}}_{A^\star,i,E}]\!] \forall i.A^\star \supset A^\star\{i \leftarrow E\}$ (**Nec**)
(c) $\mathcal{H}^\star \vdash [\![c^{\mathbf{A1d}}_{A^\star,i,E}]\!]_{\Xi'} \forall i.A^\star \supset A^\star\{i \leftarrow E\}$ (**A3***,**MP***)
(d) $\mathcal{H}^\star \vdash [\![c^{\mathbf{A1d}}]\!]_{\Xi'} \forall i.A^\star \supset A^\star\{i \leftarrow E\} \supset [\![t^\star]\!]_{\Xi'} \forall i.A^\star \supset [\![(c^{\mathbf{A1d}} \cdot t^\star)]\!]_{\Xi'} A^\star\{i \leftarrow E\}$ (**B2**)
(e) $\mathcal{H}^\star \vdash [\![t^\star]\!]_{\Xi'} \forall i.A^\star \supset [\![(c^{\mathbf{A1d}} \cdot t^\star)]\!]_{\Xi'} A^\star\{i \leftarrow E\}$ (**MP**)
(f) $\mathcal{H}^\star \vdash [\![c^{\mathbf{A1d}} \cdot t^\star]\!]_{\Xi'} A^\star\{i \leftarrow E\}$ (**MP**)

□

Corollary 5.2.22 If $\rhd_{\mathsf{FOHLP}} \cdot;\cdot;\cdot \vdash A \,|\, s$, then $\rhd_{\mathsf{FOLP}} \cdot \vdash [\![s^\star]\!]_{\mathsf{FIV}(A)} A^\star$ and $\rhd_{\mathsf{FOLP}} \cdot \vdash A^\star$.

5.3 Term Assignment

We now present the terms of the λ^{FOLP}-calculus, the term assignment for FOHLP we shall study.

Definition 5.3.1 (Terms of λ^{FOLP}) The terms of λ^{FOLP} are given by the following grammar:

$$
\begin{array}{rcl}
M, N & ::= & x^A \\
& | & v^A_{\Xi} \\
& | & (\lambda x^A.M^B)^{A \supset B} \\
& | & (M^{A \supset B} N^A)^B \\
& | & (!M^A)^{[\![s]\!]_{\Xi} A} \\
& | & (M^B \langle v^A_{\Xi'} := r, N^{[\![r]\!]_{\Xi} A} \rangle)^{B\{v^A_{\Xi'} \leftarrow r\}} \\
& | & ([\alpha^A] M^A)^{\perp} \\
& | & (\mu \alpha^A.M^{\perp})^A \\
& | & (M^A +_{\mathrm{L}} s)^A \\
& | & (s +_{\mathrm{R}} N^B)^B \\
& | & (\mathsf{gen}_i(M^A))^{\forall i.A} \\
& | & (\mathsf{ins}^E_i(M^{\forall i.A}))^{A\{i \leftarrow E\}}
\end{array}
$$

Terms are proof witnesses decorated with annotations which provide additional information about a derivation. The connection between λ^{FOLP}-terms and FOHLP-derivations is the same as the connection between λ^{LP}-terms and HLP-derivations. As in λ^{LP}, the term $(M^A +_{\mathrm{L}} s)^A$ encodes a proof of A which appends the witness s to a previous proof of A – encoded by M^A – by using PlusL. Analogously, $(s +_{\mathrm{R}} M^A)^A$ encodes the proof which results from appending s to a proof of A by PlusR. Also like in λ^{LP}, but now taking individual variables into account, the terms $(!v^A_{\Xi})^{[\![v^A_{\Xi}]\!]_{\Xi} A}$, $(!v^A_{\Xi})^{[\![(\lambda x^A.x^A) \cdot v^A_{\Xi}]\!]_{\Xi} A}$ and $(!v^A_{\Xi})^{[\![v^A_{\Xi}]\!]_{\Xi \cup \Xi'} A}$ encode different derivations, in which $\Box$I is used in different ways to prove different formulas.

Again, our terms do not encode the equivalence rules used to derive the second premise of T-$\Box$I, nor the contexts used in the derivations.

Free variables of validity, truth and falsehood, as well as free individual variables over terms are defined analogously to those for proof witnesses, and the notational conventions extend to terms as expected. As in λ^{LP}, decorations may be omitted where it is safe. We will use the letters M, N, L – with or without superindices – to refer to terms.

Definition 5.3.2 (Typing rules for λ^{FOLP}) Typing judgements in λ^{FOLP} take the form $\Theta; \Gamma; \Delta \vdash M^A \,|\, s$. The typing rules that define which such judgements are derivable are given in Fig. 5.4. They arise from the inference schemes of FOHLP of Fig. 5.1. In particular, note that if $\Theta; \Gamma; \Delta \vdash M^A \,|\, s$ is derivable with the typing rules of λ^{FOLP}, then $\Theta; \Gamma; \Delta \vdash A \,|\, s$ is derivable in FOHLP. We write $\mathcal{H} \vdash M^A \,|\, s$ to abbreviate $\Theta; \Gamma; \Delta \vdash M^A \,|\, s$.

As in the propositional case, there is a correspondence between the typing rules rules and the deduction rules for FOHLP (removing all terms from the typing rules results in the original inference schemes). Also as in λ^{LP} – and for the same reasons – every well-typed term encodes a derivation in FOHLP modulo the equivalence rules, and every FOHLP-derivation can be encoded by a term.

$$\frac{}{\mathcal{H}, x^A \vdash x^A \,|\, x^A}\ \textsf{T-Var} \qquad \frac{}{\mathcal{H}, v^A_\Xi \vdash v^A_\Xi \,|\, v^A_\Xi}\ \textsf{T-VarM}$$

$$\frac{\mathcal{H}, x^A \vdash M^B \,|\, s}{\mathcal{H} \vdash (\lambda x^A.M^B)^{A \supset B} \,|\, \lambda x^A.s}\ \textsf{T-}\supset\textsf{I} \qquad \frac{\mathcal{H} \vdash M^{A \supset B} | s \quad \mathcal{H} \vdash N^A \,|\, t}{\mathcal{H} \vdash (M^{A \supset B} N^A)^B \,|\, s \cdot t}\ \textsf{T-}\supset\textsf{E}$$

$$\frac{\Theta; \cdot; \cdot \vdash M^A \,|\, s \quad \Theta; \cdot; \cdot \vdash\ s \equiv t : A \quad \textsf{FIV}(\Theta) \cap \textsf{FIV}(A) \subseteq \Xi}{\Theta; \Gamma; \Delta \vdash (!M^A)^{[\![t]\!]_{\equiv A}} \,|\, !t}\ \textsf{T-}\Box\textsf{I}$$

$$\frac{\mathcal{H} \vdash M^{[\![r]\!]_{\equiv A}} \,|\, s \quad \mathcal{H}, v^A_{\Xi'} \vdash N^C \,|\, t \quad \Xi \cap \textsf{FIV}(A) \subseteq \Xi'}{\mathcal{H} \vdash N^C \langle v^A_{\Xi'} := r, M^{[\![r]\!]_{\equiv A}} \rangle^{C\{v^A_{\Xi'} \leftarrow r\}} \,|\, t \langle v^A_{\Xi'} := r, s \rangle}\ \textsf{T-}\Box\textsf{E}$$

$$\frac{\mathcal{H} \vdash M^A \,|\, s}{\mathcal{H} \vdash (M +_{\mathrm{L}} t)^A \,|\, s + t}\ \textsf{T-PlusL} \qquad \frac{\mathcal{H} \vdash N^B \,|\, t}{\mathcal{H} \vdash (s +_{\mathrm{R}} N)^B \,|\, s + t}\ \textsf{T-PlusR}$$

$$\frac{\mathcal{H}, \alpha^A \vdash M^A \,|\, s}{\mathcal{H}, \alpha^A \vdash ([\alpha^A]M)^\perp \,|\, [\alpha^A]s}\ \textsf{T-Name} \qquad \frac{\mathcal{H}, \alpha^A \vdash M^\perp \,|\, s}{\mathcal{H} \vdash (\mu\alpha^A.M)^A \,|\, \mu\alpha^A.s}\ \textsf{T-NAbs}$$

$$\frac{\mathcal{H} \vdash M^A | s \quad i \notin \textsf{FIV}(\mathcal{H})}{\mathcal{H} \vdash \textsf{gen}_i(M)^{\forall i.A} \,|\, \textsf{gen}_i(s)}\ \textsf{T-}\forall\textsf{I} \qquad \frac{\mathcal{H} \vdash M^{\forall i.A} \,|\, s}{\mathcal{H} \vdash \textsf{ins}^E_i(M)^{A\{i \leftarrow E\}} \,|\, \textsf{ins}^E_i(s)}\ \textsf{T-}\forall\textsf{E}$$

Figure 5.4: Typing rules for $\lambda^{\textsf{FOLP}}$

Remark 5.3.3 If $\Theta; \Gamma; \Delta \vdash M^A \,|\, s$ is derivable in $\lambda^{\textsf{FOLP}}$, then there is a derivation of $\Theta; \Gamma; \Delta \vdash M^A \,|\, s$ which does not make use of the equivalence rules. In fact, we can use the same derivation from Remark 5.2.15, adding the respective terms to encode each step of the derivation.

Remark 5.3.4 The notion of proof witness associated to a term is extended to the new terms and proof witnesses as expected, with $\textsf{w}(M^A)$ being the only proof witness s such that the judgement $\Theta; \Gamma; \Delta \vdash M^A \,|\, s$ is derivable for some Θ, Γ, Δ.

Lemma 5.3.5 (Individual Variable Strengthening) If $\Theta; \Gamma; \Delta \vdash M^D \,|\, r$ is derivable, then there exist Θ', Γ', Δ' s.t. $\Theta'; \Gamma'; \Delta' \vdash M^D \,|\, r$ is derivable, $\Theta' \subseteq \Theta$, $\Gamma' \subseteq \Gamma$, $\Delta' \subseteq \Delta$ and $\textsf{FIV}(\Theta', \Gamma', \Delta') \subseteq \textsf{FIV}(r)$.

Proof.- By induction on the derivation of $\Theta; \Gamma; \Delta \vdash M^D \,|\, r$. For simplicity, we assume that the derivation does not make use of proof witness equivalence. Such a derivation exists by Remark 5.3.3. We analyze the last rule used.

- Case T-Var: $M^D = x^D$, $x^D \in \Gamma$, and $\cdot; x^D; \cdot \vdash x^D \,|\, x^D$ holds by Var.
- Case T-VarM: $M^D = v^D_\Xi$, $v^D_\Xi \in \Theta$ and $v^D_\Xi; \cdot; \cdot \vdash v^A_\Xi \,|\, v^A_\Xi$ holds by VarM.

- Case T-Name: $M^D = ([\alpha^A]M'^A)^\perp$, $r = [\alpha^A]s$, $D = \perp$ $\Delta = \Delta_1, \alpha^A$ and $\Theta; \Gamma; \Delta_1, \alpha^A \vdash M'^A \,|\, s$ is derivable by hypothesis. Note that $\mathsf{FIV}(A) \subseteq \mathsf{FIV}([\alpha^A]s)$ by definition, so $\Delta' = \Delta'_1, \alpha^A$. We can derive $\Theta'; \Gamma'; \Delta'_1, \alpha^A \vdash M'^A \,|\, s$ by IH (and Weakening if $\mathsf{FIV}(A) \not\subseteq \mathsf{FIV}(s)$), and then obtain the result by Name.
- Case T- $\supset$ I: $M^D = (\lambda x^A.M')^{A \supset B}$, $r = \lambda x^A.s$ $D = A \supset B$ and $\Theta; \Gamma, x^A; \Delta \vdash M'^B \,|\, s$ is derivable by hypothesis. By IH (and Weakening in the case that $\mathsf{FIV}(A) \not\subseteq \mathsf{FIV}(s)$), we can derive $\Theta'; \Gamma', x^A; \Delta' \vdash M'^B \,|\, s$. The result holds by $\supset$I.
- Case T-NAbs: this case is analogous to the previous one.
- Case T- $\supset$ E: $M^D = M_1^{A \supset D} M_2^A$, $r = s \cdot t$ and both $\Theta; \Gamma; \Delta \vdash M_1^{A \supset D} \,|\, s$ and $\Theta; \Gamma; \Delta \vdash M_2^A \,|\, t$ are derivable by hypothesis. By IH, we can derive $\Theta_1; \Gamma_1; \Delta_1 \vdash M_1^{A \supset D} \,|\, s$ and $\Theta_2; \Gamma_2; \Delta_2 \vdash M_2^A \,|\, t$, where $\Theta_1, \Theta_2 \subseteq \Theta$, $\Gamma_1, \Gamma_2 \subseteq \Gamma$, $\Delta_1, \Delta_2 \subseteq \Delta$, $\mathsf{FIV}(\Theta_1, \Gamma_1, \Delta_1) \subseteq \mathsf{FIV}(s)$ and $\mathsf{FIV}(\Theta_2, \Gamma_2, \Delta_2) \subseteq \mathsf{FIV}(t)$. Let $\Theta' = \Theta_1 \cup \Theta_2$, $\Gamma' = \Gamma_1 \cup \Gamma_2$, $\Delta' = \Delta_1 \cup \Delta_2$. We know that $\Theta' \subseteq \Theta$, $\Gamma' \subseteq \Gamma$, $\Delta' \subseteq \Delta$ and $\mathsf{FIV}(\Theta', \Gamma', \Delta') \subseteq \mathsf{FIV}(s) \cup \mathsf{FIV}(t) = \mathsf{FIV}(s \cdot t)$. We can derive both $\Theta'; \Gamma'; \Delta' \vdash M_1^{A \supset D} \,|\, s$ and $\Theta'; \Gamma'; \Delta' \vdash M_2^A \,|\, t$ by Weakening, and then obtain the result by $\supset$E.
- Case T-$\Box$I': $M^D = (!M')^{[\![t]\!]_A \Xi}$, $D = [\![t]\!]_A \Xi$, $r = !t$ and, by hypothesis, $\Theta; \cdot; \cdot \vdash M'^A \,|\, t$ is derivable and $\mathsf{FIV}(\Theta) \cap \mathsf{FIV}(A) \subseteq \Xi$. By IH, we can derive $\Theta'; \cdot; \cdot \vdash M'^A \,|\, t$. Note that, since $\Theta' \subseteq \Theta$, then $\mathsf{FIV}(\Theta') \cap \mathsf{FIV}(A) \subseteq \Xi$. We can obtain $\Theta'; \cdot; \cdot \vdash !M'^{[\![t]\!]_A \Xi} \,|\, t$ by T-$\Box$I' (or T-$\Box$I and Eq-Refl).
- Case T-$\Box$E: $M = (N^C \langle v^A_{\Xi'} := r', M'^{[\![r']\!]_{\Xi} A} \rangle)^{C\{v^A_{\Xi'} \leftarrow r\}}$, $D = C\{v^A_{\Xi'} \leftarrow r\}$, $r = t\langle v^A_{\Xi'} := r', s \rangle$ and, by hypothesis, we can derive both $\Theta; \Gamma; \Delta \vdash M'^{[\![r']\!]_{\Xi} A} \,|\, s$ and $\Theta, v^A_{\Xi'}; \Gamma; \Delta \vdash N^C \,|\, t$, and $\Xi \cap \mathsf{FIV}(A) \subseteq \Xi'$.

  By IH, we can derive both $\Theta_1; \Gamma_1; \Delta_1 \vdash M'^{[\![r']\!]_{\Xi} A} \,|\, s$ and $\Theta_2, v^A_{\Xi'}; \Gamma_2; \Delta_2 \vdash N^C \,|\, t$, with the same conditions as in the $\supset$ E case (using Weakening for the second judgement if $\Xi \not\subseteq \mathsf{FIV}(t)$). We can take $\Theta' = \Theta_1 \cup \Theta_2$, $\Gamma' = \Gamma_1 \cup \Gamma_2$, $\Delta' = \Delta_1 \cup \Delta_2$. By Weakening, we can derive $\Theta'; \Gamma'; \Delta' \vdash M'^{[\![r']\!]_{\Xi} A} \,|\, s$ and $\Theta', v^A_{\Xi'}; \Gamma'; \Delta' \vdash N^C \,|\, t$. Since neither Ξ nor Ξ' has changed, we can obtain the result by $\Box$E.
- Case T-PlusL: the result holds by IH and PlusL.
- Case T-PlusR: the result holds by IH and PlusR.
- Case T-$\forall$I:$M^D = (\mathsf{gen}_i(M'^A))^{\forall i.A}$, $D = \forall i.A$, $r = \mathsf{gen}_i(s)$ and, by hypothesis, $i \notin \mathsf{FIV}(\Theta, \Gamma, \Delta)$ and $\Theta; \Gamma; \Delta \vdash M'^A \,|\, s$ is derivable. By IH, we can derive $\Theta'; \Gamma'; \Delta' \vdash M'^A \,|\, s$. We know that $i \notin \mathsf{FIV}(\Theta', \Gamma', \Delta')$, since $\Theta' \subseteq \Theta$, $\Gamma' \subseteq \Gamma$ and $\Delta' \subseteq \Delta$. We can obtain the result by $\forall$I.
- Case T-$\forall$E: $M = (\mathsf{ins}^E_i(N^{\forall i.A}))^{A\{i \leftarrow E\}}$, $D = A\{i \leftarrow E\}$, $r = \mathsf{ins}^E_i(s)$, and $\Theta; \Gamma; \Delta \vdash N^{\forall i.A} \,|\, s$ is derivable by hypothesis. By IH, we can derive $\Theta'; \Gamma'; \Delta' \vdash N^{\forall i.A} \,|\, s$ (we can assume that $i \notin \mathsf{FIV}(s)$ by variable convention). Then we can obtain the result by $\forall$E.

$\Box$

Lemma 5.3.6 (Validity Variable Substitution)

1. If $\Theta, v^A_{\Xi}; \Gamma; \Delta \vdash M^B \,|\, s$ and $\Theta; \cdot; \cdot \vdash N^A \,|\, t$ are derivable, then so is $\Theta; \Gamma; \Delta \vdash M\{v^A_{\Xi} \leftarrow N^A, t\}^{B\{v^A_{\Xi} \leftarrow t\}} \,|\, s\{v^A_{\Xi} \leftarrow t\}$.

2. If $\Theta, v^A_\Xi; \Gamma; \Delta \vdash s \equiv r : B$ and $\Theta; \cdot; \cdot \vdash N^A \mid t$ are derivable, then $\Theta; \Gamma; \Delta \vdash s\{v^A_\Xi \leftarrow t\} \equiv r\{v^A_\Xi \leftarrow t\} : B\{v^A_\Xi \leftarrow t\}$ is derivable.

Proof.- By mutual induction on the derivations of $\Theta, v^A_\Xi; \Gamma; \Delta \vdash M^B \mid s$ and $\Theta, v^A_\Xi; \Gamma; \Delta \vdash s \equiv r : B$.

If $v^A_\Xi \notin \mathsf{FVV}(M^B)$ – for 1 – or $v^A_\Xi \notin \mathsf{FVV}(s, r)$ – for 2 – the results hold trivially, since the substitutions will not affect M^B, s nor r.

We will analyze the other case ($v^A_\Xi \in \mathsf{FVV}(M^B)$), and consider the last step of each derivation for the rules which differ from those of HLP. (For all other rules, the proof is the same as shown in Lemma 4.4.13.

For 1:

- Case T-VarM: $M^B = v^A_\Xi$ (otherwise we would be in the case where $v^A_\Xi \notin \mathsf{FVV}(M^B)$) and $B = A$. $\Theta; \Gamma; \Delta \vdash N^A \mid t$ is derivable by Weakening from the hypothesis that $\Theta; \cdot; \cdot \vdash N^A \mid t$ is derivable.
- Case T-□I: $M^B = (!M')^{[\![s_2]\!]_{\Xi'} C}$ where $\mathsf{FIV}(\Theta) \cap \mathsf{FIV}(C) \subseteq \Xi'$. $\Theta; \cdot; \cdot \vdash M'^C \mid s_1$ and $\Theta; \cdot; \cdot \vdash s_1 \equiv s_2 : C$ are derivable by hypothesis, Note that, by variable convention, since $v^A_\Xi \in \mathsf{FVV}((!M')^{[\![s_2]\!]_{\Xi'} C})$, then $\mathsf{FIV}(t) \subseteq \mathsf{FIV}(([\![s_2]\!]_{\Xi'} C)\{v^A_\Xi \leftarrow t\})$ (otherwise free individual variables would be captured – that is, would become bound – upon substitution). By IH, we can derive both $\Theta; \cdot; \cdot \vdash M'\{v^A_\Xi \leftarrow N^A, t\}^{C\{v^A_\Xi \leftarrow t\}} \mid s_1\{v^A_\Xi \leftarrow t\}$ and $\Theta; \cdot; \cdot \vdash s_1\{v^A_\Xi \leftarrow t\} \equiv s_2\{v^A_\Xi \leftarrow t\} : C\{v^A_\Xi \leftarrow t\}$. The restriction that $\Theta \cap \mathsf{FIV}(C\{v^A_\Xi \leftarrow t\}) \subseteq \Xi'$ still holds by variable convention, since otherwise one or more variables in $\mathsf{FIV}(t)$ would be captured. We obtain the result by T-□I.
- Case T-□E: $M^B = (M^C_2\langle w^D_{\Xi_2} := r, M_1^{[\![r]\!]_{\Xi_1} D}\rangle)^C\{w^D_{\Xi_2} \leftarrow r\}$, $s = s_2\langle w^D_{\Xi_2} := r, s_1\rangle$ and, by hypothesis, $\Xi_1 \cap \mathsf{FIV}(D) \subseteq \Xi_2$ and both $\Theta, v^A_\Xi; \Gamma; \Delta \vdash M_1^{[\![r]\!]_{\Xi_1} D} \mid s_1$ and $\Theta, v^A_\Xi, w^D_{\Xi_2}; \Gamma; \Delta \vdash M^C_2 \mid s_2$ are derivable. There are two cases to consider:
  - If $v^A_\Xi = w^D_{\Xi_2}$, then $\Theta; \Gamma; \Delta \vdash M\{v^A_\Xi \leftarrow N^A, t\}^{B\{v^A_\Xi \leftarrow t\}} \mid s\{v^A_\Xi \leftarrow t\} = \Theta; \Gamma; \Delta \vdash M^B \mid s$, which is derivable by hypothesis (since in this case $B = C\{v^A_\Xi \leftarrow r\}$, this means that $v^A_\Xi \notin \mathsf{FVV}(B)$).
  - If $v^A_\Xi \neq w^D_{\Xi_2}$, then $M\{v^A_\Xi \leftarrow N^A, t\} = (N'\{v^A_\Xi \leftarrow N, t\}$ $\langle w^D_{\Xi_2} := r\{v^A_\Xi \leftarrow t\}, M'\{v^A_\Xi \leftarrow N, t\}\rangle)^{C\{w^D_{\Xi_2} \leftarrow r\}\{v^A_\Xi \leftarrow t\}}$, and $s\{v^A_\Xi \leftarrow t\} = s_1\{v^A_\Xi \leftarrow t\}\langle w^D_{\Xi_2} := r\{v^A_\Xi \leftarrow t\}, s_2\{v^A_\Xi \leftarrow t\}\rangle$. Note that, by variable convention, $v^A_\Xi \notin \mathsf{FVV}(D)$, and thus $([\![r]\!]_{\Xi_1} D)\{v^A_\Xi \leftarrow t\} = [\![r]\!]_{\Xi_1} D$. Additionally, $w^D_{\Xi_2} \notin \mathsf{FVV}(t)$ (nor $\mathsf{FVV}(N^A)$).
    By IH (1), $\Theta; \Gamma; \Delta \vdash M'\{v^A_\Xi \leftarrow N^A, t\}^{([\![r]\!]_{\Xi_1} D)\{v^A_\Xi \leftarrow t\}} \mid s_1\{v^A_\Xi \leftarrow t\}$ and $\Theta, w^D_{\Xi_2}; \Gamma; \Delta \vdash N'\{v^A_\Xi \leftarrow N^A, t\}^{C\{v^A_\Xi \leftarrow t\}} \mid s_2\{v^A_\Xi \leftarrow t\}$ are derivable. The restriction that $\Xi_1 \cap \mathsf{FIV}(D\{v^A_\Xi \leftarrow t\}) \subseteq \Xi_2$ still holds (since $v^A_\Xi \notin \mathsf{FVV}(D)$), so we obtain
    $\Theta; \Gamma; \Delta \vdash M\{v^A_\Xi \leftarrow N^A, t\}^{C\{v^A_\Xi \leftarrow t\}\{w^D_{\Xi_2} \leftarrow r\{v^A_\Xi \leftarrow t\}\}} \mid s\{v^A_\Xi \leftarrow t\}$ by T-□E.
- Case T-∀I: $M^B = (\mathsf{gen}_i(M'^D))^{\forall i.D}$, $s = \mathsf{gen}_i(s')$ and, by hypothesis, $\Theta, v^A_\Xi; \Gamma; \Delta \vdash M'^D \mid s'$ is derivable and $i \notin \mathsf{FIV}(\Theta, v^A_\Xi, \Gamma, \Delta)$. By IH, we can derive $\Theta; \Gamma; \Delta \vdash M'\{v^A_\Xi \leftarrow N^A, t\}^{D\{v^A_\Xi \leftarrow t\}} \mid s'\{v^A_\Xi \leftarrow t\}$, and we know that $i \notin \mathsf{FIV}(\Theta, \Gamma, \Delta)$. The result follows by ∀I.

- Case T-∀E: $M^B = (\mathsf{ins}_i^E(M'^{\forall i.A}))^{A\{i\leftarrow E\}}$, $s = \mathsf{ins}_i^E(s')$, and $\Theta, v_\Xi^A; \Gamma; \Delta \vdash M'^{\forall i.A} \,|\, s'$ is derivable by hypothesis. We can obtain the result using the IH (1) and T-∀E.

For 2:

- Case Eq-γ: $s = t\langle w_{\Xi'}^D{:=}s', !s\rangle$,$r = t\{w_{\Xi'}^D \leftarrow s'\}$, $B = C\{w_{\Xi'}^D \leftarrow s'\}$ and, by hypothesis, $\mathsf{FIV}(\Theta) \cap \mathsf{FIV}(D) \subseteq \Xi'$ and we can derive both $\Theta; \cdot; \cdot \vdash D \,|\, s'$ and $\Theta, v_\Xi^A, w_{\Xi'}^D; \Gamma; \Delta \vdash C \,|\, t$. By IH (1), we can derive $\Theta; \cdot; \cdot \vdash D \,|\, s'\{v_\Xi^A \leftarrow t\}$ and $\Theta, w_{\Xi'}^D; \Gamma; \Delta \vdash C\{v_\Xi^A \leftarrow t\} \,|\, t\{v_\Xi^A \leftarrow t\}$ (Note that, by variable convention, $v_\Xi^A \notin \mathsf{FVV}(D)$). The result follows by Eq-γ.
- Case Eq-ξ: $s = \mathsf{ins}_i^E(\mathsf{gen}_i(s'))$, $r = s'\{i \leftarrow E\}$, $B = C\{i \leftarrow E\}$ and, by hypothesis, $i \notin \mathsf{FIV}(\Theta, v_\Xi^A, \Gamma, \Delta)$ and $\Theta, v_\Xi^A; \Gamma; \Delta \vdash C \,|\, s'$ is derivable. By IH (1), we can derive $\Theta; \Gamma; \Delta \vdash C\{v_\Xi^A \leftarrow t\} \,|\, s'\{v_\Xi^A \leftarrow t\}$, and then obtain the result by Eq-ξ.
- Case Eq-ϵ_L: $s = \mathsf{ins}_i^E(s' + t)$, $r = \mathsf{ins}_i^E(s') + t$, $B = C\{i \leftarrow E\}$ and we can derive $\Theta, v_\Xi^A; \Gamma; \Delta \vdash \forall i.C \,|\, s'$ by hypothesis. The result follows by IH (1) and Eq-ϵ_L.
- Case Eq-ϵ_R: this case is analogous to the previous one.
- Case Eq-$\langle\rangle$: $s = t_1\langle w_{\Xi_2}^D{:=}r', s_1\rangle$, $r = t_2\langle w_{\Xi_2}^D{:=}r', s_2\rangle$, $B = C\{w_{\Xi_2}^D \leftarrow r'\}$ and, by hypothesis, $\Xi_1 \cap \mathsf{FIV}(D) \subseteq \Xi_2$ and both $\Theta, v_\Xi^A; \Gamma; \Delta \vdash s_1 \equiv s_2 : [\![r']\!]_{\Xi_1} D$ and $\Theta, v_\Xi^A, w_{\Xi_2}^D; \Gamma; \Delta \vdash t_1 \equiv t_2 : C$ are derivable. Note that, by variable convention, $v_\Xi^A \notin \mathsf{FVV}(D)$. By IH (2), we can derive $\Theta; \Gamma; \Delta \vdash s_1\{v_\Xi^A \leftarrow t\} \equiv s_2\{v_\Xi^A \leftarrow t\} : [\![r']\!]_{\Xi_1} D$ and $\Theta, w_{\Xi_2}^D; \Gamma; \Delta \vdash t_1\{v_\Xi^A \leftarrow t\} \equiv t_2\{v_\Xi^A \leftarrow t\} : C\{v_\Xi^A \leftarrow t\}$. The result follows by Eq-$\langle\rangle$.
- Case Eq-gen: the result follows by IH (2) and Eq-gen. Note that, since $i \notin \mathsf{FIV}(\Theta, v_\Xi^A, \Gamma, \Delta)$, then $i \notin \mathsf{FIV}(\Theta, \Gamma, \Delta)$.
- Case Eq-ins: the result follows by IH (2) and Eq-ins.

□

Lemma 5.3.7 (Validity Variable Substitution with Equivalence)
If $\Theta, v_\Xi^A; \Gamma; \Delta \vdash M^B \,|\, s$, $\Theta; \cdot; \cdot \vdash N^A \,|\, r$ and $\Theta; \cdot; \cdot \vdash r \equiv t : A$ are derivable, then there exists s' such that both $\Theta; \Gamma; \Delta \vdash M^B\{v_\Xi^A \leftarrow N^A, t\}^{B\{v_\Xi^A \leftarrow t\}} \,|\, s'$ and $\Theta; \Gamma; \Delta \vdash s' \equiv s\{v_\Xi^A \leftarrow t\} : B\{v_\Xi^A \leftarrow t\}$ are derivable.

Proof.- By induction on the derivation of $\Theta; \Gamma; \Delta \vdash M^B \,|\, s$.

If $v_\Xi^A \notin \mathsf{FVV}(M^B)$, then the result holds immediately with $s' = s$, since the substitutions do not affect M, B nor s. We will focus on the other case ($v_\Xi^A \in \mathsf{FVV}(M^B)$).

We consider the last step of the derivation for the rules which differ from those of HLP. (For all other rules, the proof is the same as shown in Lemma 4.4.14.

- Case T-VarM: $M^B = s = w_{\Xi'}^B = v_\Xi^A$ (since $v_\Xi^A \in \mathsf{FVV}(M^B)$): here $A = B$, $\Xi = \Xi'$, $M\{v_\Xi^A \leftarrow N^A, t\} = N^A$ and $s\{v_\Xi^A \leftarrow t\} = t$. Note that, since $A = B$, then $v_\Xi^A \notin \mathsf{FVV}(B)$. Take $s' = r$. Then $\Theta; \Gamma; \Delta \vdash M\{v_\Xi^A \leftarrow N^A, t\}^{B\{v_\Xi^A \leftarrow t\}} \,|\, s' = \Theta; \Gamma; \Delta \vdash N^A \,|\, r$, which is derivable by hypothesis.

- Case T-□I: $M = (!M_1)^{[\![s_1]\!]_{\Xi'}C}$, $s = !s_1$, $B = [\![s_1]\!]_{\Xi'}C$, $\mathsf{FIV}(\Theta, v^A_\Xi) \cap \mathsf{FIV}(C) \subseteq \Xi'$, both $\Theta, v^A_\Xi; \cdot; \cdot \vdash M_1^C \mid s_2$ and $\Theta, v^A_\Xi; \cdot; \cdot \vdash s_2 \equiv s_1 : C$ are derivable by hypothesis.

  By IH, we can derive both $\Theta; \cdot; \cdot \vdash M_1\{v^A_\Xi \leftarrow N^A, t\}^{C\{v^A_\Xi \leftarrow t\}} \mid s_3$ and $\Theta; \cdot; \cdot \vdash s_3 \equiv s_2\{v^A_\Xi \leftarrow t\} : C\{v^A_\Xi \leftarrow t\}$ for some proof witness s_3.

  And, by Lemma 5.3.6, we can also derive $\Theta; \cdot; \cdot \vdash s_2\{v^A_\Xi \leftarrow t\} \equiv s_1\{v^A_\Xi \leftarrow t\} : C\{v^A_\Xi \leftarrow t\}$.

  We already know that $\mathsf{FIV}(\Theta) \cap \mathsf{FIV}(C\{v^A_\Xi \leftarrow t\}) \subseteq \Xi'$, and we can assume that $\mathsf{FIV}(\Theta) \cap \mathsf{FIV}(C\{v^A_\Xi \leftarrow t\}) \subseteq \Xi'$ by variable convention, since otherwise one or more variables in $\mathsf{FIV}(t)$ would be captured.

  Take $s' = !(s_2\{v^A_\Xi \leftarrow t\})$. By T-□I, we obtain:
  $\Theta; \Gamma; \Delta \vdash !(M_1\{v^A_\Xi \leftarrow N^A, t\})^{[\![s_1\{v^A_\Xi \leftarrow t\}]\!]_{\Xi'}C\{v^A_\Xi \leftarrow t\}} \mid !(s_2\{v^A_\Xi \leftarrow t\})$. (Note that the result also holds if we take $s' = !(s_1\{v^A_\Xi \leftarrow t\})$ or $s' = !s_3$).

- Case T-□E:$M^B = (M'\langle w^D_{\Xi_2} := r', N'\rangle)^{C\{w^D_{\Xi_2} \leftarrow r'\}}$, $s = s_1\langle w^D_{\Xi_2} := r', s_2\rangle$, $B = C\{w^D_{\Xi_2} \leftarrow r'\}$, and, by hypothesis, $\Xi_1 \cap \mathsf{FIV}(D) \subseteq \Xi_2$ and the judgments $\Theta, v^A_\Xi; \Gamma; \Delta \vdash N'^{[\![r']\!]_{\Xi_1}D} \mid s_2$ and $\Theta, v^A_\Xi, w^D_{\Xi_2}; \Gamma; \Delta \vdash M'^C \mid s_1$ are derivable. Take $s' = s\{v^A_\Xi \leftarrow t\}$. There are now two cases to consider:
  - $w^D_{\Xi_2} = v^A_\Xi$: here $M\{v^A_\Xi \leftarrow N^A, t\} = M$, $s\{v^A_\Xi \leftarrow t\} = s$, $\Theta, v^A_\Xi, w^D_{\Xi_2} = \Theta, v^A_\Xi$ and $B\{v^A_\Xi \leftarrow t\} = B$ (since $w^D_{\Xi_2} = v^A_\Xi$ and $B = C\{w^D_{\Xi_2} \leftarrow r'\}$, this means that $v^A_\Xi \notin \mathsf{FVV}(B)$). We can derive $\Theta, v^A_\Xi; \Gamma; \Delta \vdash M^B \mid s$ from $\Theta, v^A_\Xi; \Gamma; \Delta \vdash N'^{[\![r']\!]_{\Xi_1}D} \mid s_2$ and $\Theta, v^A_\Xi, w^D_{\Xi_2}; \Gamma; \Delta \vdash M'^C \mid s_1$ by T-□E, and obtain $\Theta; \Gamma; \Delta \vdash M^B \mid s$ by Strengthening (since $v^A_\Xi \notin \mathsf{FVV}(M^B) \cup \mathsf{FVV}(s)$).
  - $w^D \neq v^A_\Xi$: in this case $M\{v^A_\Xi \leftarrow N^A, t\} = (M'\{v^A_\Xi \leftarrow N, t\}$ $\langle w^D_{\Xi_2} := r\{v^A_\Xi \leftarrow t\}, N'\{v^A_\Xi \leftarrow N, t\}\rangle)^{C\{w^D_{\Xi_2} \leftarrow r\}\{v^A_\Xi \leftarrow t\}}$, and $s\{v^A_\Xi \leftarrow t\} =$ $s_1\{v^A_\Xi \leftarrow t\}\langle w^D_{\Xi_2} := r\{v^A_\Xi \leftarrow t\}, s_2\{v^A_\Xi \leftarrow t\}\rangle$. Note that, by variable convention, $v^A_\Xi \notin \mathsf{FVV}(D)$, and thus $([\![r']\!]_{\Xi_1}D)\{v^A_\Xi \leftarrow t\} = [\![r'\{v^A_\Xi \leftarrow t\}]\!]_{\Xi_1}D$. Additionally, $w^D_{\Xi_2} \notin \mathsf{FVV}(t)$ (nor $\mathsf{FVV}(N^A)$).

    By IH, the following judgements are derivable for some witnesses s_3 and s_4:

    $$\begin{array}{l}\Theta; \Gamma; \Delta \vdash N'\{v^A_\Xi \leftarrow N^A, t\}^{[\![r'\{v^A_\Xi \leftarrow t\}]\!]_{\Xi_1}D} \mid s_4\\ \Theta, w^D_{\Xi_2}; \Gamma; \Delta \vdash M'\{v^A_\Xi \leftarrow N^A, t\}^{C\{v^A_\Xi \leftarrow t\}} \mid s_3\\ \Theta; \Gamma; \Delta \vdash s_3 \equiv s_1\{v^A_\Xi \leftarrow t\} : C\{v^A_\Xi \leftarrow t\}\\ \Theta, w^D_{\Xi_2}; \Gamma; \Delta \vdash s_4 \equiv s_2\{v^A_\Xi \leftarrow t\} : [\![r'\{v^A_\Xi \leftarrow t\}]\!]_{\Xi_1}D\end{array}$$

    The condition that $\Xi_1 \cap \mathsf{FIV}(D) \subseteq \Xi_2$ still holds, so we can use T-□E to derive: $\Theta; \Gamma; \Delta \vdash$ $M'\{v^A_\Xi \leftarrow N^A, t\}\langle w^D_{\Xi_2} := r'\{v^A_\Xi \leftarrow t\}, N'\{v^A_\Xi \leftarrow N^A, t\}\rangle^{C\{v^A_\Xi \leftarrow t\}\{w^D_{\Xi_2} \leftarrow (r'\{v^A_\Xi \leftarrow t\})\}} \mid$ $s_3\langle w^D_{\Xi_2} := r'\{v^A_\Xi \leftarrow t\}, s_4\rangle$ (note that $C\{v^A_\Xi \leftarrow t\}\{w^D_{\Xi_2} \leftarrow (r'\{v^A_\Xi \leftarrow t\})\}$ is the same as $C\{w^D_{\Xi_2} \leftarrow r'\}\{v^A_\Xi \leftarrow t\}$, since $w^D_{\Xi_2} \notin \mathsf{FVV}(t)$).

    Take $s' = s_3\langle w^D_{\Xi_2} := r'\{v^A_\Xi \leftarrow t\}, s_4\rangle$. We can derive $\Theta; \Gamma; \Delta \vdash s' \equiv s\{v^A_\Xi \leftarrow t\} : C\{w^D_{\Xi_2} \leftarrow r'\}\{v^A_\Xi \leftarrow t\}$ by Eq-⟨⟩.

- Case T-∀I: $M^B = \mathsf{gen}_i(M_1^A)^{\forall i.A}$, $s = \mathsf{gen}_i(s_1)$ and $\Theta, v^A_\Xi; \Gamma; \Delta \vdash M_1^A | s_1$ is derivable by hypothesis, with $i \notin \mathsf{FIV}(\Theta, v^A_\Xi, \Gamma, \Delta)$. By IH, we can derive $\Theta; \Gamma; \Delta \vdash M_1\{v^A_\Xi \leftarrow$

$N^A, t\}^{A\{v^A_\Xi \leftarrow t\}} | s_2$ and $\Theta; \Gamma; \Delta \vdash\ s_2 \equiv s_1\{v^A_\Xi \leftarrow t\} : A\{v^A_\Xi \leftarrow t\}$ for some proof witness s_2, and we already know that $i \notin \mathsf{FIV}(\Theta, \Gamma, \Delta)$. Take $s' = \mathsf{gen}_i(s_2)$. By $\forall$I, we can derive $\Theta; \Gamma; \Delta \vdash M_1\{v^A_\Xi \leftarrow N^A, t\}^{\forall i.A\{v^A_\Xi \leftarrow t\}} | \mathsf{gen}_i(s_2)$. And, by Eq-gen, we can also derive $\Theta; \Gamma; \Delta \vdash\ \mathsf{gen}_i(s_2) \equiv \mathsf{gen}_i(s_1\{v^A_\Xi \leftarrow t\}) : \forall i.A\{v^A_\Xi \leftarrow t\}$.

- Case T-$\forall$E: $M^B = \mathsf{ins}^E_i(M_1^{\forall i.A})^{A\{i \leftarrow E\}}$, $s = \mathsf{ins}^E_i(s_1)$ and $\Theta, v^A_\Xi; \Gamma; \Delta \vdash M_1^{\forall i.A} \,|\, s_1$ is derivable by hypothesis.
  By IH, we can derive $\Theta; \Gamma; \Delta \vdash M_1\{v^A_\Xi \leftarrow N^A, t\}^{\forall i.A\{v^A_\Xi \leftarrow t\}} \,|\, s_2$ and $\Theta; \Gamma; \Delta \vdash\ s_2 \equiv s_1\{v^A_\Xi \leftarrow t\} : \forall i.A\{v^A_\Xi \leftarrow t\}$. Take $s' = \mathsf{ins}^E_i(s_2)$. We obtain the results by $\forall$E and Eq-ins respectively.

□

Remark 5.3.8 For a set of individual variables Ξ, an individual variable i and an expression E, we define $\Xi\{i \blacktriangleleft E\}$ as $(\Xi \backslash \{i\}) \cup \mathsf{FIV}(E)$ if $i \in \Xi$, and Ξ otherwise.

Lemma 5.3.9 For any formula A, individual variable i and expression E:
$\mathsf{FIV}(A\{i \leftarrow E\}) = (\mathsf{FIV}(A))\{i \blacktriangleleft E\}$.

Proof.- By structural induction on A, using the definition of free individual variables. □

Lemma 5.3.10 If $\Xi \cap \Xi_2 \subseteq \Xi'$, then $\Xi\{i \blacktriangleleft E\} \cap \Xi_2\{i \blacktriangleleft E\} \subseteq \Xi'\{i \blacktriangleleft E\}$.

Proof.- We must consider three cases:

- If $i \in \Xi$ and $i \in \Xi_2$, then $i \in \Xi'$, and thus $\Xi'\{i \blacktriangleleft E\} = \Xi' \backslash \{i\} \cup \mathsf{FIV}(E)$, and $(\Xi \backslash \{i\} \cup \mathsf{FIV}(E)) \cap \Xi_2\{i \blacktriangleleft E\} \subseteq \Xi'\{i \blacktriangleleft E\}$.
- If $i \in \Xi$ and $i \notin \Xi_2$, then $\Xi_2\{i \blacktriangleleft E\} = \Xi_2$ and $(\Xi \backslash \{i\} \cup \mathsf{FIV}(E)) \cap \mathsf{FIV}(A) \subseteq \Xi'\{i \blacktriangleleft E\}$ whether or not $i \in \Xi'$.
- If $i \notin \Xi$, then $\Xi\{i \blacktriangleleft E\} = \Xi$, and $\Xi \cap \Xi_2\{i \blacktriangleleft E\} \subseteq \Xi'\{i \blacktriangleleft E\}$ whether or not $i \in \Xi'$.

In all cases, $\Xi\{i \blacktriangleleft E\} \cap \mathsf{FIV}(A\{i \leftarrow E\}) \subseteq \Xi'\{i \blacktriangleleft E\}$. □

Corollary 5.3.11 If $\Xi \cap \mathsf{FIV}(A) \subseteq \Xi'$, then $\Xi\{i \blacktriangleleft E\} \cap \mathsf{FIV}(A\{i \leftarrow E\}) \subseteq \Xi'\{i \blacktriangleleft E\}$.

Proof.- By Lemmas 5.3.9 and 5.3.10. □

Lemma 5.3.12 (Individual Variable Substitution)

1. If $\Theta; \Gamma; \Delta \vdash M^D \,|\, r$ is derivable, then
   $\Theta\{i \leftarrow E\}; \Gamma\{i \leftarrow E\}; \Delta\{i \leftarrow E\} \vdash M\{i \leftarrow E\}^{D\{i \leftarrow E\}} \,|\, r\{i \leftarrow E\}$ is also derivable.
2. If $\Theta; \Gamma; \Delta \vdash\ r_1 \equiv r_2 : D$ is derivable, then so is $\Theta\{i \leftarrow E\}; \Gamma\{i \leftarrow E\}; \Delta\{i \leftarrow E\} \vdash\ r_1\{i \leftarrow E\} \equiv r_2\{i \leftarrow E\} : D\{i \leftarrow E\}$.

Proof.- By mutual induction on the derivation of $\Theta; \Gamma; \Delta \vdash M^D \,|\, r$ and $\Theta; \Gamma; \Delta \vdash\ r_1 \equiv r_2 : D$. Let $\Theta\{i \leftarrow E\} = \Theta\{i \leftarrow E\}$, $\Gamma\{i \leftarrow E\} = \Gamma\{i \leftarrow E\}$, $\Delta\{i \leftarrow E\} = \Delta\{i \leftarrow E\}$ We analyze the last rule used:

For 1:

- Case T-Var: $M^D = r = x^D$ and $\Gamma = \Gamma', x^D$. We can derive $\Theta\{i \leftarrow E\}; \Gamma\{i \leftarrow E\}; \Delta\{i \leftarrow E\} \vdash x^{D\{\cancel{i} \leftarrow E\}} \mid x^{D\{\cancel{i} \leftarrow E\}}$ by T-Var.
- Case T-VarM: this case is also immediate.
- Case T-$\supset$I: $M = \lambda x^A.M'^B$, $D = A \supset B$, $r = \lambda x^A.s$ and $\Theta; \Gamma, x^A; \Delta \vdash M'^B \mid s$ is derivable by hypothesis.

  By IH (1), we can derive $\Theta\{i \leftarrow E\}; \Gamma\{i \leftarrow E\}, x^{A\{\cancel{i} \leftarrow E\}}; \Delta\{i \leftarrow E\} \vdash M'\{i \leftarrow E\}^{B\{\cancel{i} \leftarrow E\}} \mid s\{i \leftarrow E\}$, and then obtain the result by T-$\supset$I.
- Case T-$\supset$E, T-NAbs, T-Name, T-PlusL, T-PlusR: analogously to the previous case, the result is obtained by using the IH (1) on the premises and applying the corresponding rule.
- Case T-$\Box$I: $M = !M'^A$, $D = [\![t]\!]_{\Xi}A$, $r = !t$ and, by hypothesis, $\mathsf{FIV}(\Theta) \cap \mathsf{FIV}(A) \subseteq \Xi$ and we can derive both $\Theta; \cdot; \cdot \vdash M^A \mid s$ and $\Theta; \cdot; \cdot \vdash s \equiv t : A$. By IH (1) and (2) respectively, we can derive $\Theta\{i \leftarrow E\}; \cdot; \cdot \vdash M\{i \leftarrow E\}^{A\{\cancel{i} \leftarrow E\}} \mid s\{i \leftarrow E\}$ and $\Theta\{i \leftarrow E\}; \cdot; \cdot \vdash s\{i \leftarrow E\} \equiv t\{i \leftarrow E\} : A\{i \leftarrow E\}$. And since $\mathsf{FIV}(\Theta) \cap \mathsf{FIV}(A) \subseteq \Xi$, then $\mathsf{FIV}(\Theta\{i \leftarrow E\}) \cap \mathsf{FIV}(A\{i \leftarrow E\}) \subseteq (\Xi \backslash i \cup \mathsf{FIV}(E))$. We obtain the result by T-$\Box$I.
- Case T-$\Box$E: $M = N^C\langle v^A_{\Xi'} := r', M^{[\![r]\!]_{\Xi}A}\rangle$, $D = C\{v^A_{\Xi'} \leftarrow r'\}$, $r = t\langle v^A_{\Xi'} := r', s\rangle$ and, by hypothesis, $\Xi \cap \mathsf{FIV}(A) \subseteq \Xi'$ and both $\Theta; \Gamma; \Delta \vdash M^{[\![r']\!]_{\Xi}A} \mid s$ and $\Theta, v^A_{\Xi'}; \Gamma; \Delta \vdash N^C \mid t$ are derivable. By IH (1), we can derive $\Theta\{i \leftarrow E\}; \Gamma\{i \leftarrow E\}; \Delta\{i \leftarrow E\} \vdash M\{i \leftarrow E\}^{([\![r']\!]_{\Xi}A)\{\cancel{i} \leftarrow E\}} \mid s\{i \leftarrow E\}$ and $\Theta\{i \leftarrow E\}, (v^A_{\Xi'})\{i \leftarrow E\}; \Gamma\{i \leftarrow E\}; \Delta\{i \leftarrow E\} \vdash N\{i \leftarrow E\}^{C\{\cancel{i} \leftarrow E\}} \mid t\{i \leftarrow E\}$.

  By Corollary 5.3.11, $\Xi\{i \blacktriangleleft E\} \cap \mathsf{FIV}(A\{i \leftarrow E\}) \subseteq \Xi'\{i \blacktriangleleft E\}$. We can obtain the result by T-$\Box$E.
- T-$\forall$I: $M = \mathsf{gen}_j(M')$, $D = \forall j.A$, $r = \mathsf{gen}_j(s)$, $\Theta; \Gamma; \Delta \vdash M'^A | s$ is derivable by hypothesis and $j \notin \mathsf{FIV}(\Theta, \Gamma, \Delta)$. By IH (1), we can derive $\Theta\{i \leftarrow E\}; \Gamma\{i \leftarrow E\}; \Delta\{i \leftarrow E\} \vdash M'\{i \leftarrow E\}^{A\{\cancel{i} \leftarrow E\}} | s\{i \leftarrow E\}$. In order to ensure that $j \notin \mathsf{FIV}(\Theta\{i \leftarrow E\}, \Gamma\{i \leftarrow E\}, \Delta\{i \leftarrow E\})$, we need to know that $j \notin \mathsf{FIV}(E)$ (or $i \notin \mathsf{FIV}(\Theta, \Gamma, \Delta)$). But this is not a problem, since if $j \in \mathsf{FIV}(E)$, it can be renamed to a fresh individual variable j'. The result holds by $\forall$I.
- T-$\forall$E: $M = \mathsf{ins}^{E'}_j(M'^{\forall j.A})$, $D = A\{j \leftarrow E'\}$ $r = \mathsf{ins}^{E'}_j(s)$ and $\Theta; \Gamma; \Delta \vdash M^{\forall j.A} \mid s$ is derivable by hypothesis. By IH (1), we can derive $\Theta\{i \leftarrow E\}; \Gamma\{i \leftarrow E\}; \Delta\{i \leftarrow E\} \vdash M'\{i \leftarrow E\}^{\forall j.A\{\cancel{i} \leftarrow E\}} \mid s\{i \leftarrow E\}$, and then obtain $\Theta\{i \leftarrow E\}; \Gamma\{i \leftarrow E\}; \Delta\{i \leftarrow E\} \vdash \mathsf{ins}^{E'\{\cancel{i} \leftarrow E\}}_j(M'\{i \leftarrow E\})^{A\{\cancel{i} \leftarrow E\}\{j \leftarrow E'\{\cancel{i} \leftarrow E\}\}}$
  $\mid \mathsf{ins}^{E'\{\cancel{i} \leftarrow E\}}_j(s\{i \leftarrow E\})$ by T-$\forall$E.

For 2:

- Eq-γ: $D = C\{v^A_{\Xi} \leftarrow s\}$, $r_1 = t\langle v^A_{\Xi} := s, !s\rangle$, $r_2 = t\{v^A_{\Xi} \leftarrow s\}$ and, by hypothesis, $\mathsf{FIV}(\Theta) \cap \mathsf{FIV}(A) \subseteq \Xi$ and both $\Theta; \cdot; \cdot \vdash A \mid s$ and $\Theta, v^A_{\Xi}; \Gamma; \Delta \vdash C \mid t$ are derivable. By IH (1), we can derive $\Theta\{i \leftarrow E\}; \cdot; \cdot \vdash A\{i \leftarrow E\} \mid s\{i \leftarrow E\}$ and $(\Theta, v^A_{\Xi})\{i \leftarrow E\}; \Gamma\{i \leftarrow E\}; \Delta\{i \leftarrow E\} \vdash C\{i \leftarrow E\} \mid t\{i \leftarrow E\}$.

  Let $\Xi\{i \blacktriangleleft E\} = \Xi' \backslash \{i\} \cup \mathsf{FIV}(E)$ if $i \in \Xi$, and $\Xi\{i \blacktriangleleft E\} = \Xi$ otherwise. By Corollary 5.3.11, $\mathsf{FIV}(\Theta\{i \leftarrow E\}) \cap \mathsf{FIV}(A\{i \leftarrow E\}) \subseteq \Xi\{i \blacktriangleleft E\}$. The result hold by Eq-$\gamma$.

- Eq-ξ: $D = A\{j \leftarrow E'\}$, $r_1 = \mathsf{ins}_j^{E'}(\mathsf{gen}_j(s))$, $r_2 = s\{j \leftarrow E'\}$, $\Theta;\Gamma;\Delta \vdash A \,|\, s$ is derivable by hypothesis and $j \notin \mathsf{FIV}(\Theta,\Gamma,\Delta)$.

  We can safely assume that $j \notin \mathsf{FIV}(E)$. Otherwise, we could easily rename j to a fresh individual variable.

  By IH (1), we can derive $\Theta\{i\leftarrow E\};\Gamma\{i\leftarrow E\};\Delta\{i\leftarrow E\} \vdash A\{i\leftarrow E\} \,|\, s\{i\leftarrow E\}$. And, since $j \notin \mathsf{FIV}(E)$, then $j \notin \mathsf{FIV}(\Theta\{i\leftarrow E\},\Gamma\{i\leftarrow E\},\Delta\{i\leftarrow E\})$. The result is obtained by Eq-ξ.

  Eq-ϵ_L: $D = A\{j \leftarrow E'\}$, $r_1 = \mathsf{ins}_j^{E'}(s+t)$, $r_2 = \mathsf{ins}_j^{E'}(s)+t$ and $\Theta;\Gamma;\Delta \vdash \forall j.A \,|\, s$ is derivable by hypothesis.

  By IH (1), we can derive $\Theta\{i \leftarrow E\};\Gamma\{i \leftarrow E\};\Delta\{i \leftarrow E\} \vdash \forall j.A\{i \leftarrow E\} \,|\, s\{i \leftarrow E\}$. The result holds by Eq-ϵ_L.

  Eq-ϵ_R: this case is analogous to the previous one.

  Eq-$\langle\rangle$: $D = C\{v^A_{\Xi'} \leftarrow r\}$, $r_1 = t\langle v^A_{\Xi'}{:=}r, s\rangle$, $r_2 = t'\langle v^A_{\Xi'}{:=}r, s'\rangle$ and, by hypothesis,

  $\Theta;\Gamma;\Delta \vdash s \equiv s' : [\![r]\!]_{\Xi}A$ and $\Theta, v^A_{\Xi'};\Gamma;\Delta \vdash t \equiv t' : C$ are derivable and $\Xi \cap \mathsf{FIV}(A) \subseteq \Xi'$.

  By IH (2) we can derive $\Theta\{i\leftarrow E\};\Gamma\{i\leftarrow E\};\Delta\{i\leftarrow E\} \vdash s\{i\leftarrow E\} \equiv s'\{i\leftarrow E\} : [\![r]\!]_{\Xi}A\{i\leftarrow E\}$ and $(\Theta, v^A_{\Xi'})\{i\leftarrow E\};\Gamma\{i\leftarrow E\};\Delta\{i\leftarrow E\} \vdash t\{i\leftarrow E\} \equiv t'\{i\leftarrow E\} : C\{i\leftarrow E\}$. And by Corollary 5.3.11, $\Xi\{i \blacktriangleleft E\} \cap \mathsf{FIV}(A\{i\leftarrow E\}) \subseteq \Xi'\{i \blacktriangleleft E\}$. The result follows by Eq-$\langle\rangle$.

  All other cases are similar to the previous ones, using IH (1) for the principal rules and IH (2) for the compatibility rules.

□

Lemma 5.3.13 (Truth Variable Substitution) If $\Theta;\Gamma,y^A;\Delta \vdash M^B \,|\, s$ and $\Theta;\Gamma;\Delta \vdash N^A \,|\, t$ are derivable, then so is: $\Theta;\Gamma;\Delta \vdash M\{y^A \leftarrow N^A\}^B \,|\, s\{y^A \leftarrow t\}$.

Proof.- By induction on the derivation of $\Theta;\Gamma,y^A;\Delta \vdash M^B \,|\, s$.

For the rules which differ from those of HLP, the proof is analogous to the proof of Lemma 5.3.6, except for the rules T-$\Box\mathsf{I}$ and T-$\Box\mathsf{E}$.

For the rest of the rules – those already present in HLP – the proof is identical to the proof of Lemma 4.4.12.

- Case T-$\Box\mathsf{I}$: $M^B = (!M')^{[\![s_2]\!]_{\Xi'}C}$ and $s = !s_1$. Since $M\{y^A \leftarrow N^A\} = M$ and $s\{y^A \leftarrow t\} = s$, then $\Theta;\Gamma;\Delta \vdash M\{y^A \leftarrow N^A\}^B \,|\, s\{y^A \leftarrow t\}$ is already derivable by hypothesis.
- Case T-$\Box\mathsf{E}$: $M^B = (M_2^C\langle v^D_{\Xi_2}{:=}r, M_1^{[\![r]\!]_{\Xi_1}D}\rangle)^C\{v^D_{\Xi_2} \leftarrow r\}$, $s = s_2\langle v^D_{\Xi_2}{:=}r, s_1\rangle$ and, by hypothesis, $\Xi_1 \cap \mathsf{FIV}(D) \subseteq \Xi_2$ and both $\Theta;\Gamma,y^A;\Delta \vdash M_1^{[\![r]\!]_{\Xi_1}D} \,|\, s_1$ and $\Theta, v^D_{\Xi_2};\Gamma,y^A;\Delta \vdash M_2^C \,|\, s_2$ are derivable.

  By IH, we can derive $\Theta;\Gamma;\Delta \vdash M_1\{y^A \leftarrow N^A\}^{[\![r]\!]_{\Xi_1}D} \,|\, s_1\{y^A \leftarrow t\}$ and $\Theta, v^D_{\Xi_2};\Gamma;\Delta \vdash M_2\{y^A \leftarrow N^A\}^C \,|\, s_2\{y^A \leftarrow t\}$. The restriction that $\Xi_1 \cap \mathsf{FIV}(D) \subseteq \Xi_2$ still holds, so we obtain the result by $\Box\mathsf{E}$.

□

Lemma 5.3.14 (Falsehood Variable Renaming) If the judgement $\Theta; \Gamma; \Delta, \alpha^A, \beta^A \vdash M^B \,|\, s$ is derivable, then so is $\Theta; \Gamma; \Delta, \beta^A \vdash M\{\alpha^A \leftarrow \beta^A\}^B \,|\, s\{\alpha^A \leftarrow \beta^A\}$.

Proof.- By induction on the derivation of $\Theta; \Gamma; \Delta, \alpha^A, \beta^A \vdash M^B \,|\, s$. For the rules which differ from those of HLP, the proof is analogous to the proof of Lemma 5.3.13. For the rest of the rules (i.e. those already present in HLP) the proof is identical to the proof of Lemma 4.4.15. □

Lemma 5.3.15 (Structural Substitution)
If both $\Theta; \Gamma; \Delta, \alpha^{A \supset B} \vdash M^D \,|\, s$ and $\Theta; \Gamma; \Delta \vdash N^A \,|\, t$ are derivable, then so is $\Theta; \Gamma; \Delta, \beta^B \vdash M(\!|\, [\alpha^{A \supset B}](\bullet) \leftarrow [\beta^B](\bullet) N^A \,|\!)^D \,|\, s(\!|\, [\alpha^{A \supset B}](\bullet) \leftarrow [\beta^B](\bullet) t \,|\!)$.

Proof.- By induction on the derivation of $\Theta; \Gamma; \Delta, \alpha^{A \supset B} \vdash M^D \,|\, s$. For the rules which differ from those of HLP, the proof is analogous to the proof of Lemma 5.3.13. For the rest of the rules (i.e. those already present in HLP) the proof is identical to the proof of Lemma 4.4.16. □

Corollary 5.3.16 If $\Theta; \Gamma; \Delta, \alpha^{A \supset B} \vdash M^\perp \,|\, s$ and $\Theta; \Gamma; \Delta \vdash N^A \,|\, t$ are both derivable, then so is: $\Theta; \Gamma; \Delta \vdash (\mu\beta^B.M^\perp(\!|\, [\alpha^{A \supset B}](\bullet) \leftarrow [\beta^B](\bullet) N^A \,|\!))^B \,|\, \mu\beta^B.s(\!|\, [\alpha^{A \supset B}](\bullet) \leftarrow [\beta^B](\bullet) t \,|\!)$.

Proof.- From the hypotheses, by Lemma 5.3.15, we know that:

$$\Theta; \Gamma; \Delta, \beta^B \vdash M^\perp(\!|\, [\alpha^{A \supset B}](\bullet) \leftarrow [\beta^B](\bullet) N^A \,|\!)^\perp \,|\, s(\!|\, [\alpha^{A \supset B}](\bullet) \leftarrow [\beta^B](\bullet) t \,|\!)$$

is derivable. The result follows by T-NAbs. Note that, since $\beta^B \neq \alpha^{A \supset B}$, then $(\mu\beta^B.M^\perp)(\!|\, [\alpha](\bullet) \leftarrow [\beta](\bullet) N \,|\!) = \mu\beta^B.(M^\perp(\!|\, [\alpha](\bullet) \leftarrow [\beta](\bullet) N \,|\!))$, and analogously for s.

□

Lemma 5.3.17 (Inversion) If $\Theta; \Gamma; \Delta \vdash N^D \,|\, r$ is derivable, then:

- if $N^D = x^A$, then $x^A \in \Gamma$, $r = x^A$ and $D = A$;
- if $N^D = v^A_\Xi$, then $v^A_\Xi \in \Theta$, $r = v^A_\Xi$ and $D = A$;
- if $N^D = (\lambda x^A.M^B)^{A \supset B}$, then $\Theta; \Gamma, x^A; \Delta \vdash M^B \,|\, s$ is derivable for some s, and $r = \lambda x^A.s$ and $D = A \supset B$;
- if $N^D = (M_1^{A \supset B} M_2^A)^B$, then both $\Theta; \Gamma; \Delta \vdash M_1^{A \supset B} \,|\, s$ and $\Theta; \Gamma; \Delta \vdash M_2^A \,|\, t$ are derivable for some s and t, and $r = s \cdot t$ and $D = B$;
- if $N^D = (!M^A)^{[\![t]\!]_\Xi A}$, then $\exists s$ s.t. both $\Theta; \cdot; \cdot \vdash M^A \,|\, s$ and $\Theta; \cdot; \cdot \vdash s \equiv t : A$ are derivable, $\mathsf{FIV}(\Theta) \cap \mathsf{FIV}(A) \subseteq \Xi$, $r = !t$ and $D = [\![t]\!]_\Xi A$;
- if $N^D = (M_2^B \langle v^A_{\Xi'} := r', M_1^{[\![r']\!]_\Xi A} \rangle)^{B\{v^A \leftarrow r^A\}}$, then both $\Theta; \Gamma; \Delta \vdash M_1^{[\![r']\!]_\Xi A} \,|\, s$ and $\Theta, v^A_{\Xi'}; \Gamma; \Delta \vdash M_2^B \,|\, t$ are derivable for some s and t, $r = t\langle v^A_{\Xi'} := r, s \rangle$, $\Xi \cap \mathsf{FIV}(A) \subseteq \Xi'$ and $D = B\{v^A \leftarrow r^A\}$;
- if $N^D = ([\alpha^A]M^A)^\perp$, then $\exists\, \Delta'$, s s.t. $\Delta = \Delta', \alpha^A$, $\Theta; \Gamma; \Delta', \alpha^A \vdash M^A \,|\, s$ is derivable, and $r = [\alpha^A]s$ and $D = \perp$;

- if $N^D = (\mu\alpha^A.M^\perp)^A$, then $\Theta;\Gamma;\Delta,\alpha^A \vdash M^\perp \,|\, s$ is derivable for some s, and $r = \mu\alpha^A.s$ and $D = A$;
- if $N^D = (M^A +_{\mathrm{L}} t)^A$, then $\Theta;\Gamma;\Delta \vdash M^A \,|\, s$ is derivable for some s, and $r = s+t$ and $D = A$;
- if $N^D = (s +_{\mathrm{R}} M^B)^B$, then $\Theta;\Gamma;\Delta \vdash M^B \,|\, t$ is derivable for some t, and $r = s+t$ and $D = B$.
- if $N^D = (\mathsf{gen}_i(M^A))^{\forall i.A}$, then $\Theta;\Gamma;\Delta \vdash M^A \,|\, s$ is derivable for some s, $i \notin \mathsf{FIV}(\Theta,\Gamma\Delta)$, $r = \mathsf{gen}_i(s)$ and $D = \forall i.A$.
- if $N^D = (\mathsf{ins}_i^E(M^{\forall i.A}))^{A\{i\leftarrow E\}}$, then $\Theta;\Gamma;\Delta \vdash M^{\forall i.A} \,|\, s$ is derivable for some s, $r = \mathsf{ins}_i^E(s)$ and $D = A\{i \leftarrow E\}$.

Proof.- By structural induction on N. One and only one rule applies to each of the cases, and the only rule which can modify the witness is T-□I, which replaces it by an equivalent proof witness for the same type under the same contexts. □

5.3.1 Reduction

Reduction in λ^{FOLP} is defined by the following rules.

Principal rules:

$$
\begin{array}{lll}
\beta: & (\lambda x^A.M^B)N^A & \rightarrow\ M^B\{x^A \leftarrow N^A\} \\
\mu: & [\beta^A]\mu\alpha^A.M^\perp & \rightarrow\ M^\perp\{\alpha^A \leftarrow \beta^A\} \\
\zeta: & (\mu\alpha^{A\supset B}.M^\perp)N^A & \rightarrow\ \mu\beta^B.M^\perp(\!|\, [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)N^A \,|\!) \\
\theta: & \mu\alpha^A.[\alpha^A]M^A & \rightarrow\ M^A \ \text{ if } \alpha^A \notin \mathsf{FVF}(M^A) \\
\gamma: & M^B\langle v^A_\Xi{:=}r, !N^A\rangle & \rightarrow\ M^B\{v^A_\Xi \leftarrow N^A, r\} \ \text{ if } \mathsf{FVT}(N^A) = \mathsf{FVF}(N^A) = \emptyset \\
\xi: & \mathsf{ins}_i^E(\mathsf{gen}_i(M^A)) & \rightarrow\ (M^A)\{i \leftarrow E\}
\end{array}
$$

Permutative rules:

$$
\begin{array}{lll}
\psi_L: & (M^{A\supset B} +_{\mathrm{L}} t)^{A\supset B}N^A & \rightarrow\ (M^{A\supset B}N^A)^B +_{\mathrm{L}} t \\
\psi_R: & (s +_{\mathrm{R}} M^{A\supset B})^{A\supset B}N^A & \rightarrow\ s +_{\mathrm{R}} (M^{A\supset B}N^A)^B \\
\chi_L: & [\beta^A](M^A +_{\mathrm{L}} t)^A & \rightarrow\ ([\beta^A]M^A)^\perp +_{\mathrm{L}} t \\
\chi_R: & [\beta^B](s +_{\mathrm{R}} N^B)^B & \rightarrow\ s +_{\mathrm{R}} ([\beta^B]N^B)^\perp \\
\phi_L: & M^B\langle v^A_\Xi{:=}r, (N^{[\![r]\!]_{\Xi^A}} +_{\mathrm{L}} t)\rangle & \rightarrow\ (M^B\langle v^A_\Xi{:=}r, N^{[\![r]\!]_{\Xi^A}}\rangle)^{B\{v^A_{\Xi'}\leftarrow r^A\}} +_{\mathrm{L}} t \\
\phi_R: & M^B\langle v^A_{\Xi'}{:=}r, (s +_{\mathrm{R}} N^{[\![r]\!]_{\Xi^A}})\rangle & \rightarrow\ s +_{\mathrm{R}} (M^B\langle v^A_{\Xi'}{:=}r, N^{[\![r]\!]_{\Xi^A}}\rangle)^{B\{v^A_{\Xi'}\leftarrow r^A\}} \\
\iota_L: & \mu\alpha^A.(M^\perp +_{\mathrm{L}} t)^\perp & \rightarrow\ (\mu\alpha^A.M^\perp) +_{\mathrm{L}} t, \ \text{ if } \alpha^A \notin \mathsf{FVF}(t) \\
\iota_R: & \mu\alpha^A.(s +_{\mathrm{R}} N^\perp)^\perp & \rightarrow\ s +_{\mathrm{R}} (\mu\alpha^A.N^\perp), \ \text{ if } \alpha^A \notin \mathsf{FVF}(s) \\
\epsilon_L: & \mathsf{ins}_i^E(M^{\forall i.A} +_{\mathrm{L}} t) & \rightarrow\ \mathsf{ins}_i^E(M^{\forall i.A}) +_{\mathrm{L}} t \\
\epsilon_R: & \mathsf{ins}_i^E(s +_{\mathrm{R}} M^{\forall i.A}) & \rightarrow\ s +_{\mathrm{R}} \mathsf{ins}_i^E(M^{\forall i.A})
\end{array}
$$

The rules γ, ϕ_L and ϕ_R are modified versions of their λ^{LP} counterparts. The rules ξ, ϵ_L and ϵ_R are exclusive to this extension. All other rules are the same as in λ^{LP}.

Note that the new rules do not introduce any critical pairs. This means that FOHLP has the same critical pairs as HLP, and these can all be closed as seen in Section 4.6.

Lemma 5.3.18 If the judgment $\Theta;\Gamma;\Delta \vdash M^D \,|\, s$ is derivable and $M^D \to N^D$ by reducing a redex at the root of M^D, then $\Theta;\Gamma;\Delta \vdash N^D \,|\, s'$ is derivable for some witness s' such that $\Theta;\Gamma;\Delta \vdash s \equiv s' : D$.

Proof.- We must consider which reduction rule was used. We will analyze the rules which differ from those of HLP.

- γ: $M^D = M_1^B\langle v^A_{\Xi'} := r, (!M_2^A)^{[\![r]\!]_{\Xi A}}\rangle$, $N^D = M_1^B\{v^A_{\Xi'} \leftarrow M_2^A, r\}$ and $\mathsf{FVT}(M_2^A) = \mathsf{FVF}(M_2^A) = \emptyset$. By Inversion Lemma (twice), $D = B\{v^A \leftarrow r\}$, $s = !r\langle v^A_{\Xi'} := r, t\rangle$ and there is a witness r' such that $\mathsf{FIV}(\Theta) \cap \mathsf{FIV}(A) \subseteq \Xi$, $\Xi \cap \mathsf{FIV}(A) \subseteq \Xi'$, and $\Theta;\cdot;\cdot \vdash r' \equiv r : A$ as well as $\Theta;\cdot;\cdot \vdash M_2^A \,|\, r'$ and $\Theta, v^A_{\Xi'};\Gamma;\Delta \vdash M_1^B \,|\, t$ are derivable.

  By Lemma 5.3.7, we can derive $\Theta;\Gamma;\Delta \vdash M_1^B\{v^A_{\Xi'} \leftarrow M_2^A, r\}^{B\{v^A_{\Xi'} \leftarrow r\}} \,|\, s'$ and $\Theta;\Gamma;\Delta \vdash s' \equiv t\{v^A_{\Xi'} \leftarrow r\} : B\{v^A_{\Xi'} \leftarrow r\}$ for some witness s'; and, by Eq-Symm, we derive $\Theta;\Gamma;\Delta \vdash t\{v^A_{\Xi'} \leftarrow r\} \equiv s' : B\{v^A_{\Xi'} \leftarrow r\}$. By Eq-$\gamma$ -since $\mathsf{FIV}(\Theta) \cap \mathsf{FIV}(A) \subseteq \Xi'$-, we can derive $\Theta;\Gamma;\Delta \vdash s \equiv t\{v^A_{\Xi'} \leftarrow r\} : B\{v^A_{\Xi'} \leftarrow r\}$. And finally, by Eq-Trans, $\Theta;\Gamma;\Delta \vdash s \equiv s' : B\{v^A_{\Xi'} \leftarrow r\}$.
- ξ: $M^D = \mathsf{ins}_i^E(\mathsf{gen}_i(M_1^A))$, $N^D = (M_1^A)\{i \leftarrow E\}$, and by Inversion Lemma (twice), $D = \{i \leftarrow E\}$, $s = \mathsf{ins}_i^E(\mathsf{gen}_i(r))$ and we can derive $\Theta;\Gamma;\Delta \vdash M_1^A \,|\, r$, where $i \notin \mathsf{FIV}(\Theta,\Gamma,\Delta)$. By Lemma 5.3.12, we can derive $\Theta\{i \leftarrow E\};\Gamma\{i \leftarrow E\};\Delta\{i \leftarrow E\} \vdash M_1\{i \leftarrow E\}^{A\{i \leftarrow E\}} \,|\, r\{i \leftarrow E\}$, which is the same as $\Theta;\Gamma;\Delta \vdash M_1\{i \leftarrow E\}^{A\{i \leftarrow E\}} \,|\, r\{i \leftarrow E\}$ since $i \notin \mathsf{FIV}(\Theta,\Gamma,\Delta)$. And, by Eq-$\xi$, we can derive $\Theta;\Gamma;\Delta \vdash \mathsf{ins}_i^E(\mathsf{gen}_i(r)) \equiv r\{i \leftarrow E\} : A\{i \leftarrow E\}$.
- ϕ_L: $M = M_1^B\langle v^A_{\Xi'} := r, (M_2^{[\![r]\!]_{\Xi A}} +_{\mathrm{L}} t)\rangle$, $N = (M_1^B\langle v^A_{\Xi'} := r, M_2^{[\![r]\!]_{\Xi A}}\rangle)^{B\{v^A_{\Xi'} \leftarrow r^A\}} +_{\mathrm{L}} t$. By Inversion Lemma (twice), $D = B\{v^A_{\Xi'} \leftarrow r\}$, $s = s_1\langle v^A_{\Xi'} := r, (s_2 + t)\rangle$, $\Xi \cap \mathsf{FIV}(A) \subseteq \Xi'$, and the judgements $\Theta, v^A_{\Xi'};\Gamma;\Delta \vdash M_1^B \,|\, s_1$, $\Theta;\Gamma;\Delta \vdash (M_2^{[\![r]\!]_{\Xi A}} +_{\mathrm{L}} s_2)^{[\![r]\!]_{\Xi A}} \,|\, s_2 + t$ and $\Theta;\Gamma;\Delta \vdash M_2^{[\![r]\!]_{\Xi A}} \,|\, s_2$ are derivable. By T-$\Box$E, we can derive $\Theta;\Gamma;\Delta \vdash (M_1^B\langle v^A_{\Xi'} := r, M_2^{[\![r]\!]_{\Xi A}}\rangle)^{B\{v^A_{\Xi'} \leftarrow r\}} \,|\, s_1\langle v^A_{\Xi'} := r, s_2\rangle$. By T-PlusL, the judgment $\Theta;\Gamma;\Delta \vdash N^{B\{v^A_{\Xi'} \leftarrow r\}} \,|\, (s_1\langle v^A_{\Xi'} := r, s_2\rangle) + t$ is derivable. And, by Eq-ϕ_L, so is $\Theta;\Gamma;\Delta \vdash s_1\langle v^A_{\Xi'} := r, (s_2 + t)\rangle \equiv (s_1\langle v^A_{\Xi'} := r, s_2\rangle) + t : B\{v^A_{\Xi'} \leftarrow r\}$.
- ϕ_R: this case is analogous to the previous one.
- ϵ_L: $M^D = \mathsf{ins}_i^E(M_1^{\forall i.A} +_{\mathrm{L}} t)^{A\{i \leftarrow E\}}$, $N^D = (\mathsf{ins}_i^E(M^{\forall i.A})^{A\{i \leftarrow E\}} +_{\mathrm{L}} t)^{A\{i \leftarrow E\}}$ and by Inversion Lemma (twice), $D = A\{i \leftarrow E\}$, $s = \mathsf{ins}_i^E(r + t)$ and $\Theta;\Gamma;\Delta \vdash M_1^{\forall i.A} \,|\, r$ is derivable.

  By T-$\forall$E, we can derive $\Theta;\Gamma;\Delta \vdash \mathsf{ins}_i^E(M_1^{\forall i.A})^{A\{i \leftarrow E\}} \,|\, \mathsf{ins}_i^E(r)$. By T-PlusL, we obtain $\Theta;\Gamma;\Delta \vdash (\mathsf{ins}_i^E(M_1^{\forall i.A})^{A\{i \leftarrow E\}} +_{\mathrm{L}} t)^{A\{i \leftarrow E\}} \,|\, \mathsf{ins}_i^E(r) + t$. And, by Eq-$\epsilon_L$, we can derive $\Theta;\Gamma;\Delta \vdash \mathsf{ins}_i^E(r + t) \equiv \mathsf{ins}_i^E(r) + t : A\{i \leftarrow E\}$.
- ϵ_R: this case is analogous to the previous one.

$\Box$

Lemma 5.3.19 (Type Preservation) If $\Theta;\Gamma;\Delta \vdash M^B | s$ is derivable and $M^B \to N^B$, then $\Theta;\Gamma;\Delta \vdash N^B \,|\, s'$ is derivable for some witness s' such that $\Theta;\Gamma;\Delta \vdash s \equiv s' : B$.

Proof.- By induction on M^B. If the reduction takes place on the root, then the result holds by Lemma 5.3.18. We will now consider all other cases.

- If $M^B = x^B$ or $M^B = v^B_\Xi$, no reduction may take place, thus the result holds trivially.
- If $M^B = (\lambda x^A.M_1^C)^{A\supset C}$: then $B = A \supset C$ and $N^B = (\lambda x^A.N_1^C)^{B'}$, where $M_1^C \to N_1^C$. By Inversion Lemma, $\Theta;\Gamma,x^A;\Delta \vdash M_1^C \,|\, t$ is derivable for some t, and $s = \lambda x^A.t$. Then, by IH, $\Theta;\Gamma,x^A;\Delta \vdash N_1^C \,|\, t'$ is derivable, where $\Theta;\Gamma,x^A;\Delta \vdash t \equiv t' : C$. By Eq-$\lambda$, we obtain $\Theta;\Gamma;\Delta \vdash \lambda x^A.t \equiv \lambda x^A.t' : A \supset C$. And, by T-$\supset$I, $\Theta;\Gamma;\Delta \vdash (\lambda x^A.N_1^C)^{A\supset C} \,|\, \lambda x^A.t'$.
- If $M^B = (M_1^{A\supset B} M_2^A)^B$. By Inversion Lemma, $s = s_1 \cdot s_2$ and both $\Theta;\Gamma;\Delta \vdash M_1^{A\supset B} \,|\, s_1$ and $\Theta;\Gamma;\Delta \vdash M_2^A \,|\, s_2$ are derivable. There are two possibilities:
  - $N^B = (N_1^{A\supset B} M_2^A)^B$, with $M_1^{A\supset B} \to N_1^{A\supset B}$: by IH, we can derive $\Theta;\Gamma;\Delta \vdash N_1^{A\supset B} \,|\, s_1'$ and $\Theta;\Gamma;\Delta \vdash s_1 \equiv s_1' : A \supset B$. We can obtain $\Theta;\Gamma;\Delta \vdash (N_1^{A\supset B} M_2^A)^B \,|\, s_1' \cdot s_2$ by T-$\supset$E, and $\Theta;\Gamma;\Delta \vdash s_1 \cdot s_2 \equiv s_1' \cdot s_2 : B$ by Eq-Refl and Eq-$\cdot$.
  - $N^B = (M_1^{A\supset B} N_2^A)^B$, with $M_2^A \to N_2^A$: this case is analogous to the previous one.
- If $M^B = (!M_1^A)^{[\![r]\!]_\Xi A}$: in this case $B = [\![r]\!]_\Xi A$, $N^B = (!N_1^A)^{[\![r]\!]_\Xi A}$ where $M_1^A \to N_1^A$ and, by Inversion Lemma, there is a witness t such that both $\Theta;\cdot;\cdot \vdash M_1^A \,|\, t$ and $\Theta;\cdot;\cdot \vdash t \equiv r : A$ are derivable, $\mathsf{FIV}(\Theta) \cap \mathsf{FIV}(A) \subseteq \Xi$ and $s = !t$.

  By IH, we can derive $\Theta;\cdot;\cdot \vdash N_1^A \,|\, t'$ and $\Theta;\cdot;\cdot \vdash t \equiv t' : A$.

  By Eq-Symm and Eq-Trans, we obtain $\Theta;\cdot;\cdot \vdash t' \equiv r : A$. And, by T-$\Box$I, $\Theta;\Gamma;\Delta \vdash (!N_1^A)^{[\![r]\!]_\Xi A} \,|\, !r$. $\Theta;\Gamma;\Delta \vdash t' \equiv r : A$ is obtained by Weakening.
- If $M^B = (M_1^C \langle v^A_{\Xi'} := r, M_2^{[\![r]\!]_\Xi A} \rangle)^{C\{v^A_{\Xi'} \leftarrow r\}}$: then $B = C\{v^A_{\Xi'} \leftarrow r\}$ and, by Inversion Lemma, $\Xi \cap \mathsf{FIV}(A) \subseteq \Xi'$, both $\Theta;\Gamma;\Delta \vdash M_2^{[\![r]\!]_\Xi A} \,|\, s_1$ and $\Theta, v^A;\Gamma;\Delta \vdash M_2^B \,|\, s_2$ are derivable for some proof witnesses s_1 and s_2, and $s = s_1 \langle v^A_{\Xi'} := r, s_2 \rangle$. There are two possibilities:
  - $N^B = (M_1^C \langle v^A_{\Xi'} := r, N_2^{[\![r]\!]_\Xi A} \rangle)^{C\{v^A_{\Xi'} \leftarrow r\}}$ with $M_2^{[\![r]\!]_\Xi A} \to N_2^{[\![r]\!]_\Xi A}$.
  - $N^B = (N_1^C \langle v^A_{\Xi'} := r, M_2^{[\![r]\!]_\Xi A} \rangle)^{C\{v^A_{\Xi'} \leftarrow r\}}$ with $M_1^C \to N_1^C$.

  In both cases the result follows by IH, T-$\Box$E, Eq-Refl and Eq-$\langle\rangle$.
- If $M^B = ([\alpha^A]M_1^A)^\perp$: $B = \perp$, $N^B = ([\alpha^A]N_1^A)^\perp$ where $M_1^A \to N_1^A$ and, by Inversion Lemma, $s = [\alpha^A]s_1$, $\Delta = \Delta', \alpha^A$ and $\Theta;\Gamma;\Delta', \alpha^A \vdash M_1^A \,|\, s_1$ is derivable. By IH, we can derive $\Theta;\Gamma;\Delta', \alpha^A \vdash M_1^A \,|\, s_1'$ and $\Theta;\Gamma;\Delta', \alpha^A \vdash s_1 \equiv s_1' : A$. The result follows by T-Name and Eq-$[\alpha]$.
- If $M^B = (\mu\alpha^B.M_1^\perp)^B$: $N^B = (\mu\alpha^A.N_1^\perp)^B$ where $M_1^\perp \to N_1^\perp$ and, by Inversion Lemma, $\Theta;\Gamma;\Delta, \alpha^A \vdash M_1^\perp \,|\, s_1$ is derivable for some s_1, and $s = \mu\alpha^A.s_1$. By IH, we can derive $\Theta;\Gamma;\Delta, \alpha^A \vdash N_1^\perp \,|\, s_1'$ and $\Theta;\Gamma;\Delta', \alpha^A \vdash s_1 \equiv s_1' : A$. The result follows by T-NAbs and Eq-$\mu\alpha$.
- If $M^B = (M_1^B +_{\mathrm{L}} r)^B$: $N^B = (N_1^B +_{\mathrm{L}} r)^B$ where $M_1^B \to N_1^B$ and, by Inversion Lemma, $s = s_1 + r$ and $\Theta;\Gamma;\Delta \vdash M_1^B \,|\, s_1$ is derivable. The result follows by IH, T-PlusL and Eq-$+_{\mathrm{L}}$.
- If $M^B = (r +_{\mathrm{R}} M_2^B)^B$: this case is analogous to the previous one.
- If $M^B = \mathsf{gen}_i(M_1^A)^{\forall i.A}$, then $B = \forall i.A$ and $N^B = \mathsf{gen}_i(N_1^A)^{\forall i.A}$, where $M_1^A \to N_1^A$. By Inversion Lemma, $\Theta;\Gamma;\Delta \vdash M_1^A \,|\, t$ is derivable for some t, $i \notin \mathsf{FIV}(\Theta,\Gamma,\Delta)$ and $s = \mathsf{gen}_i(t)$.

Then, by IH, $\Theta;\Gamma;\Delta \vdash N_1^A \,|\, t'$ is derivable, where $\Theta;\Gamma;\Delta \vdash t \equiv t' : A$. By Eq-gen, we obtain $\Theta;\Gamma;\Delta \vdash \mathsf{gen}_i(t) \equiv \mathsf{gen}_i(t') : \forall i.A$. And, by T-∀I, $\Theta;\Gamma;\Delta \vdash \mathsf{gen}_i(N_1^A)^{\forall i.A} \,|\, \mathsf{gen}_i(t')$.

- If $M^B = \mathsf{ins}_i^E(M_1^{\forall i.A})^{A\{\cancel{i} \leftarrow E\}}$, then $B = A\{i \leftarrow E\}$ and $N^B = \mathsf{ins}_i^E(N_1^{\forall i.A})^{A\{\cancel{i} \leftarrow E\}}$, where $M_1^{\forall i.A} \to N_1^{\forall i.A}$. By Inversion Lemma, $\Theta;\Gamma;\Delta \vdash M_1^{\forall i.A} \,|\, t$ is derivable for some t, and $s = \mathsf{ins}_i^E(t)$. Then, by IH, $\Theta;\Gamma;\Delta \vdash N_1^{\forall i.A} \,|\, t'$ is derivable, where $\Theta;\Gamma;\Delta \vdash t \equiv t' : \forall i.A$. By Eq-ins, we obtain $\Theta;\Gamma;\Delta \vdash \mathsf{ins}_i^E(t) \equiv \mathsf{ins}_i^E(t') : A\{i \leftarrow E\}$. And, by T-∀E, $\Theta;\Gamma;\Delta \vdash \mathsf{ins}_i^E(N_1^{\forall i.A})^{A\{\cancel{i} \leftarrow E\}} \,|\, \mathsf{ins}_i^E(t')$.

□

Corollary 5.3.20 If $\Theta;\Gamma;\Delta \vdash (!M^B)^A | t$ is derivable and $M^B \to N^B$, then $\Theta;\Gamma;\Delta \vdash (!N^B)^A \,|\, t$ is derivable.

Proof.- By Inversion Lemma, $A = [\![r]\!]_\Xi B$, $t = !r$ for some proof witness r, and there is an s such that both $\Theta;\cdot;\cdot \vdash M^B | s$ and $\Theta;\cdot;\cdot \vdash s \equiv r : B$ are derivable. By Lemma 5.3.19, there is an s' such that both $\Theta;\cdot;\cdot \vdash N^B | s'$ and $\Theta;\cdot;\cdot \vdash s' \equiv s : B$ are derivable, and $\Theta \cap \mathsf{FIV}(A) \subseteq \Xi$. By Eq-Trans, $\Theta;\cdot;\cdot \vdash s' \equiv r : B$ is also derivable. And, by T-□I, so is $\Theta;\Gamma;\Delta \vdash (!N^B)^{[\![r]\!]_\Xi B} \,|\, !r$. □

Lemma 5.3.21 If $\rhd_{\lambda^{\mathsf{FOLP}}} \Theta;\Gamma;\Delta \vdash s \equiv t : B$, then there are terms M_1, M_2, M_3 s.t. :

1. $\rhd_{\lambda^{\mathsf{FOLP}}} \Theta;\Gamma;\Delta \vdash M_1^B \,|\, s$;
2. $\rhd_{\lambda^{\mathsf{FOLP}}} \Theta;\Gamma;\Delta \vdash M_2^B \,|\, t$;
3. $M_1 \twoheadrightarrow M_3$; and
4. $M_2 \twoheadrightarrow M_3$.

Proof.- By induction on the derivation of $\Theta;\Gamma;\Delta \vdash s \equiv t : B$. We analyze the last rule used. For all rules not listed here, the proof is analogous to the one shown for Lemma 4.6.2. As in the propositional case, we will work with *canonical* derivations.

- Eq-γ: $s = t'\langle v_\Xi^A := s', !s'\rangle$, $t = t'\{v_\Xi^A \leftarrow s'\}$, $B = C\{v_\Xi^A \leftarrow s'\}$, both $\Theta;\cdot;\cdot \vdash N^A \,|\, s'$ and $\Theta, v_\Xi^A;\Gamma;\Delta \vdash M^C \,|\, t'$ are derivable by hypothesis for some M^C and N^A, and $\mathsf{FIV}(\Theta) \cap \mathsf{FIV}(A) \subseteq \Xi$.
  Let $\Xi_1 = \mathsf{FIV}(\Theta)$. Since $\Xi_1 \cap \mathsf{FIV}(A) \subseteq \Xi_1$, then by T-□I, we get $\Theta;\Gamma;\Delta \vdash (!N^A)^{[\![s']\!]_{\Xi_1} A} \,|\, !s'$ – note that we use the same proof witness s' on both sides of the equivalence, which is the same as using T-□I', and thus we keep the derivation canonical. Now, by T-□E, we can derive $\Theta;\Gamma;\Delta \vdash M^C\langle v_\Xi^A := s', (!N^A)^{[\![s']\!]_{\Xi_1} A}\rangle^{C\{v_\Xi^A \leftarrow s'\}} \,|\, t'\langle v_\Xi^A := s', !s'\rangle$.
  Take $M_1 = M^C\langle v_\Xi^A := s', (!N^A)^{[\![s']\!]_{\Xi_1} A}\rangle$ and $M_2 = M_3 = M^C\{v_\Xi^A \leftarrow N^A, s'\}$.
  The judgement $\Theta;\Gamma;\Delta \vdash M_2^{C\{v_\Xi^A \leftarrow s'\}} \,|\, t'\{v_\Xi^A \leftarrow s'\}$ can be obtained from $\Theta, v_\Xi^A;\Gamma;\Delta \vdash C \,|\, t'$ and the hypotheses by Lemma 5.2.12, and $M_1 \to_\gamma M_2$.
- Eq-ξ: $s = \mathsf{ins}_i^E(\mathsf{gen}_i(s'))$, $t = s'\{i \leftarrow E\}$, $B = A\{i \leftarrow E\}$, $\Theta;\Gamma;\Delta \vdash M^A \,|\, s'$ is derivable by hypothesis for some M^A, and $i \notin \mathsf{FIV}(\Theta, \Gamma, \Delta)$.
  Take $M_1 = \mathsf{ins}_i^E((\mathsf{gen}_i(M^A))^{\forall i.A})$ and $M_2 = M_3 = M^A\{i \leftarrow E\}$.

By T-∀I, we can derive $\Theta;\Gamma;\Delta \vdash \mathsf{gen}_i(M^A)^{\forall i.A} \,|\, \mathsf{gen}_i(s')$. And then, by T-∀E, we obtain $\Theta;\Gamma;\Delta \vdash M_1^{A\{i\leftarrow E\}} \,|\, \mathsf{ins}_i^E(\mathsf{gen}_i(s'))$.

$\Theta;\Gamma;\Delta \vdash M_2 \,|\, A\{i \leftarrow E\}s'\{i \leftarrow E\}$ can be obtained from $\Theta;\Gamma;\Delta \vdash A \,|\, s'$ by Lemma 5.2.13, and $M_1 \rightarrow_\xi M_2$.

- Eq-ϕ_L: $s = u\langle v^A_{\Xi'}{:=}r, (s' + t')\rangle$, $t = u\langle v^A_{\Xi'}{:=}r, s'\rangle + t'$, $B = C\{v^A_{\Xi'} \leftarrow r\}$, both $\Theta;\Gamma;\Delta \vdash M^{[\![r]\!]_{\Xi A}} \,|\, s'$ and $\Theta, v^A_{\Xi'};\Gamma;\Delta \vdash N^C \,|\, u$ are derivable by hypothesis for some $M^{[\![r]\!]_{\Xi A}}$ and N^C, and $\Xi \cap \mathsf{FIV}(A) \subseteq \Xi'$.

  Take $M_1 = N^C\langle v^A_{\Xi'}{:=}r, (M^{[\![r]\!]_{\Xi A}} +_{\mathrm{L}} t')\rangle$ and $M_2 = M_3 = N^C\langle v^A_{\Xi'}{:=}r, M^{[\![r]\!]_{\Xi A}}\rangle +_{\mathrm{L}} t'$.

  $\Theta;\Gamma;\Delta \vdash M_1^B \,|\, s$ is obtained by applying T-PlusL and then T-□E; conversely, $\Theta;\Gamma;\Delta \vdash M_2^B \,|\, t$ results from applying T-□E and then T-PlusL; and $M_1 \rightarrow_{\phi_L} M_2$.

- Eq-ϕ_R: this case is analogous to the previous one.

- Eq-ϵ_L: $s = \mathsf{ins}_i^E(s' + t')$, $t = \mathsf{ins}_i^E(s') + t'$, $B = A\{i \leftarrow E\}$ and $\Theta;\Gamma;\Delta \vdash M^{\forall i.A} \,|\, s'$ is derivable by hypothesis for some $M^{\forall i.A}$.

  Take $M_1 = \mathsf{ins}_i^E(M^{\forall i.A} +_{\mathrm{L}} t')$ and $M_2 = M_3 = \mathsf{ins}_i^E(M^{\forall i.A}) +_{\mathrm{L}} t'$.

  $\Theta;\Gamma;\Delta \vdash M_1^B \,|\, s$ is obtained by applying T-PlusL and then T-∀E; conversely, $\Theta;\Gamma;\Delta \vdash M_2^B \,|\, t$ results from applying T-∀E and then T-PlusL; and $M_1 \rightarrow_{\epsilon_L} M_2$.

- Eq-ϵ_R: this case is analogous to the previous one.

- Eq-$\langle\rangle$: $s = t_1\langle v^A_{\Xi'}{:=}r, s_1\rangle$, $t = t_2\langle v^A_{\Xi'}{:=}r, s_2\rangle$, $B = C\{v^A_{\Xi'} \leftarrow r\}$, both $\Theta;\Gamma;\Delta \vdash s_1 \equiv s_2 : [\![r]\!]_{\Xi A}$ and $\Theta, v^A_{\Xi'};\Gamma;\Delta \vdash t_1 \equiv t_2 : C$ are derivable by hypothesis, and $\Xi \cap \mathsf{FIV}(A) \subseteq \Xi'$. By induction hypothesis, we can derive:

  - $\Theta;\Gamma;\Delta \vdash N^{[\![r]\!]_{\Xi A}} \,|\, s_1$
  - $\Theta, v^A_{\Xi'};\Gamma;\Delta \vdash M^C \,|\, t_1$
  - $\Theta;\Gamma;\Delta \vdash N'^{[\![r]\!]_{\Xi A}} \,|\, s_2$
  - $\Theta, v^A_{\Xi'};\Gamma;\Delta \vdash M'^C \,|\, t_2$
  - $M^C \twoheadrightarrow M_4$, $M'^C \twoheadrightarrow M_4$, $N^{[\![r]\!]_{\Xi A}} \twoheadrightarrow M_5$, $N'^{[\![r]\!]_{\Xi A}} \twoheadrightarrow M_5$.

  Take $M_1 = M^C\langle v^A_{\Xi'}{:=}r, N^{[\![r]\!]_{\Xi A}}\rangle$, $M_2 = M'^C\langle v^A_{\Xi'}{:=}r, N'^{[\![r]\!]_{\Xi A}}\rangle$ and $M_3 = M_4\langle v^A_{\Xi'}{:=}r, M_5^{[\![r]\!]_{\Xi A}}\rangle$. $\Theta;\Gamma;\Delta \vdash M_1^B \,|\, s$ and $\Theta;\Gamma;\Delta \vdash M_2^B \,|\, s$ are obtained by T-□E, and $M_1 \twoheadrightarrow M_3$ and $M_2 \twoheadrightarrow M_3$ by means of internal reductions.

- Eq-gen: the results are obtained by induction hypothesis, T-∀I and internal reductions.

- Eq-ins: the results are obtained by induction hypothesis, T-∀E and internal reductions.

□

Corollary 5.3.22 The $\equiv$ theory for FOHLP is consistent.

Proof.- The proof is analogous to that of Corollary 4.6.3, using Lemma 5.3.21 and Remark 5.3.4. □

5.3.2 Normalisation

In order to prove strong normalisation of term reduction, we adapt the mapping defined in Section 4.5 as follows: individual variables and expressions are ignored, and we introduce new mapping rules for $\forall i.A$, $\mathsf{gen}_i(M^A)$ and $\mathsf{ins}_i^E(M^A)$ All other rules are analogous to those for the propositional case, ignoring the sets of individual variables.

Definition 5.3.23 (From λ^{FOLP} to $\lambda\mu^{\mathbf{1}}$) We first translate types:

$$\begin{aligned}
\langle\!| P(E) |\!\rangle &\triangleq P \\
\langle\!| \bot |\!\rangle &\triangleq \bot \\
\langle\!| A \supset B |\!\rangle &\triangleq \langle\!| A |\!\rangle \supset \langle\!| B |\!\rangle \\
\langle\!| [\![s]\!]_{\Xi} A |\!\rangle &\triangleq \mathbf{1} \supset \langle\!| A |\!\rangle \\
\langle\!| \forall i.A |\!\rangle &\triangleq \mathbf{1} \supset \langle\!| A |\!\rangle
\end{aligned}$$

For terms we translate as follows:

$$\begin{aligned}
\langle\!| x^A |\!\rangle &\triangleq x^{\langle\!| A |\!\rangle} \\
\langle\!| v_{\Xi}^A |\!\rangle &\triangleq (x_v^{\mathbf{1}\supset\langle\!| A |\!\rangle})\mathtt{unit} \\
\langle\!| (\lambda x^A.M^B)^{A\supset B} |\!\rangle &\triangleq \lambda x^{\langle\!| A |\!\rangle}.\langle\!| M^B |\!\rangle \\
\langle\!| (M^{A\supset B} N^B)^B |\!\rangle &\triangleq \langle\!| M^{A\supset B} |\!\rangle \langle\!| N^B |\!\rangle \\
\langle\!| ([\alpha^A]M^A)^{\bot} |\!\rangle &\triangleq [\alpha^{\langle\!| A |\!\rangle}]\langle\!| M^A |\!\rangle \\
\langle\!| (\mu\alpha^A.M^{\bot})^A |\!\rangle &\triangleq \mu\alpha^{\langle\!| A |\!\rangle}.\langle\!| M^{\bot} |\!\rangle \\
\langle\!| (!M^A)^{[\![s]\!]_{\Xi}A} |\!\rangle &\triangleq \lambda x^{\mathbf{1}}.\langle\!| M^A |\!\rangle x^{\mathbf{1}} \text{ fresh} \\
\langle\!| (M^B\langle v_{\Xi'}^A := r, N^{[\![r]\!]_{\Xi}A}\rangle)^{B\{v_{\Xi'}^A \leftarrow r\}} |\!\rangle &\triangleq (\lambda x_v^{\mathbf{1}\supset\langle\!| A |\!\rangle}.\langle\!| M^B |\!\rangle)\langle\!| N^{[\![r]\!]_{\Xi}A} |\!\rangle \\
\langle\!| (M^A +_{\mathrm{L}} t)^A |\!\rangle &\triangleq \langle\!| M^A |\!\rangle \\
\langle\!| (s +_{\mathrm{R}} M^A)^A |\!\rangle &\triangleq \langle\!| M^A |\!\rangle \\
\langle\!| \mathsf{gen}_i(M^A) |\!\rangle &\triangleq \lambda x_i^{\mathbf{1}}.\langle\!| M^A |\!\rangle \\
\langle\!| \mathsf{ins}_i^E(M^{\forall i.A}) |\!\rangle &\triangleq \langle\!| M^{\forall i.A} |\!\rangle \mathtt{unit}
\end{aligned}$$

Lemma 5.3.24 For every formula C, validity variable v_{Ξ}^A, individual variable i, expression E and proof witness r, $\langle\!| C\{v_{\Xi}^A \leftarrow r\} |\!\rangle = \langle\!| C\{i \leftarrow E\} |\!\rangle = \langle\!| C |\!\rangle$.

Proof.- By structural induction on C. The only interesting cases are the following:

- $C = [\![s]\!]_{\Xi'}B$: $\langle\!| C\{v_{\Xi}^A \leftarrow r\} |\!\rangle = \langle\!| [\![s\{v_{\Xi}^A \leftarrow r\}]\!]_{\Xi'}B\{v_{\Xi}^A \leftarrow r\} |\!\rangle = \mathbf{1} \supset \langle\!| B\{v_{\Xi}^A \leftarrow r\} |\!\rangle =_{\mathrm{IH}} \mathbf{1} \supset \langle\!| B |\!\rangle = \langle\!| C |\!\rangle$. As for $\langle\!| C\{i \leftarrow E\} |\!\rangle$, there are two possibilities:
  - If $i \in \Xi'$, then
    $$\begin{aligned}
    &\langle\!| C\{i \leftarrow E\} |\!\rangle && = \\
    &\langle\!| [\![s\{i \leftarrow E\}]\!]_{\Xi' \setminus \{i\} \cup \mathsf{FIV}(E)} B\{i \leftarrow E\} |\!\rangle && = \\
    &\mathbf{1} \supset \langle\!| B\{i \leftarrow E\} |\!\rangle && =_{\mathrm{IH}} \\
    &\mathbf{1} \supset \langle\!| B |\!\rangle && = \\
    &\langle\!| C |\!\rangle.
    \end{aligned}$$
  - If $i \notin \Xi'$, then $C\{i \leftarrow E\} = C$ and thus $\langle\!| C\{i \leftarrow E\} |\!\rangle = \langle\!| C |\!\rangle$.

- $C = \forall j.B$: in this case, $\langle\!| C\{v^A_\Xi \leftarrow r\} |\!\rangle = \mathbf{1} \supset \langle\!| B\{v^A_\Xi \leftarrow r\} |\!\rangle =_{\mathrm{IH}} \mathbf{1} \supset \langle\!| B |\!\rangle = \langle\!| C |\!\rangle$.
  As for $\langle\!| C\{i \leftarrow E\} |\!\rangle$, there are two possibilities:
  - If $i = j$, then $C\{i \leftarrow E\} = C$ and thus $\langle\!| C\{i \leftarrow E\} |\!\rangle = \langle\!| C |\!\rangle$.
  - If $i \neq j$, then $\langle\!| C\{i \leftarrow E\} |\!\rangle = \mathbf{1} \supset \langle\!| B\{i \leftarrow E\} |\!\rangle =_{\mathrm{IH}} \mathbf{1} \supset \langle\!| B |\!\rangle = \langle\!| C |\!\rangle$.

□

Lemma 5.3.25 If $\Theta; \Gamma; \Delta \vdash M^A \,|\, s$ is derivable in λ^{FOLP}, then $\langle\!| M |\!\rangle : \langle\!| \Theta |\!\rangle \cup \langle\!| \Gamma |\!\rangle \vdash \langle\!| A |\!\rangle, \langle\!| \Delta |\!\rangle$ is derivable in $\lambda\mu^{\mathbf{1}}$.

Proof.- By induction on the derivation of $\Theta; \Gamma; \Delta \vdash M^A \,|\, s$. We consider the last step of the derivation for the rules which differ from those of HLP.

- Case T-VarM: $M = v^A_\Xi$, $\Theta = \Theta', v^A_\Xi$, $\langle\!| M |\!\rangle = (x_v^{\mathbf{1} \supset \langle\!| A |\!\rangle})\mathtt{unit}$ and $\langle\!| \Theta |\!\rangle = \langle\!| \Theta' |\!\rangle, x_v^{\mathbf{1} \supset \langle\!| A |\!\rangle}$. We can construct the following derivation in $\lambda\mu^{\mathbf{1}}$.

$$\dfrac{\dfrac{}{x_v^{\mathbf{1} \supset \langle\!| A |\!\rangle} : \langle\!| \Theta' |\!\rangle, x_v^{\mathbf{1} \supset \langle\!| A |\!\rangle} \cup \langle\!| \Gamma |\!\rangle \vdash \mathbf{1} \supset \langle\!| A |\!\rangle, \langle\!| \Delta |\!\rangle}\,\mathsf{Ax} \qquad \dfrac{}{\mathtt{unit} : \langle\!| \Theta' |\!\rangle, x_v^{\mathbf{1} \supset \langle\!| A |\!\rangle} \cup \langle\!| \Gamma |\!\rangle \vdash \mathbf{1}, \langle\!| \Delta |\!\rangle}\,\mathsf{Unit}}{(x_v^{\mathbf{1} \supset \langle\!| A |\!\rangle})\mathtt{unit} : \langle\!| \Theta' |\!\rangle, x_v^{\mathbf{1} \supset \langle\!| A |\!\rangle} \cup \langle\!| \Gamma |\!\rangle \vdash \langle\!| A |\!\rangle, \langle\!| \Delta |\!\rangle}\,\supset\mathsf{E}$$

- Case T-□I: $M = (!N^B)^{[\![r]\!]_\Xi B}$, $A = [\![r]\!]_\Xi B$, $\Theta = \Theta_1 \cup \Theta_2$, $\langle\!| M |\!\rangle = \lambda x^{\mathbf{1}}.\langle\!| N^B |\!\rangle$, $\langle\!| A |\!\rangle = \mathbf{1} \supset \langle\!| B |\!\rangle$ and, by Weakening and IH, $\langle\!| N^B |\!\rangle : \langle\!| \Theta |\!\rangle \vdash \langle\!| B |\!\rangle$ is derivable in $\lambda\mu^{\mathbf{1}}$. We can derive $\lambda x^{\mathbf{1}}.\langle\!| N^B |\!\rangle : \langle\!| \Theta |\!\rangle \vdash \mathbf{1} \supset \langle\!| B |\!\rangle$ by Weakening and $\supset$I, and then obtain $\lambda x^{\mathbf{1}}.\langle\!| N^B |\!\rangle : \langle\!| \Theta |\!\rangle \cup \langle\!| \Gamma |\!\rangle \vdash \mathbf{1} \supset \langle\!| B |\!\rangle, \langle\!| \Delta |\!\rangle$ by Weakening.
- Case T-□E: $(M = M_1^C \langle v_r^B := \Xi', M_2^{[\![r]\!]_\Xi B} \rangle)^{C\{v^B_{\Xi'} \leftarrow r\}}$, $A = C\{v^B_{\Xi'} \leftarrow r\}$, $\langle\!| M |\!\rangle = \lambda x_v^{\mathbf{1} \supset \langle\!| B |\!\rangle}.\langle\!| M_2^{[\![r]\!]_\Xi B} |\!\rangle \langle\!| M_1^C |\!\rangle$ and, by Lemma 5.3.24, $\langle\!| A |\!\rangle = \langle\!| C |\!\rangle$. We can construct the following derivation.

$$\dfrac{\dfrac{\dfrac{}{\langle\!| M_1 |\!\rangle : \langle\!| \Theta |\!\rangle, x_v^{\mathbf{1} \supset \langle\!| B |\!\rangle} \cup \langle\!| \Gamma |\!\rangle \vdash \langle\!| C |\!\rangle, \langle\!| \Delta |\!\rangle}\,\mathrm{IH}}{\lambda x_v^{\mathbf{1} \supset \langle\!| B |\!\rangle}.\langle\!| M_1 |\!\rangle : \langle\!| \Theta |\!\rangle \cup \langle\!| \Gamma |\!\rangle \vdash (\mathbf{1} \supset \langle\!| B |\!\rangle) \supset \langle\!| C |\!\rangle, \langle\!| \Delta |\!\rangle}\,\supset\mathsf{I} \qquad \dfrac{}{\langle\!| M_2 |\!\rangle : \langle\!| \Theta |\!\rangle \cup \langle\!| \Gamma |\!\rangle \vdash \mathbf{1} \supset \langle\!| B |\!\rangle, \langle\!| \Delta |\!\rangle}\,\mathrm{IH}}{\langle\!| M |\!\rangle : \langle\!| \Theta |\!\rangle \cup \langle\!| \Gamma |\!\rangle \vdash \langle\!| C |\!\rangle, \langle\!| \Delta |\!\rangle}\,\supset\mathsf{E}$$

- Case T-∀I: $M = \mathsf{gen}_i(N^B)^{\forall i.B}$, $A = \forall i.B$, $\langle\!| M |\!\rangle = \lambda i^{\mathbf{1}}.\langle\!| N^B |\!\rangle$ and, by IH, $\langle\!| N^B |\!\rangle : \langle\!| \Theta |\!\rangle \vdash \langle\!| B |\!\rangle$ is derivable in $\lambda\mu^{\mathbf{1}}$. The result is obtained by Weakening and $\supset$I.
- Case T-∀E: $M = \mathsf{ins}_i^E(N^{\forall i.B})^{B\{i \leftarrow E\}}$, $A = B\{i \leftarrow E\}$, $\langle\!| M |\!\rangle = \langle\!| N^{\forall i.B} |\!\rangle \mathtt{unit}$ and, by Lemma 5.3.24, $\langle\!| B\{i \leftarrow E\} |\!\rangle = \langle\!| B |\!\rangle$. We can construct the following derivation:

$$\dfrac{\dfrac{}{\langle\!| N^{\forall i.B} |\!\rangle : \langle\!| \Theta |\!\rangle \vdash \mathbf{1} \supset \langle\!| B |\!\rangle}\,\mathrm{IH} \qquad \dfrac{}{\mathtt{unit} : \langle\!| \Theta' |\!\rangle, x_v^{\mathbf{1} \supset \langle\!| A |\!\rangle} \cup \langle\!| \Gamma |\!\rangle \vdash \mathbf{1}, \langle\!| \Delta |\!\rangle}\,\mathsf{Unit}}{\langle\!| N^{\forall i.B} |\!\rangle \mathtt{unit} : \langle\!| \Theta |\!\rangle \cup \langle\!| \Gamma |\!\rangle \vdash \langle\!| B |\!\rangle, \langle\!| \Delta |\!\rangle}\,\supset\mathsf{E}$$

□

Lemma 5.3.26 For every HLP-term M:

1. if $w^C_\Xi \notin \mathsf{FVV}(M)$, then $x_w^{\mathbf{1} \supset \langle\!| C |\!\rangle} \notin \mathsf{FVT}(\langle\!| M |\!\rangle)$;

2. if $\beta^C \notin \mathsf{FVF}(M)$, then $\beta^{\langle\!\langle C\rangle\!\rangle} \notin \mathsf{FVF}(\langle\!\langle M\rangle\!\rangle)$.

Proof.- The proof is analogous to that of Lemma 4.5.6 for HLP. For terms of the form $\mathsf{gen}_i(N^A)$ and $\mathsf{ins}_i^E(N^{\forall i.A})$, the result holds by IH. □

Lemma 5.3.27 For all λ^{FOLP}-terms M, N, for every truth variable x^A:
$\langle\!\langle M\rangle\!\rangle\{x^{\langle\!\langle A\rangle\!\rangle} \leftarrow \langle\!\langle N\rangle\!\rangle\} = \langle\!\langle M\{x^A \leftarrow N\}\rangle\!\rangle$.

Proof.- The proof is analogous to that of Lemma 4.5.7 for HLP. For terms of the form $\mathsf{gen}_i(N^A)$ and $\mathsf{ins}_i^E(N^{\forall i.A})$, the result holds by IH. □

Lemma 5.3.28 For all λ^{FOLP}-terms M, N, for every validity variable v_Ξ^A, proof witness r and truth variable $y^{\mathbf{1}} \notin \mathsf{FVT}(\langle\!\langle N\rangle\!\rangle)$:
$\langle\!\langle M\rangle\!\rangle\{x_v^{\mathbf{1}\supset\langle\!\langle A\rangle\!\rangle} \leftarrow \lambda y^{\mathbf{1}}.\langle\!\langle N\rangle\!\rangle\} \twoheadrightarrow_{R_1} \langle\!\langle M\{v_\Xi^A \leftarrow N,\mathsf{w}(N)\}\rangle\!\rangle$, where $\twoheadrightarrow_{R_1}$ is the reflexive transitive closure of β-reduction (rule R_1 in $\lambda\mu^{\mathbf{1}}$).

Proof.- The proof is analogous to that of Lemma 4.5.8 for HLP. For terms of the form $\mathsf{gen}_i(N^A)$ and $\mathsf{ins}_i^E(N^{\forall i.A})$, the result holds by IH. □

Lemma 5.3.29 For every λ^{FOLP}-term M, for all falsehood variables α^A, β^A:
$\langle\!\langle M\rangle\!\rangle\{\alpha^{\langle\!\langle A\rangle\!\rangle} \leftarrow \beta^{\langle\!\langle A\rangle\!\rangle}\} = \langle\!\langle M\{\alpha^A \leftarrow \beta^A\}\rangle\!\rangle$.

Proof.- By structural induction on M. All cases hold either immediately or by IH. □

Lemma 5.3.30 For all λ^{FOLP}-terms M, N^A, for all falsehood variables $\alpha^{A\supset B}$, β^B:
$\langle\!\langle M\rangle\!\rangle(\!(\,[\alpha^{\langle\!\langle A\supset B\rangle\!\rangle}](\bullet) \leftarrow [\beta^{\langle\!\langle B\rangle\!\rangle}](\bullet)\langle\!\langle N^A\rangle\!\rangle\,)\!) = \langle\!\langle M(\!(\,[\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)N^A\,)\!)\rangle\!\rangle$.

Proof.- The proof is analogous to that of Lemma 4.5.10 for HLP. For terms of the form $\mathsf{gen}_i(N^A)$ and $\mathsf{ins}_i^E(N^{\forall i.A})$, the result holds by IH. □

Lemma 5.3.31 For every λ^{FOLP}-term M, for every individual variable i and expression E:
$\langle\!\langle M\{i \leftarrow E\}\rangle\!\rangle = \langle\!\langle M\rangle\!\rangle$.

Proof.- By structural induction on M.

- If $M = x^A$: $M\{i \leftarrow E\} = x^{A\{i \leftarrow E\}}$.

$$\begin{array}{ll} \langle\!\langle M\{i \leftarrow E\}\rangle\!\rangle & = \\ x^{\langle\!\langle A\{i \leftarrow E\}\rangle\!\rangle} & =_{\text{L. 5.3.24}} \\ x^{\langle\!\langle A\rangle\!\rangle} & = \\ \langle\!\langle M\rangle\!\rangle. & \end{array}$$

- If $M = v_\Xi^A$: if $i \notin \Xi$, then $M\{i \leftarrow E\} = M$ and the result is immediate.
  Otherwise, $M\{i \leftarrow E\} = v_{\Xi\setminus\{i\}\cup\mathsf{FIV}(E)}^{A\{i \leftarrow E\}}$, and

$$
\begin{array}{ll}
\langle\!| M\{i \leftarrow E\} |\!\rangle & = \\
(x_v^{\mathbf{1} \supset \langle\!| A\{\cancel{i} \leftarrow E\} |\!\rangle})\texttt{unit} & =_{\text{L. 5.3.24}} \\
(x_v^{\mathbf{1} \supset \langle\!| A |\!\rangle})\texttt{unit} & = \\
\langle\!| M |\!\rangle.
\end{array}
$$

- If $M = (\lambda x^A.N^B)^{A \supset B}$: $M\{i \leftarrow E\} = \lambda x^{A\{i \leftarrow E\}}.(N^B)\{i \leftarrow E\}$, and

$$
\begin{array}{ll}
\langle\!| M\{i \leftarrow E\} |\!\rangle & = \\
\lambda x^{\langle\!| A\{i \leftarrow E\} |\!\rangle}.\langle\!| (N^B)\{i \leftarrow E\} |\!\rangle & =_{\text{IH}} \\
\lambda x^{\langle\!| A\{i \leftarrow E\} |\!\rangle}.\langle\!| N^B |\!\rangle & =_{\text{L. 5.3.24}} \\
\lambda x^{\langle\!| A |\!\rangle}.\langle\!| N^B |\!\rangle & = \\
\langle\!| M |\!\rangle.
\end{array}
$$

- If $M = (M'^{A \supset B} N^A)^B$: $M\{i \leftarrow E\} = (M'^{A \supset B})\{i \leftarrow E\}(N^A)\{i \leftarrow E\}$, and

$$
\begin{array}{ll}
\langle\!| M\{i \leftarrow E\} |\!\rangle & = \\
\langle\!| (M'^{A \supset B})\{i \leftarrow E\} |\!\rangle \langle\!| (N^A)\{i \leftarrow E\} |\!\rangle & =_{\text{IH}} \\
\langle\!| M'^{A \supset B} |\!\rangle \langle\!| N^A |\!\rangle & = \\
\langle\!| M |\!\rangle.
\end{array}
$$

- If $M = (!N^A)^{[\![s]\!] \equiv A}$: $M\{i \leftarrow E\} = !N^A\{i \leftarrow E\}^{([\![s]\!] \equiv A)\{\cancel{i} \leftarrow E\}}$, and

$$
\begin{array}{ll}
\langle\!| M\{i \leftarrow E\} |\!\rangle & = \\
\lambda x^{\mathbf{1}}.\langle\!| (N^A)\{i \leftarrow E\} |\!\rangle & =_{\text{IH}} \\
\lambda x^{\mathbf{1}}.\langle\!| N^A |\!\rangle & = \\
\langle\!| M |\!\rangle.
\end{array}
$$

- If $M = (M'^B \langle v^A_{\Xi'} := r, N^{[\![r]\!] \equiv A} \rangle)^{B\{v^A_{\Xi'} \leftarrow r\}}$:
$M\{i \leftarrow E\} = (M'^B)\{i \leftarrow E\}\langle v^A_{\Xi'}\{i \leftarrow E\} := r\{i \leftarrow E\}, (N^{[\![r]\!] \equiv A})\{i \leftarrow E\}\rangle$, and since, by Lemma 5.3.24, $\langle\!| A |\!\rangle\{i \leftarrow E\} = \langle\!| A |\!\rangle$, then

$$
\begin{array}{ll}
\langle\!| M\{i \leftarrow E\} |\!\rangle & = \\
(\lambda x_v^{\mathbf{1} \supset \langle\!| A |\!\rangle}.\langle\!| (M'^B)\{i \leftarrow E\} |\!\rangle)\langle\!| (N^{[\![r]\!] \equiv A})\{i \leftarrow E\} |\!\rangle & =_{\text{IH}} \\
(\lambda x_v^{\mathbf{1} \supset \langle\!| A |\!\rangle}.\langle\!| M'^B |\!\rangle)\langle\!| N^{[\![r]\!] \equiv A} |\!\rangle & = \\
\langle\!| M |\!\rangle.
\end{array}
$$

- If $M = ([\alpha^A]N^A)^{\perp}$: $M\{i \leftarrow E\} = [\alpha^{A\{i \leftarrow E\}}](N^A)\{i \leftarrow E\}$, and

$$
\begin{array}{ll}
\langle\!| M\{i \leftarrow E\} |\!\rangle & = \\
[\alpha^{\langle\!| A\{\cancel{i} \leftarrow E\} |\!\rangle}](N^A)\{i \leftarrow E\} & =_{\text{L. 5.3.24}} \\
[\alpha^{\langle\!| A |\!\rangle}](N^A)\{i \leftarrow E\} & =_{\text{IH}} \\
[\alpha^{\langle\!| A |\!\rangle}]N^A & = \\
\langle\!| M |\!\rangle.
\end{array}
$$

- If $M = (\mu\alpha^A.N^{\perp})^A$: this case is analogous to the previous one.
- If $M = (N^A +_{\mathrm{L}} s)^A$: $M\{i \leftarrow E\} = ((N^A)\{i \leftarrow E\} +_{\mathrm{L}} s\{i \leftarrow E\})^{A\{\cancel{i} \leftarrow E\}}$, and

$$
\begin{array}{ll}
\llparenthesis M\{i \leftarrow E\}\rrparenthesis & = \\
\llparenthesis (N^A)\{i \leftarrow E\}\rrparenthesis & =_{\text{IH}} \\
\llparenthesis N^A\rrparenthesis & = \\
\llparenthesis M\rrparenthesis.
\end{array}
$$

- If $M = (s +_{\mathrm{R}} N^B)^B$: this case is analogous to the previous one.
- If $M = (\mathsf{gen}_j(N^A))^{\forall j.A}$: if $i = j$, then $M\{i \leftarrow E\} = M$ and the result is immediate. Otherwise, $M\{i \leftarrow E\} = \mathsf{gen}_j((N^A)\{i \leftarrow E\})$ and

$$
\begin{array}{ll}
\llparenthesis M\{i \leftarrow E\}\rrparenthesis & = \\
\lambda x_i^{\mathbf{1}}.\llparenthesis (N^A)\{i \leftarrow E\}\rrparenthesis & =_{\text{IH}} \\
\lambda x_i^{\mathbf{1}}.\llparenthesis N^A\rrparenthesis & = \\
\llparenthesis M\rrparenthesis.
\end{array}
$$

- If $M = (\mathsf{ins}_j^{E'}(N^{\forall j.A}))^{A\{j \leftarrow E'\}}$: if $i = j$, then $M\{i \leftarrow E\} = M$ and the result is immediate. Otherwise, $M\{i \leftarrow E\} = \mathsf{ins}_j^{E'\{i \leftarrow E\}}(N^{\forall j.A}\{i \leftarrow E\})$, and

$$
\begin{array}{ll}
\llparenthesis M\{i \leftarrow E\}\rrparenthesis & = \\
\llparenthesis (N^{\forall j.A})\{i \leftarrow E\}\rrparenthesis \mathtt{unit} & =_{\text{IH}} \\
\llparenthesis N^{\forall j.A}\rrparenthesis \mathtt{unit} & = \\
\llparenthesis M\rrparenthesis.
\end{array}
$$

□

Lemma 5.3.32 For all λ^{FOLP}-terms M and N, if $M \to N$ in λ^{FOLP} *without* the use of permutative rules, then $\llparenthesis M\rrparenthesis \to^+ \llparenthesis N\rrparenthesis$ in $\lambda\mu^{\mathbf{1}}$. That is, $\llparenthesis M\rrparenthesis$ reduces to $\llparenthesis N\rrparenthesis$ in 1 or more steps.

Proof.- By induction on the derivation of M. The base cases are trivial, since no reductions may originate from x^A or v_{Ξ}^A. The inductive cases follow from the fact that reduction is closed under all constructors in $\lambda\mu^{\mathbf{1}}$. If the reduction takes place at the root of M, we must consider which reduction rule was used.

- $\beta : M = (\lambda x^A.M_1^B)M_2^A$ and $N = M_1^B\{x^A \leftarrow M_2^A\}$.

$$
\begin{array}{ll}
\llparenthesis M\rrparenthesis & = \\
(\lambda x^{\llparenthesis A\rrparenthesis}.\llparenthesis M_1^B\rrparenthesis)\llparenthesis M_2^A\rrparenthesis & \to_{R_1} \\
\llparenthesis M_1^B\rrparenthesis\{x^{\llparenthesis A\rrparenthesis} \leftarrow \llparenthesis M_2^A\rrparenthesis\} & =_{\text{L. 5.3.27}} \\
\llparenthesis M_1^B\{x^A \leftarrow M_2^A\}\rrparenthesis & = \\
\llparenthesis N\rrparenthesis.
\end{array}
$$

- $\gamma : M = (M_1^B\langle v_{\Xi'}^A := r, (!M_2^A)^{[\![r]\!]_{\Xi A}}\rangle)^{B\{v_{\Xi'}^A \leftarrow r\}}$ and $N = (M_1^B\{v_{\Xi'}^A \leftarrow M_2^A, r\})^{B\{v^A \leftarrow r\}}$.

$$
\begin{array}{ll}
\llparenthesis M\rrparenthesis & = \\
(\lambda x_v^{\mathbf{1}\supset\llparenthesis A\rrparenthesis}.\llparenthesis M_1^B\rrparenthesis)\lambda y^{\mathbf{1}}.\llparenthesis M_2^A\rrparenthesis & \to_{R_1} \\
\llparenthesis M_1^B\rrparenthesis\{x_v^{\mathbf{1}\supset\llparenthesis A\rrparenthesis} \leftarrow \lambda y^{\mathbf{1}}.\llparenthesis M_2^A\rrparenthesis\} & \xrightarrow{\text{L. 5.3.28}}_{R_1} \\
\llparenthesis M_1^B\{v_{\Xi'}^A \leftarrow M_2^A, r\}\rrparenthesis & = \\
\llparenthesis N\rrparenthesis.
\end{array}
$$

- $\mu : M = [\beta^A]\mu\alpha^A.M_1^\perp$ and $N = M_1^\perp\{\alpha^A \leftarrow \beta^A\}$.

$$\begin{array}{ll} \langle\!| M |\!\rangle & = \\ [\beta^{\langle\!| A |\!\rangle}]\mu\alpha^{\langle\!| A |\!\rangle}.\langle\!| M_1^\perp |\!\rangle & \to_{S_1} \\ \langle\!| M_1^\perp |\!\rangle\{\alpha^{\langle\!| A |\!\rangle} \leftarrow \beta^{\langle\!| A |\!\rangle}\} & =_{\text{L. 5.3.29}} \\ \langle\!| M_1^\perp\{\alpha^A \leftarrow \beta^A\} |\!\rangle & = \\ \langle\!| N |\!\rangle. & \end{array}$$

- $\zeta : M = (\mu\alpha^{A\supset B}.M_1^\perp)^{A\supset B}M_2^A$ and $N = \mu\beta^B.M_1^\perp(\!|\, [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)M_2^A \,|\!)$.

$$\begin{array}{ll} \langle\!| M |\!\rangle & = \\ (\mu\alpha^{\langle\!| A\supset B |\!\rangle}.\langle\!| M_1^\perp |\!\rangle)\langle\!| M_2^A |\!\rangle \to_{R_2} & \\ \mu\beta^{\langle\!| B |\!\rangle}.\langle\!| M_1^\perp |\!\rangle(\!|\, [\alpha^{\langle\!| A\supset B |\!\rangle}](\bullet) \leftarrow [\beta^{\langle\!| B |\!\rangle}](\bullet)\langle\!| M_2^A |\!\rangle \,|\!) & =_{\text{L. 5.3.30}} \\ \mu\beta^{\langle\!| B |\!\rangle}.\langle\!| M_1^\perp(\!|\, [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)M_2^A \,|\!)|\!\rangle & = \\ \langle\!| N |\!\rangle. & \end{array}$$

- $\theta : M = \mu\alpha^A.[\alpha^A]M_1^A$, $N = M_1^A$ and $\alpha^A \notin \mathsf{FVF}(M^A)$.
  Note that, by lemma 5.3.26, $\alpha^{\langle\!| A |\!\rangle} \notin \mathsf{FVF}(\langle\!| M^A |\!\rangle)$.

$$\begin{array}{ll} \langle\!| M |\!\rangle & = \\ \mu\alpha^{\langle\!| A |\!\rangle}.[\alpha^{\langle\!| A |\!\rangle}]\langle\!| M_1^A |\!\rangle & \to_{S_2} \\ \langle\!| M_1^A |\!\rangle & = \\ \langle\!| N |\!\rangle. & \end{array}$$

- $\xi : M = \mathsf{ins}_i^E(\mathsf{gen}_i(M_1^A))$, $N = (M_1^A)\{i \leftarrow E\}$.

$$\begin{array}{ll} \langle\!| M |\!\rangle & = \\ (\lambda x_i^{\mathbf{1}}.\langle\!| M_1^A |\!\rangle)\mathtt{unit} & \to_{R_1} \\ \langle\!| M_1^A |\!\rangle & =_{\text{L. 5.3.31}} \\ \langle\!| N |\!\rangle. & \end{array}$$

□

Lemma 5.3.33 For all λ^{FOLP}-terms M and N, if $M \to N$ in λ^{FOLP} using *only* permutative rules, then $\langle\!| M |\!\rangle = \langle\!| N |\!\rangle$.

Proof.- By induction on M. As in the previous lemma, no reductions originate from variables. The inductive cases are immediate. We now focus on the permutative reductions at the root of M.

- $\psi_L : M = (M_1^{A\supset B} +_{\mathrm{L}} s)^{A\supset B}M_2^A$ and $N = (M_1^{A\supset B}M_2^A)^B +_{\mathrm{L}} s$.
  $\langle\!| M |\!\rangle = \langle\!| M_1^{A\supset B} |\!\rangle\langle\!| M_2^A |\!\rangle = \langle\!| N |\!\rangle$.
- ψ_R : this case is analogous to the previous one.
- $\phi_L : M = M_1^B\langle v_{\Xi'}^A := r, (M_2^{[\![r]\!]_{\Xi A}} +_{\mathrm{L}} s)\rangle$, $N = (M_1^B\langle v_{\Xi'}^A := r, M_2^{[\![r]\!]_{\Xi A}}\rangle)^{B\{v_{\Xi'}^A \leftarrow r^A\}} +_{\mathrm{L}} s$.
  $\langle\!| M |\!\rangle = (\lambda x_v^{\mathbf{1}\supset\langle\!| A |\!\rangle}.\langle\!| M_1^B |\!\rangle)\langle\!| M_2^{[\![r]\!]_{\Xi A}} |\!\rangle = \langle\!| N |\!\rangle$.
- ϕ_R : this case is analogous to the previous one.
- $\chi_L : M = [\beta^A](M_1^A +_{\mathrm{L}} s)^A$ and $N = ([\beta^A]M_1^A)^\perp +_{\mathrm{L}} s$. $\langle\!| M |\!\rangle = [\beta^{\langle\!| A |\!\rangle}]\langle\!| M_1^A |\!\rangle = \langle\!| N |\!\rangle$.

- χ_R : this case is analogous to the previous one.
- ι_L : $M = \mu\alpha^A.(M_1^{\perp} +_{\mathrm{L}} s)^{\perp}$ and $N = (\mu\alpha^A.M_1^{\perp})^A +_{\mathrm{L}} s$. $\langle\!| M |\!\rangle = \mu\alpha^{\langle\!| A |\!\rangle}.\langle\!| M_1^{\perp} |\!\rangle = \langle\!| N |\!\rangle$.
- ι_R : this case is analogous to the previous one.
- ϵ_L: $M = \mathsf{ins}_i^E(M_1^{\forall i.A} +_{\mathrm{L}} t)$, $N = \mathsf{ins}_i^E(M_1^{\forall i.A}) +_{\mathrm{L}} t$. $\langle\!| M |\!\rangle = \langle\!| M_1^{\forall i.A} |\!\rangle \mathtt{unit} = \langle\!| N |\!\rangle$.
- ϵ_R : this case is analogous to the previous one.

□

Lemma 5.3.34 Permutative reduction is SN.

Proof.- We will show this by means of a polynomial interpretation $(\cdot)_{\mathcal{A}}$ in $\mathbb{N}_{\geq 2}$, using the standard order for natural numbers. We extend the interpretation shown in Section 4.5, with two new cases (for $\mathsf{gen}_i(M^A)$ and $\mathsf{ins}_i^E(M^A)$ respectively). For all terms involving sets of individual variables (Ξ), the interpretation is analogous to that of their propositional counterpart, ignoring the Ξ.

Terms are interpreted as follows:

$$
\begin{aligned}
x_{\mathcal{A}}^B &\triangleq 2 \\
v_{\Xi\ \mathcal{A}}^B &\triangleq 2 \\
(M^{A \supset B} N^A)_{\mathcal{A}}^B &\triangleq M_{\mathcal{A}}^{A \supset B} \times N_{\mathcal{A}}^A \\
(\lambda x^A.M^B)_{\mathcal{A}}^{A \supset B} &\triangleq 2 \times M_{\mathcal{A}}^B \\
([\alpha^B]M^B)_{\mathcal{A}}^{\perp} &\triangleq 2 \times M_{\mathcal{A}}^B \\
(\mu\alpha^B.M^{\perp})_{\mathcal{A}}^B &\triangleq 2 \times M_{\mathcal{A}}^{\perp} \\
(!M^B)_{\mathcal{A}}^{[\![s]\!]_{\Xi}B} &\triangleq 1 + M_{\mathcal{A}}^B \\
(M^B\langle v_{\Xi'}^C := r, N^{[\![r]\!]_{\Xi}B}\rangle)_{\mathcal{A}}^{B\{v_{\Xi'}^C \leftarrow r\}} &\triangleq M_{\mathcal{A}}^B \times N_{\mathcal{A}}^{[\![r]\!]_{\Xi}B} + 1 \\
(M^B +_{\mathrm{L}} t)_{\mathcal{A}}^B &\triangleq 2 \times M_{\mathcal{A}}^B + 2 \\
(s +_{\mathrm{R}} M^B)_{\mathcal{A}}^B &\triangleq 2 \times M_{\mathcal{A}}^B + 2 \\
(\mathsf{gen}_i(M^A))_{\mathcal{A}}^{\forall i.A} &\triangleq 2 \times M_{\mathcal{A}}^A \\
(\mathsf{ins}_i^E(M^{\forall i.A}))_{\mathcal{A}}^{A\{i \leftarrow E\}} &\triangleq 2 \times M_{\mathcal{A}}^A
\end{aligned}
$$

We now show that the permutative rules are compatible with this interpretation:

- ψ_L : $(M^{A \supset B} +_{\mathrm{L}} t)^{A \supset B} N^A \to (M^{A \supset B} N^A)^B +_{\mathrm{L}} t$.

$$(2 \times M_{\mathcal{A}}^{A \supset B} + 2) \times N_{\mathcal{A}}^A > 2 \times M_{\mathcal{A}}^{A \supset B} \times N_{\mathcal{A}}^A + 2 \text{ since } N_{\mathcal{A}}^A > 1.$$

- ψ_R : $(s +_{\mathrm{R}} M^{A \supset B})^{A \supset B} N^A \to s +_{\mathrm{R}} (M^{A \supset B} N^A)^B$. Same as ψ_L.
- ϕ_L : $(M^B +_{\mathrm{L}} t)\langle v_{\Xi'}^A := r, N^{[\![r]\!]_{\Xi}A}\rangle \to (M^B\langle v_{\Xi'}^A := r, N^{[\![r]\!]_{\Xi}A}\rangle)^{B\{v_{\Xi'}^A \leftarrow r\}} +_{\mathrm{L}} t$.

$$M_{\mathcal{A}}^B \times (2 \times N_{\mathcal{A}}^{[\![r]\!]_{\Xi}A} + 2) + 1 > 2 \times (M_{\mathcal{A}}^B \times N_{\mathcal{A}}^{[\![r]\!]_{\Xi}A} + 1) + 2$$

  since $M_{\mathcal{A}}^B > 1.5$.
- ϕ_R : $(s +_{\mathrm{R}} M^B)\langle v_{\Xi'}^A := r, N^{[\![r]\!]_{\Xi}A}\rangle \to s +_{\mathrm{R}} (M^B\langle v_{\Xi'}^A := r, N^{[\![r]\!]_{\Xi}A}\rangle)^{B\{v_{\Xi'}^A \leftarrow r\}}$. Same as ϕ_L.

- $\chi_L : [\beta^A](M^A +_{\mathrm{L}} t)^A \to ([\beta^A]M^A)^\perp +_{\mathrm{L}} t$.

$$2 \times (2 \times M^A_{\mathcal{A}} + 2) > 2 \times 2 \times M^A_{\mathcal{A}} + 2$$

- $\chi_R : [\beta^B](s +_{\mathrm{R}} N^B)^B \to s +_{\mathrm{R}} ([\beta^B]N^B)^\perp$. Same as χ_L.
- $\iota_L : \mu\alpha^A.(M^\perp +_{\mathrm{L}} t)^\perp \to (\mu\alpha^A.M^\perp) +_{\mathrm{L}} N^B$, if $\alpha^A \notin \mathsf{FVF}(t)$.

$$2 \times (2 \times M^A_{\mathcal{A}} + 2) > 2 \times 2 \times M^A_{\mathcal{A}} + 2$$

- $\iota_R : \mu\alpha^A.(s +_{\mathrm{R}} N^\perp)^\perp \to M^B +_{\mathrm{R}} (\mu\alpha^A.N^\perp)$, if $\alpha^A \notin \mathsf{FVF}(s)$. Same as ι_L.
- $\epsilon_L : \mathsf{ins}^E_i(M^{\forall i.A} +_{\mathrm{L}} t) \to \mathsf{ins}^E_i(M^{\forall i.A}) +_{\mathrm{L}} t$.

$$2 \times (2 \times M^{\forall i.A}_{\mathcal{A}} + 2) > 2 \times 2 \times M^{\forall i.A}_{\mathcal{A}} + 2$$

Thus, permutative reduction is polynomially terminating, and therefore SN. □

Proposition 5.3.35 Every typable λ^{FOLP}-term is SN.

Proof.- We prove this result by contradiction, in the same way as for Proposition 4.5.14. Assume that there is an infinite reduction sequence starting from a typable λ^{FOLP}-term M_0. We will distinguish between principal reductions ($\overset{\mathrm{B}}{\to}$) and permutative reductions ($\overset{\mathrm{P}}{\to}$) within this sequence.

Since, by Lemma 5.3.34, permutative reduction is SN, our sequence must contain an infinite number of principal reduction steps. Between any two principal steps, there may be 0 or more permutative steps (always a finite number). Therefore, the reduction sequence has the form:

$$M_0 \overset{\mathrm{P}}{\twoheadrightarrow} M'_0 \overset{\mathrm{B}}{\to} M_1 \overset{\mathrm{P}}{\twoheadrightarrow} M'_1 \overset{\mathrm{B}}{\to} M_2 \overset{\mathrm{P}}{\twoheadrightarrow} M'_2 \overset{\mathrm{B}}{\to} \cdots$$

Additionally, by Lemma 5.3.33, $\langle\!| M_i |\!\rangle = \langle\!| M'_i |\!\rangle$ for every i. Also, by Lemma 5.3.32, we know that for every i, $\langle\!| M_i |\!\rangle \to^+ \langle\!| M_{i+1} |\!\rangle$ in $\lambda\mu^1$. We can therefore construct an infinite $\lambda\mu^1$-reduction sequence:

$$\langle\!| M_0 |\!\rangle \to^+ \langle\!| M_1 |\!\rangle \to^+ \langle\!| M_2 |\!\rangle \to^+ \cdots$$

However, M_0 is typable in λ^{FOLP} and, by Lemma 5.3.19, so is every M_i. Since the mapping preserves typability (Lemma 5.3.25), then we have an infinite reduction sequence of typable $\lambda\mu^1$-terms. This is an absurd, since reduction of typable $\lambda\mu^1$-terms is SN. Therefore, there cannot be an infinite reduction sequence starting from a typable λ^{FOLP}-term. □

Chapter 6

Conclusions and Future Work

This two part thesis has addressed two topics in rewriting.

In the first part we have presented a system of combinatory logic (CL_P) for a λ-calculus (λP) in which functions may be abstracted over a general notion of pattern that includes applications and abstractions themselves. We have proved that the associated notion of pattern-matching can be handled within the context of a Term Rewriting System (TRS), that is, by means of first-order rewriting. For that, two issues have been addressed:

1. Emulating successful matching in λP by means of combinators; and
2. Computing the resulting substitution, also by means of combinators which have to decompose applicative terms to achieve the desired result.

We have also worked out the necessary restrictions in order to avoid ill-formed patterns which may break confluence. This has been achieved by introducing syntactic characterizations of such restrictions. Moreover, given a term or pattern, these restrictions can be efficiently verified. We have also presented a type system based on simple types, for which we proved normalization of typable terms. Finally, a number of extensions have been addressed of which $CL_\star$ stands out. In contrast to CL_P, which although presented in terms of a finite number of rule schemas has an infinite number of rule instances, $CL_\star$ is a finite TRS. This is possible by encoding matching within the calculus itself, albeit at the cost of complicating the combinator syntax.

In the second part of this thesis we have developed a presentation of LP based on hypothetical reasoning, dubbed HLP. The work builds, on the one hand, on Parigot's Classical Natural Deduction and, on the other, on prior work on hypothetical presentations of an intuitionistic fragment of LP [AB07, BF12, BB10]. This yields a Natural Deduction formalism for proving LP theorems. A term assignment is proposed, which is accompanied by a fine analysis of normalisation of derivations in HLP: derivations are represented as terms and normalisation steps on derivations are encoded as reduction steps over terms. The resulting lambda calculus, the λ^{LP}-calculus, is shown to enjoy Type Preservation, Strong Normalization and Confluence. A notion of proof equivalence was introduced along the way in order to address known problems of failure

of Type Preservation where the operator responsible for proof internalization (and it's introduction rule $\Box\mathsf{I}$) causes proof witnesses to be modified upon reduction. It also was necessary to reformulate the inference rules related to the "+" operator – which introduces non-determinism into the proofs – in order to express all LP theorems, since theorems with free variables exist in LP: they are allowed as long as they are located on the side of the "+" which is not used in the derivation. Finally, we developed a first-order extension of HLP, named FOHLP, and proved a correspondence between FOHLP and Artemov and Yavorskaya's FOLP. We introduced a term assignment λ^{FOLP} including principal contractions and permutative conversions, and proved the fundamental properties of Confluence of reduction, Type Preservation and Strong Normalization of typable terms. Some difficulties were encountered in dealing with bound variables, particularly the individual variables bound by the first-order "$\Box$" operator. Indeed, the $\Box\mathsf{I}$ and $\Box\mathsf{E}$ rules proved to be rather elusive.

6.1 Avenues for Further Research

We mention possible avenues to explore in each of the two topics developed in the thesis.

6.1.1 Combinators for patterns

Combinators for dynamic patterns. The Pure Pattern Calculus or PPC [Jay06, JK05, JK09] is a pattern calculus in which patterns may be created during runtime. For example, consider the PPC-term M:

$$x \hookrightarrow_x (x \hookrightarrow \texttt{true} | y \hookrightarrow \texttt{false})$$

The symbol "$\hookrightarrow$" allows abstractions to be built, its first argument being the pattern and the second one the body. A set of variable subscripts, such as x in "$\hookrightarrow_x$", indicates which variables in the pattern are considered matching variables (x in this example) and which are considered free variables (these are to be replaced from the "outside"). The order among the patterns is important, as reduction will proceed according to the first match. The PPC-term $M(\texttt{Succ 0})(\texttt{Succ 0})$ reduces in two steps to `true`. Note that, after the first step, $(\texttt{Succ 0})$ becomes a new pattern producing:

$$\texttt{Succ 0} \hookrightarrow \texttt{true} | y \hookrightarrow \texttt{false}$$

Similarly, $M(\texttt{Succ 0})\texttt{0}$ reduces to `false`. It would be interesting to obtain a combinatory account of PPC.

Higher-order λP-unification via combinators. We are interested in studying unification with respect to CL_P-equality [Dou93]. The formulation of a higher-order unification algorithm turns out to be complicated (even when restricting to simply typed λ-calculus terms), due to the presence of bound variables. For this reason, [Dou93] emphasizes the importance of considering the conversion to combinatory logic as an intermediate step, thus eliminating bound variables. In this way the formulation of the algorithm is simplified considerably, according to [Dou93]. Note that the absence of bound variables is the only requirement for a calculus to serve this

purpose. Our system of combinators seems to be an adequate language due to the fact that all required properties extend the classical ones.

Detecting matching failure through type checking. It would be interesting to study other possible type systems of CL_P, particularly dependent type systems in which the patterns play a role in detecting matching failure. The rules would have to be chosen carefully to account for undecided matching. For instance, they should allow the correct typing of $S_P M N P^\sigma$, taking into account that the normal form of NP^σ cannot be known in advance.

6.1.2 Curry-Howard Isomorphism for LP

LP through Contextual Modal Type Theory. Inference schemes typically uphold the invariant that all free variables are declared in the hypotheses. The PlusL scheme:

$$\frac{\Theta; \Gamma; \Delta \vdash A \mid s}{\Theta; \Gamma; \Delta \vdash A \mid s + t}\ \mathsf{PlusL}$$

fails in this respect. Although we have argued that this is required in order to prove all LP-theorems in HLP, it seems reasonable to explore truth dependent modalities [NPP08] at the possible cost of capturing a subset of LP-theorems (for example, replacing axiom **A4** by $[\![s]\!]A \supset [\![t]\!]B \supset [\![s+t]\!]A$, and analogously for **A5**). As mentioned, the modality $\Box A$ is replaced by $[\Gamma]A$ which informally may be read as $\Box(\Gamma \supset A)$. That is, validity of A is dependent on the truth of the hypotheses in Γ. A sample of three inference schemes of the resulting *Contextual Modal Logic* [NPP08] are:

$$\frac{\Theta; \Gamma_1 \vdash A}{\Theta; \Gamma_2 \vdash [\Gamma_1]A}\ \Box\mathsf{I} \qquad \frac{\Theta; \Gamma_1 \vdash [\Gamma_2]A \quad \Theta, v :: A[\Gamma_2]; \Gamma_1 \vdash C}{\Theta; \Gamma_1 \vdash C}\ \Box\mathsf{E}^v$$

$$\frac{\Theta_1, v :: A[\Gamma_1], \Theta_2; \Gamma_2 \vdash \Gamma_1}{\Theta_1, v :: A[\Gamma_1], \Theta_2; \Gamma_2 \vdash A}\ \mathsf{mvar}$$

Rather than proving **A4** ($[\![t]\!]A \supset [\![s+t]\!]A$) one would prove a formula of the form:

$$[\![\Gamma, \Delta.s]\!]A \supset [\![\Gamma, \Delta.s+t]\!]A$$

Note that t would be allowed to have free variables in Γ, Δ.

Logical framework based on FOHLP. It would be interesting to develop a logical framework based on first-order HLP. We believe that the Beluga framework [Pie10] may provide relevant inspiration for this purpose, since Beluga itself is based on [NPP08], and it allows the use of multiple contexts as well as dependent types.

Quantification over proof variables. An HLP-based system with the ability to quantify over proof variables, in the spirit of [Fit05b], is worthy of study. This should include an analysis of the induced polymorphism over the term assignment.

Type inference techniques for realization. We think that a fresh look on realization of S4 in the setting of HLP could be an interesting avenue for exploration. It should be noted that this is a non-trivial problem in the presence of inference schemes which mix polarities such as $\supset$ E, hence the reason why the first such proof [Art95, Art01] relied on a cut-free sequent calculus presentation of LP. Indeed, all known (to the author of this thesis) realization proofs rely on presentations where related[1] occurrences of a $\Box$ do *not* occur both in positive and negative positions. We think it could be interesting to put the well-developed type-inference technology to work *but* to infer the decorations of boxes rather than to infer types. Relations with higher-order unification may appear along the way.

[1] See notion of "family" in [Art01].

Appendix A

Resumen en Castellano

A.1 Introducción

El Cálculo Lambda fue introducido por Alonzo Church a fines de la década de 1940 [Chu36, Chu40, Chu41] como parte de un esfuerzo por desarrollar un sistema de lógica formal. En 1936, Kleene [K+36], un estudiante de doctorado de Church, probó la equivalencia entre λ-definibilidad y la recursividad de Gödel-Herbrand, y Turing mostró su equivalencia con su propia noción de computabilidad [Tur37]. El Cálculo Lambda emergió entonces como un modelo de cómputo conciso basado en el concepto ya conocido de **función**. La abstracción de variables en una expresión da lugar a dichas funciones. Consideremos la expresión:

$$x + 2 * x$$

Al abstraer la indeterminada "x" se genera la expresión formal:

$$\lambda x.x + 2 * x \tag{A.1}$$

Esta expresión denota una función cuyo parámetro formal es "x" y cuyo cuerpo es "$x + 2 * x$". Una teoría de igualdad asociada está dada por la noción fundamental noción de reemplazar un parámetro formal por el argumento correspondiente:

$$(\lambda x.M)\, N =_\beta M\{x \leftarrow N\} \tag{A.2}$$

donde $M\{x \leftarrow N\}$ denota el reemplazo de cada ocurrencia libre de x en M por N. Por ejemplo:

$$(\lambda x.x + \underline{2} * x)\underline{3} =_\beta \underline{3} + \underline{2} * \underline{3} =_\beta (\lambda x.\underline{3} + x * \underline{3})\underline{2}$$

donde $\underline{2}$ y $\underline{3}$ son los numerales correspondientes a 2 y 3, respectivamente. El conjunto de expresiones sobre las cuales se formula esta teoría (1.2), las expresiones λ, está dado por la gramática:

$$M, N \quad ::= \quad x \mid M\, N \mid \lambda x.M$$

Al orientar (1.2) de izquierda a derecha, surge una noción de **paso de reducción** $\rightarrow_\beta$. La reducción de expresiones λ puede representarse como una secuencia de pasos de reducción. Por ejemplo, $(\lambda x.\lambda y.yx)S\underline{0}$ reduce en dos pasos a $S\underline{0}$:

$$(\lambda x.\lambda y.yx)\underline{0}S \to_\beta (\lambda y.y\underline{0})S \to_\beta S\underline{0}$$

La expresión $S\underline{0}$ ya no se puede **reducir**. En otras palabras, no hay un M tal que $S\underline{0} \to_\beta M$. Esta expresión se denomina una **forma normal**. Intuitivamente, las formas normales representan el resultado final de un cómputo. Este modelo de cómputo es la base del paradigma de programación funcional, en el cual el programador especifica un cierto número de declaraciones de la forma $f \doteq \lambda x_1.\ldots.\lambda x_n.M$ y luego presenta una expresión (que puede contener ocurrencias de funciones definidas por el usuario) para ser reducida a forma normal (o a alguna noción similar de resultado).

Un esfuerzo independiente por revisitar los fundamentos de la lógica formal, motivado, en particular, por el deseo de eliminar la noción de sustitución de variables en sistemas formales de lógica, fue la introducción de la **Lógica Combinatoria** (CL) por Schönfinkel [Sch24] en 1924. Una teoría deductiva fue incorporada a CL por Curry en 1930 [Cur30]. Curry, Feys, Hindley y Seldin han estudiado CL en profundidad [Cur34, Cur69, CF58, CSH72]. En CLexisten expresiones y una teoría asociada de equivalencia entre las mismas. Las expresiones en CL se construyen a partir de constantes predefinidas (**combinadores**), típicamente K, S e I, y un constructor binario Ap (aplicación):

$$M, N ::= x \mid K \mid S \mid I \mid Ap(M, N)$$

Para mejorar la legibilidad, Ap suele omitirse, de manera que la expresión $Ap(M, N)$ se escribe $M\ N$. La teoría de equivalencia entre expresiones de CL está dada por las siguientes ecuaciones (se asume que la aplicación asocia a izquierda):

$$\begin{array}{lcl} Sxyz & \doteq & xz(yz) \\ Kxy & \doteq & x \\ Ix & \doteq & x \end{array}$$

Por ejemplo, puede probarse en esta teoría que $SKIK$ es equivalente a K:

$$SKIK \doteq KK(IK) \doteq K$$

Poco después de su introducción, se descubrieron grandes semejanzas entre CL y el Cálculo Lambda. En efecto, Rosser [Ros35] probó la equivalencia entre ambos sistemas. Existen mapeos entre el Cálculo Lambda y CL, denotados como $\cdot_{CL}$ y $\cdot_\lambda$, los cuales gozan de las siguientes propiedades:

1. Para todo par de expresiones de CL M y N: si $M \to_{CL} N$, entonces $M_\lambda \twoheadrightarrow_W N_\lambda$.

2. Para todo par de expresiones λ M y N: si $M \to_W N$, entonces $M_{CL} \twoheadrightarrow_{CL} N_{CL}$.

3. Para toda expresión λ M: $(M_{CL})_\lambda \twoheadrightarrow_\beta M$.

Aquí $\twoheadrightarrow_W$ representa la clausura reflexo-transitiva de la relación de **reducción débil** [ÇH98], una reducción β restringida en la cual un redex solo puede contraerse si sus variables libres no están ligadas por un λ externo. $\to_\beta$ significa reducción β en el Cálculo Lambda, $\to_{CL}$ significa reducción en CL, y $\twoheadrightarrow_\beta$ y $\twoheadrightarrow_{CL}$ denotan las clausuras reflexo-transitivas de las respectivas

reducciones. Estas nociones y resultados se estudian en las Secciones 2.1 y 2.4. Se ha demostrado así que *CL* es Turing-completa, lo cual dio a *CL* una posición equivalente a la del Cálculo Lambda como teoría de la computabilidad.

Actualmente, el interés en la Lógica Combinatoria se ha reavivado gracias a sus aplicaciones en Ciencias de la Computación, particularmente en relación con la compilación de lenguajes de programación funcional. Estas aplicaciones datan de los primeros trabajos en los '50 [Cur52, Fit58], pasando por la *máquina SECD* de Landin, el lenguaje CUCH de Böhm y Gross [BG66], los combinadores categóricos de Curien [Cur86, Cur93, CCM87], y la *G-Machine* y sus variantes, utilizadas en la compilación de Haskell [Joh83, Aug84, Kie85, BPJR88, PJS89, HK84, PJHH+93].

A.1.1 Patrones en la Programación Funcional

Pattern matching es una herramienta fundamental de la programación funcional. Un ejemplo simple de programa que utiliza patrones es el programa `length` que computa la longitud de una lista (el código está escrito en Haskell).

```
length []     = 0
length (x:xs) = 1 + length xs
```

Otro ejemplo es el siguiente, que exhibe el uso de patrones anidados, siendo `Sueldo` un constructor de datos cuyos argumentos son un id de empleado y su salario:

```
actualizar []                       f = []
actualizar ((Sueldo id valor):xs) f = (Sueldo id (f valor)): actualizar xs f
```

En reconocimiento de su utilidad y en un esfuerzo por estudiar los patrones y sus propiedades en su forma más pura, han surgido diversos **cálculos de patrones** [CK01, Jay01, WH08, JK09, AMR06, Jay09, JGW11]. Como ejemplo representativo, describimos brevemente el pionero λP [vO90] recientemente revisitado en [KvOdV08] (ver Sección 3.1 para más detalles). En λP, la abstracción funcional tradicional $\lambda x.M$ es reemplazada por la más general $\lambda P.M$:

$$M, N, P ::= x \mid MN \mid \lambda P.M$$

El patrón P puede ser un término cualquiera. En particular, puede ser una variable, subsumiendo así a la abstracción funcional estándar del Cálculo Lambda. Ejemplos de términos de λP son $\lambda(\lambda x.y).y$, $\lambda z.\lambda(\lambda x.x).\lambda y.y$, $(\lambda(\lambda x.y).y)(\lambda w.z)$ y, por supuesto, todos los términos del Cálculo Lambda. El patrón especifica qué forma puede tener el argumento para que el *matching* tenga éxito. Una función solo puede aplicarse a un argumento que sea instancia de su patrón. Luego, la aplicación se ejecuta sustituyendo los términos ligados a las variables libres del patrón dentro del cuerpo de la función:

$$(\lambda P.M)P^\sigma \to_{\beta_P} M^\sigma$$

Aquí σ denota una sustitución de variables por términos, y P^σ el resultado de aplicarla a P. Notar que en el caso en que P es a variable, obtenemos $\to_\beta$. Un ejemplo de un paso de reducción en λP es:

$$(\lambda xy.x)((\lambda x.x)(zw)) \to_{\beta_P} \lambda x.x \tag{A.3}$$

Otro ejemplo es el paso de reducción a partir del término $(\lambda(\lambda x.y).y)(\lambda z.w)$. El patrón $\lambda x.y$ admite cualquier función constante como argumento, y el resultado de aplicar $\lambda(\lambda x.y).y$ a un argumento de la forma $\lambda z.M$ (con $z \notin \mathsf{FV}(M)$) será M (en este caso, $M = w$). Sin embargo, la aplicación $(\lambda(\lambda x.y).y)z$ no reduce, ya que el argumento z no coincide con el patrón $\lambda x.y$. Notar que $(\lambda(\lambda x.y).y)(\lambda z.z)$ tampoco reduce, ya que $\lambda z.z$ no es una instancia de $\lambda x.y$ (no se admite la captura de variables). Esta situación, donde la reducción queda bloqueada de forma *permanente* debido a la falta de correspondencia entre el patrón y su argumento, se denomina **falla de matching** (en inglés **matching failure**). Existen casos en los cuales el argumento no coincide aún con el patrón, pero puede reducir a un término que sí coincida. Por ejemplo, $(\lambda(\lambda x.y).y)\,((\lambda x.x)\,(\lambda z.w))$. En este caso, decimos que el *matching* aún **no está decidido**.

Un punto clave es establecer las condiciones bajo las cuales λP es confluente, dado que el cálculo irrestricto no lo es. Por ejemplo, la siguiente reducción parte de la misma expresión de λP que (A.3), pero termina en una forma normal diferente:

$$(\lambda xy.x)((\lambda x.x)(zw)) \to_{\beta_P} (\lambda xy.x)(zw) \to_{\beta_P} z$$

A.1.2 Combinadores para Cálculos de Patrones

Del mismo modo en que CL permite computar sin usar variables en el Cálculo Lambda, buscamos alcanzar un resultado similar para los *cálculos de patrones*. Esto significa que las variables y sustituciones de los cálculos de patrones pueden desaparecer durante la compilación, preservando el comportamiento de la reducción. Para esto fijamos λP como objeto de estudio y nos centramos en la tarea de formular una lógica combinatoria correspondiente. Este es el foco del Capítulo 3 de esta tesis.

Con el fin de motivar CL_P, el sistema de combinadores que introducimos en el Capítulo 3, recordamos la traducción estándar del Cálculo Lambda a CL:

$$\begin{array}{rcl} x_{CL} & \triangleq & x \\ (MN)_{CL} & \triangleq & M_{CL}N_{CL} \\ (\lambda x.M)_{CL} & \triangleq & [x].M_{CL} \end{array}$$

donde "$\triangleq$" denota igualdad por definición, y $[x].M$ se define recursivamente como:

$$\begin{array}{lcll} [x].x & \triangleq & SKK & \\ [x].M & \triangleq & KM, & \text{si } M = K, S, I \text{ o } M = y \neq x \\ [x].(MM') & \triangleq & S([x].M)([x].M') & \end{array}$$

Notar que las abstracciones se traducen a términos de la forma KM o SM_1M_2. Por lo tanto, la aplicación de una abstracción a un argumento se traduce a KMN o a SM_1M_2N, siendo N la traducción del argumento. En CL_P, las aplicaciones de una abstracción a un argumento en λP se traducen términos de la forma K_PMN or $S_PM_1M_2N$, donde N es la traducción del argumento. Las expresiones K_P y S_P son combinadores de CL_P. La gramática de expresiones de CL_P es:

$$M, N ::= x \mid K_M \mid S_M \mid \Pi^1_{MN} \mid \Pi^2_{MN} \mid MN$$

Los subíndices de los combinadores en CL_P son una parte integral de los mismos. Los combinadores K_P y S_P se comportan de manera similar a K y S, pero con una diferencia importante:

mientras K y S siempre forman redexes al ser aplicados a la cantidad necesaria de argumentos (2 y 3 respectivamente), K_P (resp. S_P) compara su segundo (resp. tercer) argumento con el patrón P, y solo forma un redex si el *match* tiene éxito. Es importante notar que "match" en este contexto significa *matching* en un marco combinatorio, es decir *matching de primer orden*, el cual es mucho más simple que en λP (más detalles hacia el final de esta sección). Por ejemplo, $K_{S_y} x S_y$ es un redex, pero $K_{S_y} x K_y$ no lo es. Notar que el comportamiento de los combinadores K y S de CL es el mismo que el de K_P y S_P, resp., cuando P es una variable. Por esta razón, solemos abreviar K_x, para cualquier variable x, como K (y de manera análoga para S).

Los combinadores de la forma K_P y S_P por sí solos no tienen el poder suficiente para modelar la reducción de λP, ya que carecen de la habilidad para descomponer aplicaciones. Dado que el patrón P de $\lambda P.M$ se traduce a una aplicación en CL_P (excepto en el caso en que P es una variable) necesitamos poder definir funciones que reciban argumentos de la forma MN y los desarmen, devolviendo términos como M, N y $S(KN)M$. Por ejemplo, para traducir $\lambda(\lambda x.y).y$, necesitamos un término capaz de acceder a la variable y de la traducción de $\lambda x.y$, que es Ky. Para esto, definimos los combinadores Π^1_{PQ} y Π^2_{PQ}, denominados **proyectores**. $\Pi^1_{PQ}(MN)$ reduce a M cuando M es instancia de P y N es instancia de Q. Análogamente, $\Pi^2_{PQ}(MN)$ reduce a N bajo las mismas condiciones.

A continuación veremos algunos ejemplos de la traducción presentada en el Capítulo 3. Los términos del términos Cálculo Lambda se traducen a términos de CL tal como en la traducción original. las abstracciones cuyos patrones no son variables se traducen a términos de CL_P donde el combinador principal (el de más a la izquierda) está decorado con un patrón. Por ejemplo, $\lambda(\lambda x.x).y$ se traduce a $K_{SKK}y$, y $\lambda(\lambda x.y).y$ se traduce a $S_{Ky}(K(SKK))\Pi^2_{Ky}$.

Sea el término de λP $(\lambda(\lambda x.x).y)(\lambda z.z)$, el cual compara $\lambda z.z$ con el patrón $\lambda x.x$ y, como el *match* es exitoso, reduce en un paso a y. Este término se traduce como $K_{SKK}y(SKK)$. El subíndice SKK en K_{SKK} es un patrón (es, de hecho, la traducción del patrón $\lambda x.x$) con el cual se comparará el segundo argumento de K_{SKK}, es decir SKK, la traducción de $\lambda z.z$. En este caso el *matching* tiene éxito, y luego el término traducido reduce a y, tal como el término original en λP.

Términos más complejos pueden requerir más pasos para reducir. Por ej., $(\lambda(\lambda x.y).y)(\lambda w.z)$, que en λP reduce a z en un paso, se traduce a $S_{Ky}(K(SKK))\Pi^2_{Ky}(Kz)$. Este último requiere varios pasos para llegar a una forma normal:

$$\begin{array}{rcl} S_{Ky}(K(SKK))\Pi^2_{Ky}(Kz) & \to & K(SKK)(Kz)(\Pi^2_{Ky}(Kz)) \\ & \to & SKK(\Pi^2_{Ky}(Kz)) \\ & \to & K(\Pi^2_{Ky}(Kz))(K(\Pi^2_{Ky}(Kz))) \\ & \to & \Pi^2_{Ky}(Kz) \to z \end{array}$$

Las *fallas de matching* también son preservadas por la traducción: $(\lambda(\lambda x.x).y)(\lambda w.z)$ se traduce a $K_{SKK}y(Kz)$, que también contiene una falla de *matching*, ya que Kz no es instancia del patrón SKK.

Si bien a simple vista puede parecer que el *pattern matching* requiere sustituciones para ser computado (después de todo, un término M coincide con un patrón P si y solo si existe una sustitución σ tal que $M = P^\sigma$), computar la sustitución no es necesario en este marco. Todo lo que debe hacer el algoritmo de *matching* es descomponer el patrón y el argumento y

verificar que, donde haya un combinador en el patrón, el mismo combinador esté presente (en la misma posición) en el argumento. Las variables en los patrones producen un *match* exitoso con cualquier argumento.

Otro motivo por el cual el *matching* es más fácil de computar en un sistema basado en combinadores es la ausencia de ligadores. Por ejemplo, en λP, el término $\lambda x.x$ no coincide con el patrón $\lambda x.y$, ya que no hay una sustitución que aplicada a $\lambda x.y$ retorne $\lambda x.x$ (la convención establecida en [vO90] no permite la captura de variables). Como consecuencia, un algoritmo de *matching* debería verificar explícitamente que una variable no ocurra libre en un término. Por otra parte, si traducimos $\lambda x.y$ y $\lambda x.x$ a nuestro sistema de combinadores, obtenemos Ky y SKK, y ahora la *falla de matching* es evidente por el hecho de que SK no es instancia de K.

A continuación mostramos, a vuelo de pájaro, como CL_P y sus variantes se relacionan con λP y sus variantes, y también con SF (ver discusión abajo). Todos los resultados mencionados se desarrollan en el Capítulo 3, la primera de las dos partes de esta tesis:

$$\begin{tikzcd}
\lambda P \arrow[d, leftrightarrow, "Cor.\ 3.3.15, Prop.\ 3.3.18"'] & & \lambda C \arrow[d, "Cor.\ 3.6.15"] \\
CL_P \arrow[rr, "(\subset)"] \arrow[dr, "Prop.\ 3.7.10"'] & & CL_P + \theta & SF \arrow[l, dotted, "Rem.\ 3.6.16"] \\
& CL_\star & &
\end{tikzcd}$$

λP puede traducirse a CL_P preservando la reducción (débil) (Corolario 3.3.15) y lo mismo sucede en el sentido inverso (Proposición 3.3.18). λC [KvOdV08] es una variante de λP en la cual los patrones son términos algebraicos, y en la cual se permite el *pattern matching* múltiple:

$$(\lambda P_1.M_1|\ldots|\lambda P_k.M_k)P_i^\sigma \to M_i^\sigma$$

Pueden incorporarse constructores a CL_P junto con un nuevo conjunto de reglas ($CL_P + \theta$) y el sistema resultante sirve como destino de una traducción que preserva la reducción de λC. El sentido inverso también debería valer (los resultados quedesarrollamos no requirieron que definiéramos la traducción inversa). Si bien CL_P es un sistema de reescritura de términos de primer orden compuesto por un conjunto finito de esquemas de reglas, un conjunto infinito de reglas resulta de instanciar estos esquemas. $CL_\star$ es una variante de CL_P que es en efecto un sistema de reescritura de términos finito, aunque más verboso. Finalmente, hacemos un breve comentario sobre SF [JGW11], una lógica combinatoria similar en espíritu a la nuestra (ver la Sección 3.8 para una comparación en profundidad, y la Sección 3.6.3 para una traducción de una restricción de SF a $CL_P + \theta$ con patrones adicionales y *pattern matching* generalizado). No es posible traducir $CL_P + \theta$ a $CL_\star$, debido a que el *matching* múltiple no puede ser expresado por un mecanismo de *matching* expícito en el marco de un TRS. Para más detalles, referimos al lector a la Sección 3.7.4.

A.1.3 El Isomorfismo de Curry-Howard y la Lógica de Pruebas

Una llamativa semejanza entre las proposiciones lógicas y los tipos del Cálculo Lambda tipado fue observada por H. Curry y W. Howard mientras intentaban desarrollar una noción adecuada de construcción para la interpretación de la matemática intuicionista. Curry [CF58] notó que existe una fuerte conexión entre los tipos de los axiomas del fragmento implicativo de la Lógica Intuitionista proposicional y los combinadores de *CL*. Por ejemplo, el tipo del combinador K corresponde al axioma $A \supset B \supset A$. Tait [Tai65] descubrió una correspondencia cercana entre la eliminación de cortes y la reducción de términos del Cálculo Lambda. En 1969, Curry y Howard completaron la analogía al mostrar la correspondenc'a entre términos λ y derivaciones intuicionistas. las notas originales fueron resumidas por Howard [How95].

Desde la perspectiva de esta correspondencia, una proposición lógica $A \supset B$ puede verse como la expresión de tipo funcional $A \to B$ y vice versa. De manera similar, consideremos la siguiente prueba en Deducción Natural de $A \supset A$:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{A \vdash A}\,\mathsf{Ax}}{\vdash A \supset A}\,\supset\mathsf{I} \qquad \dfrac{\dfrac{}{C \vdash C}\,\mathsf{Ax}}{\vdash C \supset C}\,\supset\mathsf{I}}{\vdash (A \supset A) \wedge (C \supset C)}\,\wedge_{\mathsf{I}}}{\vdash A \supset A}\,\wedge_{\mathsf{E1}}$$

La misma puede codificarse como la expresión λ (tipada) $\pi_1 \langle (\lambda x^A.x), (\lambda y^C.y) \rangle$, donde $\langle \bullet, \bullet \rangle$ denota el constructor de pares y π_1 la expresión que proyecta la primera componente (ver Sección 2.5 para más detalles).

Esta semejanza fue el punto de partida para una analogía profunda entre la lógica y la computación, evidenciada claramente por la fuerte correspondencia entre sistemas de Deducción Natural y cálculos lambda tipados. La lógica y la computación resultan ser dos caras de la misma moneda. Conceptos como redexes, sustitución, reducción de términos, excepciones, continuaciones, consumo de recursos, computación en etapas, máquinas abstractas, estrategias de reducción, concurrencia, etc. tienen todas contrapartes lógicas. En esta tesis nos enfocamos en el caso particular de la lógica modal.

La *lógica modal* es un tipo de lógica simbólica que extiende a la lógica proposicional o de primer orden con la inclusión de operadores que expresan modalidad (*operadores modales*). Una modalidad representa una expresión (como 'necesariamente' o 'posiblemente') que se utiliza para calificar la verdad de un juicio. La lógica modal, en su furmulación original, es el estudio del comportamiento deductivo de las expresiones 'es necesario que' (representada por el operador "$\Box$") y 'es posible que' (representada por "$\Diamond$"). Sin embargo, el término *lógica modal* suele utilizarse de manera más amplia para referirse a una familia de sistemas. Esta familia incluye lógicas de conocimiento, de creencias, de conjugaciones y otras expresiones temporales, de expresiones deónticas (morales) como 'es obligatorio' y 'está permitido', y muchas otras. Entre la pltora de sistemas de lógica modal, destacamos la lógica $\mathsf{S4}$ de Lewis[1] dada su correspondencia cercana con la lógica que estudiamos en esta tesis. El lenguaje de $\mathsf{S4}$ es:

$$A, B \quad ::= \quad P \mid A \supset B \mid \Box A$$

[1]Esta presentación de $\mathsf{S4}$ fue acuñada por Gödel [Göd33].

donde P pertenece a un conjunto de variables proposicionales. $\mathsf{S4}$ consiste de los siguientes axiomas y reglas de inferencia:

A0.	Axiomas de la lógica proposicional clásica
T.	$\Box A \supset A$
K.	$\Box(A \supset B) \supset (\Box A \supset \Box B)$
4.	$\Box A \supset \Box\Box A$
MP.	$\vdash A \supset B$ y $\vdash A \Rightarrow \vdash B$
Nec.	$\vdash A \Rightarrow \vdash \Box A$

En uno de los numerosos intentos de interpretar la verdad intuitionista enn términos de demostrabilidad clásica (véase [Art01]), trabajos tempranos de Orlov [Orl25] y Gödel [Göd33] sugirieron prefijar cada subfórmula en Int (Lógica Intuicionista Proposicional) con "$\Box$", donde "$\Box$" está sujeto a las reglas fundamentales de $\mathsf{S4}$. Gödel estableció luego que dicha traducción, aplicada a fórmulas demostrables en Int, resulta en fórmulas demostrables en $\mathsf{S4}$. Más tarde, McKinsey y Tarski mostraron que este mapeo es *fiel* (i.e. la traducci'on de fórmulas no demostrables en Int no es demostrable en $\mathsf{S4}$). No obstante, para completar la explicación de la verdad intuitionista en términos de demostrabilidad clásica, es necesario relacionar la modalidad "$\Box$" con la demostrabilidad en la aritmética de Peano (PA):

$$\mathsf{Int} \hookrightarrow \mathsf{S4} \hookrightarrow \ldots ? \ldots \hookrightarrow \mathsf{PA} \tag{A.4}$$

Sin embargo, al leer "$\Box A$" como "$\exists x.\mathit{Proof}(x, \ulcorner A \urcorner)$", donde "$\ulcorner A \urcorner$" denota una codificación numérica adecuada de A, surge un problema ya que el teorema de $\mathsf{S4}$ $\Box(\neg\Box\ \bot)$, que expresa la consistencia de PA, es demostrable en PA. Esta situación fue observada por Gödel [Göd33], quien planteó dos problemas:

1. Develar la lógica modal del predicado de demostrabilidad formal $\exists x.\mathit{Proof}(x, \ulcorner A \urcorner)$.
2. Hallar la semántica de demostrabilidad exacta para $\mathsf{S4}$ (el "?" en (A.4)).

Ambos problemas han sido resueltos. En el primer caso tenemos el teorema de completitud de Solovay [Sol76] para la lógica de Löb (o GL). El segundo problema fue resuelto mediante la Lógica de Pruebas (LP) de Artemov.

A.1.4 La Lógica de Pruebas

La Lógica de Pruebas es una lógica modal que más tarde dio lugar a la familia de *Justification Logics* [Art08a, Art08b]. Fue introducida por Sergei Artemov en [Art95, Art01] en respuesta a la pregunta fundamental de cómo cerrar la brecha antes mencionada:

$$\mathsf{Int} \hookrightarrow \mathsf{S4} \overbrace{\hookrightarrow}^{a} \mathsf{LP} \overbrace{\hookrightarrow}^{b} \mathsf{PA} \tag{A.5}$$

LP surge esencialmente de skolemizar el cuantificador existencial implícito en el operador $\Box$. Esto es, reemplazar sentencias de la forma $\Box A$ (leído como *"existe una prueba de A"*) por:

$$[\![t]\!]A$$

que se lee como *"t es una prueba de A"*. La expresión t se denomina un **polinomio de prueba**, y pertenece al conjunto de expresiones especificadas por la siguiente gramática:

$$s, t \quad ::= \quad x \,|\, c \,|\, s \cdot t \,|\, !s \,|\, s + t$$

Los polinomios de prueba se construyen a partir de variables de prueba, constantes de prueba, aplicación, *bang* y suma. Los axiomas y reglas de inferencia de LP son los siguientes:

A0. Axiomas de la lógica proposicional clásica en el lenguage de LP
A1. $[\![s]\!]A \supset A$
A2. $[\![s]\!](A \supset B) \supset ([\![t]\!]A \supset [\![s \cdot t]\!]B)$
A3. $[\![s]\!]A \supset [\![!s]\!][\![s]\!]A$
A4. $[\![s]\!]A \supset [\![s + t]\!]A$
A5. $[\![t]\!]A \supset [\![s + t]\!]A$
MP. $\vdash A \supset B$ y $\vdash A \Rightarrow \vdash B$
Nec. A axioma **A0** – **A5** $\Rightarrow \vdash [\![c]\!]A$

Notar que al descartar los the polinomios de prueba que decoran estos axiomas, se obtienen los axiomas de S4 (**A4** y **A5** se colapsan en un teorema trivial). Volviendo a (A.5), la flecha marcada con (a) en (A.5) es el **teorema de realización** de Artemov:

Teorema 9.4 en [Art01] $\mathsf{S4} \vdash A \Rightarrow \mathsf{LP} \vdash A^r$, para alguna realización normal $\bullet^r$.

Un **realización** es una función que decora cada ocurrencia de $\Box$ con un polinomio de prueba; sew considera *normal* si cada ocurrencia negativa de $\Box$ se decora con una variable de prueba diferente. Esto implica que cada teorema de S4 tiene asociada una afirmación con respecto a las pruebas. Por ejemplo: $\Box A \supset \Box B$ puede realizarse como $[\![x]\!]A \supset [\![t(x)]\!]B$, para un polinomio de prueba $t(x)$ apropiado. La flecha marcada como (b) es el **teorema de corrección y completitud aritméticas** de Artemov. La correspondencia (A.5) fue luego extendida a un fragmento de LP que captura la demostrabilidad en aritmética de Heyting (HA) [AI07, Das11].

Otra propiedad destacada de LP es que está dotada de un mecanismo de reflexión.

Corolario 5.5 en [Art01] $\vdash A$ implica que existe un t cerrado t.q. $\vdash [\![t]\!]A$.

Aquí el polinomio de prueba t codifica una prueba de A. La prueba del Corolario 5.5 en [Art01] consiste de un análisis de la derivación dada de A en LP, y su codificación mediante polinomios de prueba.

A.1.5 Interpretación Computacional de la Lógica de Pruebas

La segunda parte de esta tesis es un esfuerzo por develar las metáforas computacionales que surgen de las capacidades de reflexión de LP. Si t se interpreta como una derivación de tipado de un término de tipo A, entonces un término de tipo $[\![t]\!]A$ debería incluir una codificación de t. Si vemos esta derivación de tipado como una derivación lógica, entonces todos los pasos de normalización que se le apliquen producirían nuevas derivaciones de tipado para A. Más aun, su relación con t debería explicitarse para que las derivaciones sean cerradas por normalización

(en la jerga de sistemas de tipos: para que valga la Preservación de Tipos). Esto sugiere que la interpretación computacional de LP debería ser un lenguaje de programación que incluya alguna manifestación explícita de las derivacions de tipos/certificados/trazas de cómputo.

El primer paso de esta búsqueda es diseñar una presentación en Deducción Natural con buen comportamiento. Siguiendo trabajos recientes sobre reconstrucciones de S4 intuicionista mediante juicios [DP96, DP01b, DP01a, ML84], introducimos juicios en los cuales se distingue entre las proposiciones cuya *verdad* se asume de aquellas cuya *validez* es asumida:

$$v_1 : A_1 \ \textit{válida}, \ldots, v_n : A_n \ \textit{válida}; a_1 : B_1 \ \textit{verdadera}, \ldots, a_m : B_m \ \textit{verdadera} \vdash A \ \textit{verdadera} | s$$

este juicio expresa que: "*s es evidencia de que A es verdadera, asumiendo que para cada $i \in 1..n$, v_i es evidencia de que A_i es válida y también que para cada $j \in 1..m$, a_j es evidencia de que B_j es verdadera*". Estos juicios se denominan *juicios hipotéticos* [ML84]. Se abrevian como:

$$\Theta; \Gamma \vdash A | s \tag{A.6}$$

donde Θ es un contexto de hipótesis de validez y Γ es un contexto de hipótesis de verdad; además los calificativos "válida" y "verdadera" se omiten. La evidencia s es una parte integral del juicio sin la cual la lectura propuesta no sería posible.

Interpretaciones de estos juicios son presentadas para un fragmento de LP basado en lógica minimal y sin los axiomas **A4** y **A5** en [AB07, BB10, BB14]. La siguiente regla de inferencia, pere a parecer natural, no brinda una explicación con buen comportamiento para $\Box$ (véase la discusión en la Sección 4.2.1):

$$\frac{\Theta; \cdot \vdash B \mid s}{\Theta; \Gamma \vdash [\![s]\!]B \mid !s} \Box\mathsf{I}'$$

En efecto, puede verse que resulta en un sistema donde falla la Preservación de Tipos. Op.cit. la reemplaza por la siguiente regla:

$$\frac{\Theta; \cdot \vdash B \mid s \quad \Theta; \Gamma \vdash s \equiv t : B}{\Theta; \Gamma \vdash [\![t]\!]B \mid !t} \Box\mathsf{I}$$

la cual apela a un nuevo juicio de equivalencia de testigos de prueba. Se han realizado algunos esfuerzos por reconocer las metáforas computacionales subyacentes a la presentación de Deducción Natural resultante a través de la interpretación de la normalización de pruebas como reducción:

- Un *Cálculo Lambda intensional* donde la información den *cómo* se computa un resultado (véase etiquetado de Lévy [Lév78]), en lugar de solamente *cuál* es el resultado [AB07].

- Un cálculo con *trazas de cómputo* [BB10, BB14], donde el juicio $\Theta; \Gamma \vdash s \equiv t : B$ codificado en la asignación de términos para $\Box\mathsf{I}$, es interpretado como una traza o historia del cómputo, con aplicaciones en el modelado de control de acceso basado en historia [AF03] y flujo de información basado en historia [BN05].

- Un cálculo de *unidades móviles certificadas* [BF12] que enriquece el código móvil con certifiacdos (que representan derivaciones de tipos). Dichas unidades toman la forma $\mathsf{box}_s\, M$, siendo s el certificado y M el ejecutable. La composición de unidades móviles certificadas permite construir código móvil a partir de otras piezas de código móvil *junto* con certificados, que están a su vez compuestos por otros certificados.

Lamentablemente, ninguno de los sistemas anteriores logra abordar la normalización de pruebas como cómputo para LP *en su totalidad*. Es decir, LP proposicional clásica, incluyendo todos sus axiomas. Este es el tema en el cual se centra el Capítulo 4. El plan de trabajo es entonces incorporar las nociones de cómputo que surgen de una interpretación al estilo Curry-Howard de la lógica proposicional clásica al análisis antes mencionado de razonamiento hipotético sobre LP. Una de estas nociones es la de la Deducción Natural Clásica de Parigot [Par92, Par97, Sau10] (ver reseña en la Sección 2.5.3). Esta tiene una conocida interpretación computacional y está formulada en el marco de (una variante de) la Deducción Natural, por lo cual resulta adecuada para nuestro propósito. Los juicios (A.6) se reemplazan entonces por:

$$\Theta; \Gamma; \Delta \vdash A \,|\, s$$

donde Δ es un contexto de fórmulas negadas. Una presentación en Deducción Natural para LP completa incluyendo estos juicios se desarrolla en el Capítulo 4, junto con pruebas de sus propiedades destacadas.

El Caso de Primer Orden

Dado un lenguaje de primer orden $\mathcal{L}$, el lenguaje de la Lógica de Pruebas de Primer Orden (FOLP) se obtiene al extender $\mathcal{L}$ con variables de prueba y símbolos de función para operaciones sobre pruebas. El conjunto de fórmulas se extiende con una versión skolemizada del operador modal modal $\Box$. Un aspecto crucial es cómo se entienden los parámetros en esta versión skolemizada. Consideremos la fórmula $\Box A(x)$, donde A tiene un parámetro x. Este parámetro puede jugar uno de dos roles en una *prueba* de $A(x)$. Puede interpretarse como un *parámetro global*. Los parámetros globales son "agujeros" que pueden ser sustituidos por cualquier término. Por ejemplo, en la siguiente derivación $\pi(y)$, donde R es un símbolo de predicado binario:

$$\dfrac{\dfrac{\forall x, y.R(x,y) \supset R(y,x)}{R(E,y) \supset R(y,E)} \qquad R(E,y)}{R(y,E)}$$

la variable y actúa como un parámetro global, ya que puede ser reemplazada por cualquier expresión de primer orden F para obtener una prueba $\pi(F)$ de $A(F)$. Por otro lado, los parámetros también pueden jugar un rol diferente, a saber el de *eigenvariables*: objetos sintácticos sujetos a generalización. Por ejemplo, consideremos la derivación:

$$\dfrac{\pi(y)}{\forall y.\forall x.R(x,y)}$$

donde la derivación $\pi(y)$ es:

$$\dfrac{\dfrac{\dfrac{\forall x.\forall y.R(x,y)}{\forall y.R(x,y)}}{R(x,y)}}{\forall x.R(x,y)}$$

El parámetro y en este caso no espera ser sustituido; en cambio actúa como una constant fresca. Estos dos roles han sido identificados en las Ciencias de la Computación en el contexto de asistentes de pruebas donde se explora el razonamiento sobre términos abiertos ([GJ02] y sus citas). Véase también la discusión sobre la demostración sobre expresiones cuantificadas universalmente utilizando un enfoque extensional versus intensional de Miller y Tiu [MT05].

Las consideraciones mencionadas conducen al siguiente operador modal skolemizado, propuesto en [AY11], que permite tener en cuenta ambas interpretaciones:

$$[\![s]\!]_{\Xi}A$$

Aquí Ξ es un conjunto variables que determina el rol que juega cada variable en una prueba de A. Las variables en Ξ juegan el rol de parámetros globales en A y por lo tanto en s (que codifica una prueba de A). Las variables que ocurren en A pero *no* están en Ξ se entienden como eigenvariables. las mismas se asumen entonces implícitamente ligadas en A: el conjunto de variables libres de $[\![s]\!]_{\Xi}A$ – denotado como $\mathsf{FIV}([\![s]\!]_{\Xi}A)$ – se define como Ξ. Dos resultados adicionales con respecto a los axiomas asociados a estenuevo operador son la realización de $\mathsf{S4}$ de primer orden ($\mathsf{FOS4}$) y reflexión (llamada "internalización" en [AY11]):

Teorema 2 en [AY11] $\mathsf{FOS4} \vdash A \Rightarrow \mathsf{FOLP} \vdash A^r$, para alguna realización normal $\bullet^r$.

Teorema 1 en [Art01] Sea $\Xi = \Xi_0 \cup \ldots \cup \Xi_n$. Luego $[\![x_0]\!]_{\Xi_0}A_0, \ldots, [\![x_n]\!]_{\Xi_n}A_n \vdash A \Rightarrow \exists t.\ [\![x_0]\!]_{\Xi_0}A_0, \ldots, [\![x_n]\!]_{\Xi_n}A_n \vdash [\![t(x_0, \ldots, x_n)]\!]_{\Xi}A$.

Cabe mencionar que FOLP no es recursivamente axiomatizable [AY01]. Esto no significa que no se pueda formular una presentación de FOLP con una semántica de demostrabilidad exacta, que a su vez sea capaz de realizar todos los teoremas de $\mathsf{S4}$ de primer orden. De hecho, esto ha sido logrado recientemente por Artemov e Yavorskaya [AY11] (ver Sección 5.1 para más detalles). Sin embargo, toda pretensión de completitud aritmétical debe abandonarse. No obstante, la completitud con respecta una semántica diferente, a saber la semántica de Kripke, ha sido establecida por Fitting [Fit14]. Podrían también considerarse cuantificadores sobre variables de prueba. Un sistema así fue estudiado en [Yav01] y mostró no ser axiomatizable. Otro trabajo relacionado es [WWdRZ02], donde el parámetro x en la fórmula $\Box A(x)$ se asume ligado (llamado "binding interpretation" en op.cit.). Este sistema posee una axiomatización completa, aunque no alcanza a realizar la lógica modal de primer orden.

El Capítulo 5 está dedicado a desarrollar una presentación en Deducción Natural de FOLP. Se demuestra la corrección y completitud con respecto al sistema de [AY11], se desarrolla un análisis de la normalización de pruebas y se demuestra también la Normalización Fuerte.

A.1.6 Estructura de la Tesis y Resumen de Contribuciones

En el Capítulo 2 definimos algunas nociones básicas que se usarán a lo largo de este trabajo. Tiene el objetivo general de fijar notación. El resto de la tesis, a excepción del Capítulo 6 (el capítulo final, que sugiere posibles direcciones para investigaciones futuras), está estructurado en dos partes que pueden ser leídas de manera independiente.

- **Parte I (Capítulo 3).** Presentamos CL_P, una Lógica Combinatoria enriquecida con capacidades de *pattern matching*. Este trabajo fue desarrollado bajo la supervisión de Ariel Arbiser; las referencias a "nosotros" en este capítulo se refieren a nosotros dos. Las contribuciones de esta parte pueden resumirse como sigue:

  - Una lógica combinatoria (TRS, sin variables ligadas) CL_P caz de simular la reducción de λP.
  - Condiciones bajo las cuales CL_P es confluente.
  - Un sistema de tipos para CL_P y pruebas de Preservación de Tipos y Normalización.
  - Extensiones de CL_P, donde el *matching* del meta-nivel de CL_P pasa a estar embebido en el sistema.

  Presentaciones: UNILOG 2010.

- **Parte II (Capítulos 4 and 5).** En el primero, presentamos HLP, un refinamiento de la Lógica de Pruebas de Artemov, basado en juicios y razonamiento hipotético. En el segundo, presentamos una extensión de primer orden de HLP llamada FOHLP, capaz de realizar la lógica modal S4 de primer orden. Este trabajo fue desarrollado bajo la supervisión de Eduardo Bonelli; las referencias a "nosotros" en este capítulo se refieren a nosotros dos. Las contribuciones de esta parte pueden resumirse como sigue:

  - Capítulo 4
    * Formulación de una presentación de Deducción Natural para LP proposicional clásica.
    * Una asignación de términos (Cálculo Lambda) para esta presentación.
    * Confluencia y terminación del proceso de normalización de pruebas (reducción de términos).

    Presentaciones: UNILOG 2013, IMLA 2013. Publicaciones: Proceedings of IMLA 2013 [SB13] y *Logica Universalis* [BS14].
  - Capítulo 5
    * Formulación de una presentación de Deducción Natural para LP clásica de primer orden.
    * Una asignación de términos (Cálculo Lambda) para esta presentación.
    * Confluencia y terminación del proceso de normalización de pruebas (reducción de términos).

    Presentaciones: SLALM 2014. Publicaciones: enviado al *Journal of Logic and Computation*.

$$\begin{tikzcd} & x \arrow[dl, two heads] \arrow[dr, two heads] & \\ y_1 \arrow[dr, dotted, two heads] & & y_2 \arrow[dl, dotted, two heads] \\ & z & \end{tikzcd} \tag{A.7}$$

Figura A.1: Confluencia

A.2 Reescritura, Lógica Combinatoria y Cálculo Lambda

Esta sección introduce nociones de reescritura usadas en esta tesis. A saber, las de *Sistemas Abstractos de Reescritura* (o ARS por su sigla en inglés) *Sistemas de Reescritura de Términos* (TRS), y algunos sistemas en particular como el Cálculo Lambda y la Lógica Combinatoria. Finalmente, revisitamos el isomorfismo de Curry-Howard, tanto en el caso intuicionista como en el clásico (con particular énfasis en la Deducción Natural Clásica de Parigot).

A.2.1 Sistemas Abstractos de Reescritura

Un **Sistema Abstracto de Reescritura**, o ARS, es un par $(\mathcal{A},\rightarrow)$ que consiste de un conjunto $\mathcal{A}$ y una relación binaria sobre $\mathcal{A}$ llamada **reducción**. Un ARS modela la transformación paso a paso de un objeto (por ejemplo un término de un lenguaje) en otro. Para $(x, y) \in \rightarrow$, escribimos simplemente $x \rightarrow y$ y decimos que x **reduce** a y, y también que y es el **reducto** de x. Al proceso de ir de x a y lo llamamos un **paso de reducción**. A los elementos de $\mathcal{A}$ los llamamos **objetos** o **términos**.

Un término x es una **forma normal** si no existe un y tal que $x \rightarrow y$. Decimos que y es la **forma normal de** x si y es una forma normal y $x \twoheadrightarrow y$, donde $\twoheadrightarrow$ denota la clausura reflexo-transitiva de $\rightarrow$. Definimos a continuación dos propiedades centrales de los ARSs:

Definición A.2.1 (Confluencia y Normalización) Una relación de reducción $\rightarrow$ se considera:

- **confluente** si para todo x, y_1, y_2: $(x \twoheadrightarrow y_1 \wedge x \twoheadrightarrow y_2) \supset \exists z(y_1 \twoheadrightarrow z \wedge y_2 \twoheadrightarrow z)$ (ver fig. A.1),
- **localmente confluente** si para todo x, y, z: $(x \rightarrow y_1 \wedge x \rightarrow y_2) \supset \exists z(y_1 \twoheadrightarrow z \wedge y_2 \twoheadrightarrow z)$,
- **fuertemente normalizante** (SN) si no existe una secuencia infinita $x_0 \rightarrow x_1 \rightarrow \ldots$,
- **débilmente normalizante** (WN) si todo término tiene una forma normal.

Lema A.2.2 (Lema de Newman [New42]) Un sistema de reescritura (abstracto) es confluente si es localmente confluente.

A.2.2 Sistemas de Reescritura de Términos

Un **Sistema de Reescritura de Términos** (TRS) es un ARS cuyos objetos son términos de primer orden, y cuya relación de reducción está presentda en un formato estándar de **reglas de reducción**.

Llamamos $\mathrm{Var}\, t$ al conjunto de variables presentes en un término t.

Una **sustitución** σ es una función de variables en términos tal que $\sigma(x) \neq x$ para una cantidad finita de variables x. Este conjunto finito de variables para x tales que $\sigma(x) \neq x$ se llama el **dominio** of σ.

Una sustitución σ puede ser **extendida** a una función $\widehat{\sigma}$ de términos en términos, la cual reemplaza todas las ocurrencias de variables en el término original por sus respectivas imágenes en σ. Por ejemplo, sean $t = f(0, x)$, $t' = f(y, f(x, y))$, y $\sigma = \{x \leftarrow g(y), y \leftarrow 0\}$. Entonces $\widehat{\sigma}(t) = f(0, g(y))$ y $\widehat{\sigma}(t') = f(0, f(g(y), 0))$. Por simplicidad, $\widehat{\sigma}$ también se denota como σ.

Para facilitar la lectura, podemos usar la notación t^σ para referirnos a $\sigma(t)$. Cuando una sustitución no tiene nombre, denotamos su aplicación a un a término escribiendo la sustitución a la derecha del término. Ej. $s\{x \leftarrow t\}$ significa s^σ donde $\sigma = \{x \leftarrow t\}$.

Se dice que hay un **match** entre de un término s contra un término t (o que s coincide con t) si existe una sustitución σ tal que $s = t^\sigma$, de manera que s y t^σ son sintácticamente idénticos. Cuando s coincide con un término t, llamamos a s una **instancia** de t.

Las reglas de reducción de un TRS se presentan de forma esquemática en el sentido de que 1) se admiten sustituciones arbitrarias de variables por términos; y 2) un paso de reducción así obtenido puede efectuarse en posiciones arbitrarias de un término más complejo.

Una **regla de reducción** es un par de términos $\langle l, r\rangle$ tales que l no es una variable y $\mathrm{Var}(r) \subseteq \mathrm{Var}(l)$. Se denota como $l \to r$. Una regla de reducción puede tener un nombre, ej. ρ, y entonces escribimos $\rho : l \to r$. Una **instancia** de ρ se obtiene aplicando una sustitución σ a ambos lados. El resultado es un paso de reducción $l^\sigma \to_\rho r^\sigma$. El lado izquierdo l^σ se denomina un **redex**, más precisamente un ρ-redex. El lado derecho r^σ se denomina su **reducto**.

Ejemplo A.2.1 Consideremos la siguiente regla de reducción:

$$x + 0 \to x$$

Esta regla nos permite realizar, entre otros, los siguientes pasos de reducción:

$$\begin{array}{lcl} x + 0 & \to & x \\ 1 + 0 & \to & 1 \\ (2 + 0) + y & \to & 2 + y \end{array}$$

La relación de reducción de un TRS $\mathcal{R}$, denotada como $\to$ (o $\to_{\mathcal{R}}$ para evitar ambigüedades), se define como la unión $\bigcup\{\to_\rho \mid \rho \in R\}$. De manera que $t \to_{\mathcal{R}} s$ si $t \to_\rho s$ para alguna regla de reducción $\rho \in R$.

Un término puede contener más de un redex, ofreciendo la opción de qué redex contraer primero. En algunos casos, esto hace que la reducción no sea confluente. Decimos que dos reglas $l_1 \to r_1$ y $l_2 \to r_2$ se **solapan** si existe un subtérmino l_1 que no es una variable y puede unificarse con l_2. Se asume que las reglas no comparten variables. En estos casos, el par de reductos en un paso de l_1^σ que surge de este solapamiento se llama **par crítico**.

Un par crítico $\langle s, t\rangle$ puede **cerrarse** si s y t tienen un reducto en común, el cual puede cerrarse en 0 o más pasos de reducción.

Lema A.2.3 (Lema del Par Crítico) Un TRS es localmente confluente si y solo si todos sus pares críticos pueden cerrarse.

Un TRS es **ortogonal** si es lineal a izquierda (los lados izquierdos de sus reglas de reducción no tienen variables repetidas) y no posee pares críticos. Está demostrado [BKdV03] que todo TRS ortogonal es confluente.

A.2.3 Cálculo Lambda

Recordemos de la introducción la sintaxis del Cálculo Lambda puro:

$$M, N ::= x \mid MN \mid \lambda x.M$$

donde x pertenece a un conjunto infinito numerable de variables $\mathcal{X}$.

En la expresión $\lambda x.M$, $\lambda x.$ es un **ligador**, que liga las ocurrencias libres de x en M.

Definición A.2.4 El conjunto de variables libres de un término M (notación: $\mathsf{FV}(M)$) se define como:

$$\begin{array}{lcl} \mathsf{FV}(X) & \triangleq & \{x\} \\ \mathsf{FV}(MN) & \triangleq & \mathsf{FV}(M) \cup \mathsf{FV}(N) \\ \mathsf{FV}(\lambda x.M) & \triangleq & \mathsf{FV}(M) \backslash \{x\} \end{array}$$

Los términos que solo difieren en los nombres de sus variables ligadas se consideran iguales, o **α-equivalentes**. De este modo $\lambda x.x$ y $\lambda y.y$ son la misma **función identidad**.

Al escribir términos, utilizamos paréntesis para eliminar ambigüedades. Adoptamos además las convenciones de que la aplicación asocia a izquierda y el alcance de un ligador se extiende hacia la derecha tanto como sea posible. Por ejemplo fgh significa $(fg)h$ y $\lambda x.\lambda y.Ma$ significa $\lambda x.(\lambda y.(Ma))$.

El Cálculo Lambda en su forma más simple tiene una única regla de reducción, que describe cómo sustituir el parámetro de una función por su argumento:

$$\beta \ : \ (\lambda x.M)N \to M\{x \leftarrow N\}$$

donde $M\{x \leftarrow N\}$ significa "sustituir todas las ocurrencias libres de x en M por N". La reducción del Cálculo Lambda es confluente, aunque no es (ni siquiera débilmente) normalizante.

El **Cálculo Lambda simplemente tipado** ($\lambda^{\to}$) introduce **tipos** en la sintaxis del lenguaje. La gramática de los términos pasa a ser

$$M, N ::= x \mid MN \mid \lambda x^A.M$$

con A un tipo.

$$\begin{array}{ll} Sxyz & \to xz(yz) \\ Kxy & \to x \\ Ix & \to x \end{array}$$

Un **contexto de tipado** es un conjunto finito $\{x_1^{A_1}, \ldots, x_n^{A_n}\}$ de variables decoradas con tipos, con $x_i \neq x_j$ para $i \neq j$.

Un **juicio de tipado** es una expresión de la forma $\Gamma \vdash M : A$, con Γ un contexto de tipado, M un término y A un tipo.

Las **reglas de tipado** de $\lambda^{\to}$, que definen la **relación de tipabilidad** $\Vdash$, son las siguientes:

$$\frac{}{\Gamma, x^A \vdash x : A} \qquad \frac{\Gamma, x^A \vdash M : B}{\Gamma \vdash \lambda x^A.M : A \to B} \qquad \frac{\Gamma \vdash M : A \to B \quad \Gamma \vdash N : A}{\Gamma \vdash MN : B}$$

Decimos que un juicio de tipado $\Gamma \vdash M : A$ es **derivable** si $\Gamma \vdash M : A \in \Vdash$. Si $\Gamma \vdash M : A$ es derivable, decimos que M tiene tipo A en Γ. Decimos que M es **tipable** si existen Γ y A tales que $\Gamma \vdash M : A \in \Vdash$.

La normalización fuerte para términos tipables fue probada independientemente por Dragalin [Dra68], Gandy [Gan80], Hinata [Hin67], Hanatani [HM66] y Tait [Tai67].

A.2.4 Lógica Combinatoria

Recordemos de la introducción que las expresiones de CL se construyen a partir de constantes predefinidas (**combinadores**) y un constructor binario Ap (aplicación):

$$M, N ::= x \mid K \mid S \mid I \mid Ap(M, N)$$

Las constantes I, K, S se llaman **combinadores**. La aplicación asocia a izquierda. Las reglas de reducción pueden verse en la tabla A.2.4, donde, como se mencionó, $Ap(M, N)$ se abrevia como $M\,N$.

El combinador I actúa como una identidad "universal" (sin restricciones de dominio). La aplicación está definida para todo par de términos. El combinador K puede verse como un constructor de funciones constantes. Al aplicar K a un argumento M obtenemos la función constante $K\ M$, que devuelve el valor M para cualquier argumento N. El combinador S es algo más complejo; está relacionado con la sustitución, y es relevante para lograr la completitud combinatoria (Teorema A.2.5).

CL es confluente, ya que es un TRS ortgonal. Sin embargo, CL no es fuertemente normalizante, como puede verse en el siguiente ciclo de reducción:

Ejemplo A.2.2 $SII(SII) \to I(SII)(I(SII)) \to SII(I(SII)) \to SII(SII)$

El combinador I puede omitirse para obtener una fórmulación más simple de CL, ya que I puede definirse en términos de S y K como SKK.

Relación entre *CL* y el Cálculo Lambda. Existe una correspondencia cercana entre el Cálculo Lambda y la Lógica Combinatoria. A partir de ahora, llamaremos Ter(CL) al conjunto de términos de CL, $\to_{CL}$ a la reducción en un paso generada por las reglas de reducción dadas por la Tabla A.2.4 y $\twoheadrightarrow_{CL}$ a su clausura reflexo-transitiva.

Definición A.2.5 (De combinadores a términos λ) Sea $M \in \mathrm{Ter}(CL)$. Su término λ correspondiente, denotado t_λ, se define como:

$$\begin{array}{lcl} x_\lambda & \triangleq & x \\ K_\lambda & \triangleq & \lambda x.\lambda y.x \\ S_\lambda & \triangleq & \lambda x.\lambda y.\lambda z.xz(yz) \\ I_\lambda & \triangleq & \lambda x.x \\ (tt')_\lambda & \triangleq & t_\lambda t'_\lambda \end{array}$$

Proposición A.2.3 Para todo $M, N \in \mathrm{Ter}(CL)$, si $M \twoheadrightarrow_{CL} N$ entonces $M_\lambda \twoheadrightarrow_\beta N_\lambda$.

La inversa, sin embargo, es falsa. En particular, KI es una forma normal en CL, mientras que $(KI)_\lambda = (\lambda x.\lambda y.x)(\lambda z.z)$ contiene un β-redex, y reduce en un paso a $\lambda y.(\lambda z.z)$. Por este motivo, a la reducción en CL se la llama **reducción débil**.

Definición A.2.6 (De términos λ a combinadores) La traducción de términos λ a combinadores, como se vio en la introducción, se define como:

$$\begin{array}{rcl} x_{CL} & \triangleq & x \\ (MN)_{CL} & \triangleq & M_{CL}N_{CL} \\ (\lambda x.M)_{CL} & \triangleq & [x].M_{CL} \end{array}$$

con $[x].M$ definida recursivamente como:

$$\begin{array}{llcll} 1) & [x].x & \triangleq & I & \\ 2) & [x].M & \triangleq & KM, & \text{if } M \text{ is a constant or a variable other than } x \\ 3) & [x].(MN) & \triangleq & S([x].M)([x].N) & \end{array}$$

Notar que la variable x no ocurre en el término de CL denotado por $[x].M$.

Proposición A.2.4 (Ver [BKdV03]) Sean $M, N \in \mathrm{Ter}(CL)$.

(i) $([x].M)x \twoheadrightarrow M$.

(ii) $([x].M)N \twoheadrightarrow M\{x \leftarrow N\}$.

Finalmente enunciamos la completitud combinatoria de CL.

Teorema A.2.5 (Completitud Combinatoria) Dado un término de CL M con todas sus variables en $\{x_1 \ldots x_n\}$, existe un término F tal que $Fx_1 \ldots x_n \twoheadrightarrow M$.

A.2.5 El Isomorfismo de Curry-Howard

Presentamos la Deducción Natural para la Lógica Intuicionista proposicional (Int), y mostramos la correspondencia entre su fragmento implicativo y el Cálculo Lambda simplemente tipado. Finalmente, presentamos el cálculo $\lambda\mu$, una extensión del Cálculo Lambda que posee una correspondencia con la Lógica Proposicional Clásica. Para más detalles, ver [Gen35, Pra65, SU06].

La **Lógica Intuicionista**, desarrollada por Arend Heyting [Hey30], es una lógica simbólica que difiere de la lógica clásica en que reemplaza el concepto clásico de *verdad* por el concepto de *demostrabilidad constructiva*.

Asumiendo un conjunto infinito de variables proposicionales $P, Q, \ldots$, el conjunto de fórmulas se define por inducción:

$$A, B ::= \bot \mid P \mid A \supset B \mid A \wedge B \mid A \vee B$$

Las fórmulas A, B, $\neg A$ y $A \Leftrightarrow B$ son abreviaturas de $A \supset \bot$ y $(A \supset B) \wedge (B \supset A)$ respectivamente. Se asume que la implicación $A \supset B$ asocia a derecha. La negación tiene la mayor precedencia, y la implicación la menor, sin prioridad establecida entre $\wedge$ y $\vee$. Es decir, $\neg A_1 \vee A_2 \supset A_3$ significa $((\neg A_1) \vee A_2) \supset A_3$.

Los operadores se interpretan de la siguiente manera.

- Una construcción de $A \wedge B$ consiste de una construcción de A y una construcción de B.
- Una construcción de $A_1 \vee A_2$ consiste de a number $i \in \{1, 2\}$ y una construcción de A_i.
- Una construcción de $A \supset B$ es un método (función) que transforma cada construcción de A en una construcción de B.
- No existe una construcción para $\bot$.

Presentamos ahora la Deducción Natural para Int.

Definición A.2.7

- Un **contexto** Γ es un subconjunto finito de A. Las fórmulas en Γ representan hipótesis. Estas hipótesis pueden estar etiquetadas (ej. x^A en vez de solo A) para permitir la presencia de más de una instancia de la misma fórmula.
- La relación $\Gamma \vdash A$ se define según las reglas de la Figura A.2.
- Escribimos Γ, A (o Γ, x^A) en lugar de $\Gamma \cup \{A\}$ (o $\Gamma \cup \{x^A\}$). También escribimos $\vdash A$ en lugar de $\emptyset \vdash A$.
- Una **prueba** formal de $\Gamma \vdash A$ es un árbol finito, cuyos nodos están etiquetados con pares $\langle \Gamma', A' \rangle$, los cuales se escriben como $\Gamma' \vdash A'$, y cumplen las siguientes condiciones:
  - La etiqueta de la raíz es $\Gamma \vdash A$.
  - Todas las hojas están etiquetadas con axiomas.

$$\frac{}{\Gamma, A \vdash A}\,\mathsf{Ax} \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \supset B}\,\supset\mathsf{I} \qquad \frac{\Gamma \vdash A \supset B \quad \Gamma \vdash A}{\Gamma \vdash B}\,\supset\mathsf{E}$$

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B}\,\vee_{\mathsf{I1}} \qquad \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B}\,\vee_{\mathsf{I2}} \qquad \frac{\Gamma \vdash A \vee B \quad \Gamma, A \vdash B \quad \Gamma, B \vdash B}{\Gamma \vdash B}\,\vee_{\mathsf{E}}$$

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B}\,\wedge_{\mathsf{I}} \qquad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A}\,\wedge_{\mathsf{E1}} \qquad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash B}\,\wedge_{\mathsf{E2}} \qquad \frac{\Gamma \vdash \bot}{\Gamma \vdash A}\,\bot E$$

Figura A.2: Cálculo intuicionista proposicional

  - La etiqueta de cada nodo padre se obtiene a partir de las etiquetas de sus hijos usando una de las reglas.

- Si existe una prueba de $\vdash A$, decimos que A es un **teorema** del cálculo intuicionista proposicional.

Existe un interés natural en las pruebas sin "rodeos". Estas pruebas pueden obtenerse a partir de pruebas arbitrarias por medio de reglas de **normalización de pruebas**, las cuales eliminan la redundancia en una prueba.

Como se mencionó en la introducción, existe una correspondencia precisa entre las pruebas en lógica intuicionista proposicional y los términos tipables en $\lambda^{\rightarrow}$. Veamos el caso del fragmento implicativo de la lógica intuicionista proposicional ($\mathsf{IPC}^{\supset}$).

Si igualamos el conjunto de variables proposicionalesal conjunto de tipos, y asociamos el constructor de tipos $\rightarrow$ al operador lógico $\supset$, entonces el conjunto de fórmulas de $\mathsf{IPC}^{\supset}$ y el conjunto de tipos simples son idénticos. Además, si etiquetamos las hipótesis usadas en una prueba en Deducción Natural, entonces los contextos utilizados en la prueba son también idénticos a los utilizados para el tipado.

Proposición A.2.6 (Isomorfismo de Curry-Howard)

(i) $\Gamma \vdash M : A$ derivable en $\lambda^{\rightarrow} \Rightarrow \Gamma \vdash A$ derivable en $\mathsf{IPC}^{\supset}$.

(ii) $\Gamma \vdash A$ derivable en $\mathsf{IPC}^{\supset} \Rightarrow \exists M \in \mathrm{Ter}(\lambda^{\rightarrow})$ t.q. $\Gamma \vdash M : A$ derivable en $\lambda^{\rightarrow}$.

De este modo, $\lambda^{\rightarrow}$ y $\mathsf{IPC}^{\supset}$ pueden verse esencialmente como dos nombres para un mismo concepto.

Esta correspondencia se extiende al proceso de normalización de pruebas. Un *redex* es un árbol de prueba que contiene la aplicación de una regla de introducción seguida inmediatamente por la aplicación de la regla correspondiente de eliminación. La reducción de términos se corresponde con la normalización de pruebas. Estas y otras correspondencias entre diversas nociones pueden verse en la Fig. A.2.5.

$\lambda^{\rightarrow}$	$\mathsf{IPC}^{\supset}$
variable de término	hipótesis
término	construcción (prueba)
variable de tipo	variable proposicional
tipo	fórmula
constructor de tipos	conectivo
habitabilidad	demostrabilidad
término tipable	prueba de una proposición
redex	árbol de prueba con redundancia
reducción	normalización

A.2.6 Deducción Natural Clásica

Parigot [Par92] introdujo la Deducción Natural Clásica (CND) con el fin de evitar los problemas que surgían de intentar usar $\neg\neg A \to A$ como tipo para los operadores de control a la vez que se restringían los juicios de tipado a juicios con una única conclusión (i.e. un único tipo). Además de servir como presentación en Deducción Natural para la lógica clásica, CND fue la base para un cálculo algorítmico llamado $\lambda\mu$. Las fórmulas de CND (de segundo orden) se construyen con la siguiente gramática:

$$A, B ::= P(s_1, \ldots, s_n) \mid A \supset B \mid \forall x.A \mid \forall X.A$$

donde los s_i son términos de primer orden, x pertenece al conjunto de individuos, y X al de proposiciones. Las letras Γ, Δ denotan conjuntos de fórmulas. Los secuentes $\Gamma \vdash \Delta$ se interpretan del modo habitual: Γ se entiende como una conjunción de premisas, y Δ como una disyunción de conclusiones. Las reglas de inferencia pueden verse en la Fig. A.3 (donde y (resp. Y) no está libre en la conclusión de la introducción de primer orden (resp. segundo orden) del operador "$\forall$"). El cálculo $\lambda\mu$ [Par92] es una extensión del Cálculo Lambda que introduce dos nuevos operadores: "μ" y "[ ]". Según el isomorfismo de Curry-Howard, el Cálculo Lambda puede expresar los express teoremas de la lógica intuicionista, pero no todos los teoremas de la lógica clásica. En cambio, con estos nuevos operadores es posible construir términos cuyo tipo sea, por ejemplo, la ley del tercero excluido: $(A \vee \neg A)$, o la ley de Peirce: $(((A \supset B) \supset A) \supset A)$. Este cálculo fue introducido como una asignación de términos para una presentación de CND de segundo orden. El cálculo $\lambda\mu$ tipado utiliza dos tipos de variables, a saber variables λ: x, y, z y variables μ: α, β, γ. Las variables λ se usan para etiquetar fórmulas sel lado izquierdo de un secuente, mientras que las variables μ etiquetan las fórmulas del lado derecho (excepto por a lo sumo una fórmula del lado derecho,que puede no estar etiquetada). Distintas fórmulas no pueden tener la misma etiqueta.

Definición A.2.8 Los términos de $\lambda\mu$ con y sin nombre se definen por recursión mutua de la siguiente manera:

$$\begin{array}{rcll} T, U & ::= & x \mid \lambda x^A.U \mid T\ U \mid \mu\alpha^A.N & \text{(términos sin nombre)} \\ N & ::= & [\alpha^A]U & \text{(términos con nombre)} \end{array}$$

$$A \vdash A$$

$$\frac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \supset B, \Delta} \supset I \qquad \frac{\Gamma \vdash A \supset B, \Delta \quad \Gamma' \vdash A, \Delta'}{\Gamma \cup \Gamma' \vdash B, \Delta \cup \Delta'} \supset E$$

$$\frac{\Gamma, A \vdash \Delta}{\Gamma \vdash \neg A, \Delta} \neg I \qquad \frac{\Gamma \vdash \neg A, \Delta \quad \Gamma' \vdash A, \Delta'}{\Gamma \cup \Gamma' \vdash \Delta \cup \Delta'} \neg E$$

$$\frac{\Gamma \vdash A\{x \leftarrow y\}, \Delta}{\Gamma \vdash \forall x.A, \Delta} \forall \mathsf{I}_1 \qquad \frac{\Gamma \vdash \forall x.A, \Delta}{\Gamma \vdash A\{x \leftarrow T\}, \Delta} \forall \mathsf{E}_1$$

$$\frac{\Gamma \vdash A\{X \leftarrow Y\}, \Delta}{\Gamma \vdash \forall X.A, \Delta} \forall \mathsf{I}_2 \qquad \frac{\Gamma \vdash \forall X.A, \Delta}{\Gamma \vdash A\{X \leftarrow T\}, \Delta} \forall \mathsf{E}_2$$

Figura A.3: Reglas de Deducción Natural Clásica (de segundo orden)

Las reglas de tipado de $\lambda\mu$ pueden verse en la Fig. A.4.

De acuerdo con la correspondencia entre pruebas y términos, las reglas de normalización pueden expresarse como reglas de reducción de términos [Par97]:

$$\begin{array}{lllll}
R_1: & (\lambda x^A.T)U & \to & T\{x \leftarrow U\} & \\
R_2: & (\mu\alpha^{A\supset B}.N)U & \to & \mu\beta^B.N(\!|\, [\alpha^{A\to B}](\bullet) \leftarrow [\beta^B](\bullet)U \,|\!) & \\
S_1: & [\beta^A]\mu\alpha^A.N & \to & N\{\alpha^A \leftarrow \beta^A\} & \\
S_2: & \mu\alpha^A.[\alpha^A]U & \to & U, & \text{if } \alpha^A \notin \mathsf{FV}(U)
\end{array}$$

donde la sustitución estructural $n(\!|\, [\alpha^{A\to B}](\bullet) \leftarrow [\beta^B](\bullet)U \,|\!)$ se define como:

$$\begin{array}{llll}
x & (\!|\, [\alpha^{A\to B}](\bullet) \leftarrow [\beta^B](\bullet)U \,|\!) & \triangleq & x \\
\lambda x^C.T & (\!|\, [\alpha^{A\to B}](\bullet) \leftarrow [\beta^B](\bullet)U \,|\!) & \triangleq & \lambda x^C.(T(\!|\, [\alpha^{A\to B}](\bullet) \leftarrow [\beta^B](\bullet)U \,|\!)) \\
(T\ T') & (\!|\, [\alpha^{A\to B}](\bullet) \leftarrow [\beta^B](\bullet)U \,|\!) & \triangleq & (T(\!|\, [\alpha^{A\to B}](\bullet) \leftarrow [\beta^B](\bullet)U \,|\!)\ (T'(\!|\, [\alpha^{A\to B}](\bullet) \leftarrow [\beta^B](\bullet)U \,|\!))) \\
[\alpha^{A\supset B}]T & (\!|\, [\alpha^{A\to B}](\bullet) \leftarrow [\beta^B](\bullet)U \,|\!) & \triangleq & [\beta^B](T\ U) \\
[\gamma^C]T & (\!|\, [\alpha^{A\to B}](\bullet) \leftarrow [\beta^B](\bullet)U \,|\!) & \triangleq & [\gamma^C](T(\!|\, [\alpha^{A\to B}](\bullet) \leftarrow [\beta^B](\bullet)U \,|\!)), \text{ if } \gamma^C \neq \alpha^{A\supset B} \\
\mu\alpha^{A\supset B}.N & (\!|\, [\alpha^{A\to B}](\bullet) \leftarrow [\beta^B](\bullet)U \,|\!) & \triangleq & \mu\alpha^{A\supset B}.N \\
\mu\gamma^C.N & (\!|\, [\alpha^{A\to B}](\bullet) \leftarrow [\beta^B](\bullet)U \,|\!) & \triangleq & \mu\gamma^C.(N(\!|\, [\alpha^{A\to B}](\bullet) \leftarrow [\beta^B](\bullet)U \,|\!)), \text{ if } \gamma^C \neq \alpha^{A\supset B}
\end{array}$$

La regla R_1 corresponde a la β-reducción en el Cálculo Lambda. R_2 corresponde a la reducción estructural. En [Sau10], estas reglas de llaman β, μ, ρ y θ respectivamente.

$\lambda\mu$ satisface las propiedades estándar del Cálculo Lambda, en particular Confluencia del cálculo tipado y sin tipos [Par92] y Preservación de Tipos [Par92]. SN también se satisface para el cálculo $\lambda\mu$ tipado [Par97].

Reglas lógicas:

$$x : A^x \vdash A$$

$$\frac{U : \Gamma, A^x \vdash B, \Delta}{\lambda x^A.U : \Gamma \vdash A \supset B, \Delta} \qquad \frac{T : \Gamma \vdash A \supset B, \Delta \quad U : \Gamma' \vdash A, \Delta'}{(T\ U) : \Gamma \cup \Gamma' \vdash B, \Delta \cup \Delta'}$$

$$\frac{U : \Gamma \vdash A\{y \leftarrow x\}, \Delta}{U : \Gamma \vdash \forall x.A, \Delta} \qquad \frac{U : \Gamma \vdash \forall x.A, \Delta}{U : \Gamma \vdash A\{x \leftarrow T\}, \Delta}$$

$$\frac{U : \Gamma \vdash A\{Y \leftarrow X\}, \Delta}{U : \Gamma \vdash \forall X.A, \Delta} \qquad \frac{U : \Gamma \vdash \forall X.A, \Delta}{U : \Gamma \vdash A\{X \leftarrow T\}, \Delta}$$

Reglas de nombres:

$$\frac{T : \Gamma \vdash A, \Delta}{[\alpha^A]T : \Gamma \vdash A^\alpha, \Delta} \qquad \frac{N : \Gamma \vdash A^\alpha, \Delta}{\mu\alpha^A.N : \Gamma \vdash A, \Delta}$$

Figura A.4: Axiomas y reglas de inferencia de $\lambda\mu$.

A.3 Lógicas Combinatorias Para Cálculos Lambda con Patrones

En esta sección (correspondiente al Capítulo 3 de esta tesis) exploramos un sistema de combinadores para el cálculo de patrones λP. Como se mencionó en la introducción, nuestro objetivo es desarrollar un sistema de reescritura basado en combinadores que pueda simular el mecanismo de abstracción de λP. Para esto introducimos el sistema CL_P.

Recordemos el conjunto de términos de λP, presentado en la introducción:

$$M, N, P ::= x \mid MN \mid \lambda P.M$$

La regla de reducción en λP, β_P, generaliza a la regla β original de la siguiente manera:

$$(\lambda P.M)P^\sigma \to_{\beta_P} M^\sigma$$

donde σ es una sustitución simultánea de las variables libres de P por términos de λP. No se admite la reducción dentro de los patrones

Como se mencionó en la introdución, la reducción irrestricta en λP no es confluente. Se han definido dos restricciones para garantizar la confluencia de la reducción [KvOdV08]. Estas son RPC (Def. 3.1.4) y RPC^+ (Def. 3.1.5). Para nuestro trabajo, tomamos como base RPC^+, ya que esta permite una caracterización sintáctica de los patrones admisibles.

A.3.1 El Cálculo CL_P

El conjunto de términos de CL_P es el descripto por la gramática:

$$M, N ::= x \mid K_M \mid S_M \mid \Pi^1_{MN} \mid \Pi^2_{MN} \mid MN$$

donde x pertenece a un conjunto infinito numerable de variables $\mathcal{X}$. La aplicación asocia a izquierda, como es habitual. K_M, S_M, Π^1_{MN} y Π^2_{MN} son los **combinadores**, o funciones primitivas de nuestr cálculo. Los combinadores K_M y S_M son versiones de los combinadores de CL decoradas con patrones. Π^1_{MN} y Π^2_{MN}, denominados **proyectores**, se introducen con el propósito de extraer información de una aplicación a través de la descomposición de la misma. Llamaremos simplemente "términos" a los términos de CL_P cuando no haya ambigüedad. Si bien la gramática admite términos arbitrarios como subíndices de K, S, Π^1 y Π^2, restringiremos las opciones a un subconjunto de los términos, a cuyos elementos llamaremos **patrones** (Def. 3.2.5). Usaremos las letras P y Q para referirnos a patrones.

El propósito de los términos de CL_P es reproducir los 3 pasos requeridos para la reducción en λP:

1. verificar el *matching* entre el patrón y el argumento, y en caso de éxito
2. obtener una asociación de las variables libres del patrón con subtérminos del argumento,
3. para luego aplicarla al cuerpo de la abstracción.

En CL_P, los combinadores de la forma S_P y K_P se encargan del tercer paso, los proyectores del segundo, y los subíndices del primero.

Notar que existen infinitos combinadores: un S_P, K_P, Π^1_{PQ}, Π^2_{PQ} para cada posible P y Q. Por ejemplo, los términos K_x, K_{K_y} y K_{S_y} son distintos combinadores. Los subíndices de los combinadores son decoraciones, y no son afectados por las sustituciones (ver Def. 3.2.3).

Adoptamos las siguientes convenciones:

- Las variables libres se asumen distintas de las variables en los patrones, y estas últimas son distintas entre sí, aun si sus nombres coinciden. No existen variables ligadas en este cálculo; la presencia de una variable en un patrón sirve para indicar que cualquier término producirá un *matching* exitoso.
- Consideraremos que dos patrones son iguales si solo difieren en los nombres de sus variables, y trabajaremos módulo renombre de variables en patrones (Def. 3.2.6).

Los términos admitidos como patrones son los que satisfacen una restricción llamada RPC^{++} (ver Sección 3.2.1), la cual es una generalización de RPC^{+} adaptada al escenario combinatorio. El objetivo de RPC^{++} es lograr la otrogonalidad – y por ende la confluencia – de CL_P, al impedir que los patrones unifiquen con términos que contengan redexes.

Definición A.3.1 (Reducción W_P) La **reducción** W_P (notación: $\to_{W_P}$) está dada por el siguiente TRS definido sobre los términos de CL_P:

$$\begin{array}{lcll} K_P x P & \to & x, & x \notin \mathsf{FV}(P) \\ S_P x y P & \to & xP(yP), & x, y \notin \mathsf{FV}(P) \\ \Pi^1_{PQ}(PQ) & \to & P & \\ \Pi^2_{PQ}(PQ) & \to & Q & \end{array}$$

donde P y Q son patrones, y la aplicación PQ – donde aparece – también es un patrón.

Las reglas son esquemáticas, ya que existen infinitas opciones para P y Q. Tenemos, de hecho, infinitas reglas capturadas por una cantidad finita de esquemas.

Hemos definido mapeos en ambos sentidos entre λP y CL_P ($-_{CL}$ y $-_\lambda$), extendiendo los mapeos originales entre el Cálculo Lambda y CL. Las definiciones de los mismos pueden verse en la Sección 3.3.1.

A.3.2 Resultados principales

- CL_P es una extensión de CL.
- W_P es ortogonal en CL_P, y por lo tanto es confluente.
- Todos los patrones de λP que satisfacen RPC^+ se traducen a patrones de CL_P. De hecho la restricción RPC^{++} es más débil que RPC^+, permitiendo un conjunto mayor de patrones.
- La traducciíon preserva el *pattern-matching*: para todo término de λP P tal que P_{CL} es un patrón de CL_P, y para todo término de $\lambda P M$: $(\exists\sigma \text{ s.t. } M = P^\sigma) \Leftrightarrow (\exists\sigma' \text{ s.t. } M_{CL} = P_{CL}^{\sigma'})$, donde σ es una subtitución en λP y σ' es una sustitución en CL_P.
- Para todo término de λP M y patrón P representables en CL_P, y para toda sustitución σ en λP t.q. $\mathsf{dom}(\sigma) \subseteq \mathsf{FV}(P)$: $((\lambda P.M)P^\sigma)_{CL} \twoheadrightarrow_{W_P} (M^\sigma)_{CL}$.
- Para todos los términos de CL_P M, N: si $M \to_{W_P} N$ entonces $M_\lambda \xrightarrow{+}_{\beta_P} N_\lambda$.
- Para todo término de λP M: $(M_{CL})_\lambda \twoheadrightarrow_{\beta_P} M$.
- Para todos los términos de CL_P M, N: si $M \equiv_{W_P} N$, entonces $M_\lambda \equiv_{\beta_P} N_\lambda$.
- La traducción $-_{CL}$ preserva unificación y *matching* de alto orden.
- Para todos los términos de λP M, N, si $M \equiv_{\beta_P} N$ entonces $M_{CL} \equiv_{W_P} N_{CL}$.

Hemos definido también un sistema de tipos simple para CL_P, que satisface las propiedades de Preservación de Tipos y Normalización Fuerte para términos tipables. Definimos también variantes de CL_P incluyendo constructores, un combinador θ que modela el comportamiento de un *case* – permitiendo capturar la reducción del cálculo de patrones λC [KvOdV08] – una versión generalizada (condicional) del proceso de *pattern matching* – capturando una restriccón del cálculo *SF* [JGW11] – y una variante de CL_P llamada $CL_\star$, en la cual el *matching* del meta-nivel pasa a estar embebido dentro del sistema, capturando las nociones de *matching* y reducción de CL_P con un conjunto finito – aunque más verboso – de reglas.

A.4 Lógica Hipotética de Pruebas Proposicional

Proponemos una presentación en Deducción Natural para la Lógica de Pruebas proposicional, llamada Lógica Hipotética de Pruebas (HLP). La intertraducibilidad entre ambos sistemas es establecida. Propiedades esperadas de HLP son probadas, tales como normalización fuerte y confluencia, con relación a una asignación de términos (extensión del Cálculo Lambda) introducida para este sistema.

A.4.1 La Lógica de Pruebas

Recordemos de la introducción que LP es una lógica en la cual los conectivos proposicionales habituales se extienden con uno nuevo: dados un polinomio de prueba s y una proposición A, construimos $[\![s]\!]A$. Esto se lee como: "*s es una prueba de A*".

Definition A.4.1 (Polinomios de prueba y Fórmulas) El conjunto de polinomios de prueba y fórmulas de LP se define como:

$$\begin{array}{rcl} s, t & ::= & x^A \mid c \mid s \cdot t \mid !s \mid s + t \\ A, B & ::= & P \mid \bot \mid A \supset B \mid [\![s]\!]A \end{array}$$

Las pruebas se representan con **polinomios de prueba**, los cuales se construyen a partir de **variables de prueba** y **constantes de prueba** por medio de símbolos de función para operaciones computables elementales sobre pruebas, los símbolos binarios "$\cdot$", "$+$", y el unario "!". "$\cdot$" es el operador de aplicación tradicional. "$+$" representa la concatenación de pruebas (o unión de pruebas). El operador "!" se conoce como **chequeador de pruebas**, y puede entenderse como un medio para verificar la validez de una prueba. Las variables de prueba se utilizan como hipótesis y se obtienen de un conjunto infinito numerable. Las fórmulas de LP se obtienen aplicando conectivos Booleanos a partir de átomos proposicionales, más las proposiciones $[\![s]\!]A$, donde s es un polinomio de prueba y A es una fórmula. Las operaciones de LP están dadas por los siguientes esquemas (donde A, B son fórmulas arbitrarias y s, t son polinomios de prueba arbitrarios).

Definición A.4.2 Recordemos de la introducción los axiomas y reglas de inferencia de LP:

A0. Axiomas de la lógica proposicional clásica en el lenguaje de LP
A1. $[\![s]\!]A \supset A$
A2. $[\![s]\!](A \supset B) \supset ([\![t]\!]A \supset [\![s \cdot t]\!]B)$
A3. $[\![s]\!]A \supset [\![!s]\!][\![s]\!]A$
A4. $[\![s]\!]A \supset [\![s + t]\!]A$
A5. $[\![t]\!]A \supset [\![s + t]\!]A$
MP. $\vdash A \supset B$ y $\vdash A \Rightarrow \vdash B$
Nec. A axioma $\mathbf{A0} - \mathbf{A5} \Rightarrow \vdash [\![c]\!]A$

El axioma **A1** se lee:"*si s es una prueba de A, entonces A es verdadera*". **A2** se lee: "*si s es una prueba de $A \supset B$ y t es una prueba de A, entonces $s \cdot t$ es una prueba de B*". Luego "$\cdot$" representa la composición de pruebas. **A3** se lee: "*si s es una prueba de A, entonces $!s$ es una prueba de la sentencia* 's es una prueba de A' ", de modo que $!s$ es visto como un cómputo que verifica $[\![s]\!]A$. La suma se lee: "*si s es una prueba de A, $s + t$ también lo es, independientemente de la forma de t*". Usamos las letras π, π' para nombrar derivaciones en LP.

Una diferencia menor entre nuestra presentación de polinomios de prueba y la de [Art95, Art01] es que las variables de prueba se asumen decoradas con una proposición. Esto no induce una pérdida de generalidad ya que las variables de prueba en LP pueden usarse libremente ignorando esta decoración. Por otro lado, el beneficio es que al razonar bajo un contexto

de hipótesis Γ, las variables servirán para etiquetar proposiciones y esto nos ayudará en la traducción de LP a HLP.

Una **especificación de constantes** ($\mathcal{CS}$) es un conjunto finito de fórmulas $[\![c_1]\!]A_1, \ldots, [\![c_n]\!]A_n$ tales que c_i es una constante, y A_i es un axioma **A0-A5**. Cada derivación en LP genera naturalmente el $\mathcal{CS}$ que consiste de todas las fórmulas introducidas en la derivación por la regla de Necesitación. Una especificación de constantes $\mathcal{CS}$ es **inyectiva** si para cada constante c existe a lo sumo una fórmula $[\![c]\!]A \in \mathcal{CS}$ (cada constante denota una prueba de a lo sumo un axioma). Podemos distinguir en nuestra notación el axioma al que hace referencia una constante escribiendo el nombre del axioma como superíndice y las instancias de sus metavariables como subíndices. Por ejemplo, escribimos $c^{\mathbf{A1}}_{x^P,P}$ para la constante correspondiente a la instancia $[\![x^P]\!]P \supset P$ del axioma **A1** (y de manera similar para los otros axiomas).

Con frecuencia utilizamos deducción bajo la asunción de un **contexto** de hipótesis. Un contexto Γ en LP es un conjunto de hipótesis representadas por variables de prueba t.q. si $x^{A_1} \in \Gamma$ y $x^{A_1} \in \Gamma$, entonces $A_1 = A_2$. Una misma fórmula puede aparecer en un contexto más de una vez con distintos nombres (i.e. variables diferentes). En ocasiones escribimos Γ, A para indicar Γ, x^A con x un nombre fresco, y $A \in \Gamma$ significa que existe una hipótesis x^A en Γ. Utilizaremos a, b, c, v, w, x, y, z, α como nombres de hipótesis en contextos de LP. Sea $\Gamma = \{x_1^{A_1}, \ldots, x_n^{A_n}\}$ un contexto y $\vec{u} = u_1, \ldots, u_n$ una lista de polinomios de prueba, definimos $[\![\vec{u}]\!]\Gamma$ como $\{{x'_1}^{[\![u_1]\!]A_1}, \ldots, {x'_n}^{[\![u_n]\!]A_n}\}$, con $x'_1, \ldots, x'_n$ nombres frescos. Notar que las variables $x_i^{A_i}$ y ${x'_i}^{[\![u_i]\!]A_i}$ tienen distintos tipos (distintas decoraciones). Además, escribimos $\rhd_{\mathsf{LP}} \Gamma \vdash A$ para indicar que A es demostrable en LP bajo hipótesis Γ. Decimos que un juicio $\Gamma \vdash A$ es *derivable* en LP para indicar que $\rhd_{\mathsf{LP}} \Gamma \vdash A$.

Un resultado destacable de la Lógica de Pruebas es la Internalización:

Lema A.4.3 (Internalización) Si $\rhd_{\mathsf{LP}} [\![\vec{u}]\!]\Gamma \vdash D$, entonces existe un polinomio de prueba r tal que $\rhd_{\mathsf{LP}} [\![\vec{u}]\!]\Gamma \vdash [\![r]\!]D$.

A.4.2 Lógica Hipotética de Pruebas

Una **fórmula** en HLP es igual que en LP, excepto porque los $r, s, \ldots$ son testigos de prueba en lugar de polinomios de prueba.

Definition A.4.4 (Proof testigo) Un **testigo de prueba** es una expresión generada por la siguiente gramática:

$$
\begin{array}{rcll}
r, s, t & ::= & x^A & \text{hipótesis de verdad} \\
 & | & v^A & \text{hipótesis de validez} \\
 & | & \lambda x^A.s & \text{abstracción} \\
 & | & s \cdot t & \text{aplicación} \\
 & | & !s & \text{bang} \\
 & | & t\langle v^A{:=}r, s\rangle & \text{unbox} \\
 & | & s + t & \text{suma} \\
 & | & [\alpha^A]s & \text{aplicación de nombre} \\
 & | & \mu\alpha^A.s & \text{abstracción de nombre}
\end{array}
$$

En $\lambda x^A.s$, el alcance de la variable ligada x^A es s; en $t\langle v^A{:=}r, s\rangle$ el alcance de la variable ligada v^A es t y en $\mu\alpha^A.s$ el alcance de la variable ligada α^A es s. Además, $!s$ liga todas las ocurrencias libres de variables de verdad y falsedad en s; del mismo modo, $t\langle v^A{:=}r, s\rangle$ liga todas las ocurrencias libres de variables de verdad y falsedad en r. Las abstracciones, aplicaciones, abstracciones de nombres y aplicaciones de nombres son similares a sus análogas en $\lambda\mu$ (ver Sección 2.5.4), con la diferencia de que aquí se las utiliza como testigos de prueba en lugar de términos. Los operadores "!" y "+" son similares a los de LP. Los testigos de prueba de la forma $t\langle v^A{:=}r, s\rangle$ pueden leerse como "reemplazar todas las ocurrencias libres de v^A por r en la fórmula cuyo testigo es t, siendo s testigo de la verdad de $[\![r]\!]A$".

Asumimos la siguiente **convención de variables**: los nombres de todas las variables ligadas son distintos entre sí, y distintos de todas las variables libres. Asumimos también que la aplicación "$\cdot$" y la suma "+" asocian a izquierda, y la implicación "$\supset$" asocia a derecha. Escribimos "$\neg A$" para abreviar "$A \supset \bot$". Los operadores "!", "$\neg$" y "$[\![\ ]\!]$" tienen precedencia sobre "$\cdot$", "+" y "$\supset$", los cuales a su vez tienen precedencia sobre "λ", "μ" y "$[\ ]$". Por ejemplo, $[\alpha^{([\![r]\!]A)\supset((\neg B)\supset C)}]((!s) + t)$ puede escribirse como $[\alpha^{[\![r]\!]A\supset\neg B\supset C}]!s + t$.

Definition A.4.5 (Contextos y juicios para HLP) Un **contexto de verdad** (Γ) es un conjunto de hipótesis de verdad $\{x_1^{A_1}, \ldots, x_n^{A_n}\}$; un **contexto de validez** (Θ) es un conjunto de variables de validez $\{v_1^{A_1}, \ldots, v_m^{A_m}\}$; un **contexto de falsedad** (Δ) es un conjunto de variables de falsedad $\{\alpha_1^{A_1}, \ldots, \alpha_k^{A_k}\}$. Escribimos $\cdot$ para denotar el contexto vacío. Un **juicio** es una expresión de la forma:

$$\Theta; \Gamma; \Delta \vdash A \,|\, s$$

A menudo será conveniente abreviar $\Theta; \Gamma; \Delta$ para facilitar la lectura. Usaremos $\mathcal{H}$ para este fin y nos referiremos a $\mathcal{H}$ como un *contexto compuesto*. De este modo, el juicio anterior puede escribirse como:

$$\mathcal{H} \vdash A \,|\, s \tag{A.8}$$

Además escribimos:

$$\begin{array}{l} \mathcal{H}, x^A \text{ para } \Theta; \Gamma, x^A; \Delta \\ \mathcal{H}, \alpha^A \text{ para } \Theta; \Gamma; \Delta, \alpha^A \\ \mathcal{H}, v^A \text{ para } \Theta, v^A; \Gamma; \Delta \end{array}$$

Finalmente, scribimos $\rhd_{\mathsf{HLP}} \mathcal{H} \vdash A \,|\, s$, para indicar que el juicio $\mathcal{H} \vdash A \,|\, s$ es derivable en HLP.

Los conjuntos de **variables libres de validiez, verdad y falsedad** en una fórmula A se denotan como $\mathsf{FVT}(A)$, $\mathsf{FVV}(A)$ y $\mathsf{FVF}(A)$ respectivamente.

Los axiomas y reglas de inferencia de HLP pueden verse en la figura A.5.

El axioma Var indica que el juicio $\Theta; \Gamma, x^A; \Delta \vdash A \,|\, x^A$ es evidente por sí mismo. En efecto, si asumimos que x^A es testigo de que la proposición A es verdadera, podemos concluir inmediatamente que A es verdadera con testigo x^A.

La regla de introducción de la modalidad $[\![t]\!]$ internaliza la evidencia del meta-nivel en la lógica objeto. Afirma que si s es evidencia incondicional de que A es verdadera, entonces A es de hecho válida con testigo s o, de forma más general, cualquier testigo de prueba t equivalente a

$$\frac{}{\mathcal{H}, x^A \vdash A \,|\, x^A}\ \mathsf{Var}$$

$$\frac{\mathcal{H}, x^A \vdash B \,|\, s}{\mathcal{H} \vdash A \supset B \,|\, \lambda x^A.s}\ \supset\mathsf{I} \qquad \frac{\mathcal{H} \vdash A \supset B \,|\, s \quad \mathcal{H} \vdash A \,|\, t}{\mathcal{H} \vdash B \,|\, s \cdot t}\ \supset\mathsf{E}$$

$$\frac{}{\mathcal{H}, v^A \vdash A \,|\, v^A}\ \mathsf{VarM}$$

$$\frac{\Theta; \cdot; \cdot \vdash B \,|\, s \quad \Theta; \cdot; \cdot \vdash s \equiv t : B}{\Theta; \Gamma; \Delta \vdash [\![t]\!]B \,|\, !t}\ \Box\mathsf{I} \quad \frac{\mathcal{H} \vdash [\![r]\!]A \,|\, s \quad \mathcal{H}, v^A \vdash C \,|\, t}{\mathcal{H} \vdash C\{v^A \leftarrow r\} \,|\, t\langle v^A{:=}r, s\rangle}\ \Box\mathsf{E}$$

$$\frac{\mathcal{H} \vdash A \,|\, s}{\mathcal{H} \vdash A \,|\, s + t}\ \mathsf{PlusL} \qquad \frac{\mathcal{H} \vdash A \,|\, t}{\mathcal{H} \vdash A \,|\, s + t}\ \mathsf{PlusR}$$

$$\frac{\mathcal{H}, \alpha^A \vdash A \,|\, s}{\mathcal{H}, \alpha^A \vdash \bot \,|\, [\alpha^A]s}\ \mathsf{Name} \qquad \frac{\mathcal{H}, \alpha^A \vdash \bot \,|\, s}{\mathcal{H} \vdash A \,|\, \mu\alpha^A.s}\ \mathsf{NAbs}$$

Figura A.5: Axiomas y reglas de inferencia de HLP.

s (con la equivalencia definida en la Sección 4.2.1). La evidencia de la verdad de $[\![t]\!]A$ se construye a partir de la evidencia (verificada) de que A es incondicionalmente verdadera, prefijándola con un *bang*. La capacidad de reemlazar el testigo s por otro equivalente es necesaria para el correcto funcionamiento del proceso de normalización de pruebas (ver Sección 4.2.1 para más detalles).

Una propiedad importante de la equivalencia de testigos es que si $\rhd_{\mathsf{HLP}} \Theta; \Gamma; \Delta \vdash s \equiv t : D$, tanto $\Theta; \Gamma; \Delta \vdash D \,|\, s$ como $\Theta; \Gamma; \Delta \vdash D \,|\, t$ son derivables.

La regla $\Box$E permite descargar hipótesis de validez. Para poder descargar la hipótesis de validez v^A,es necesaria una prueba de la validez de A. En este sistema, esto requiere probar que $[\![r]\!]A$ es verdadera con algún testigo s, para algún testigo r. Notar que r es testigo de que A es incondicionalmente verdadera (i.e. válida) mientras que s es testigo de la verdad de $[\![r]\!]A$. Luego se reemplazan todas las ocurrencias libres de v por r en la proposición C. Esta construcción se registra con el testigo $t\langle v^A{:=}r, s\rangle$ en la conclusión, indicando que s es una prueba de que r puede usarse en lugar de v^A en t. Esto tiene el efecto práctico de permitirnos extraer el testigo r de su "caja" en $[\![r]\!]A$. La expresión $C\{v^A \leftarrow r\}$ denota la sustitución de v^A por r en C. Esta y otras formas de sustitución se explican en detalle en la Sección 4.4.1. Como nota final sobre $\Box$E, la inclusión de s en su testigo es requerida para la prueba de que las fórmulas derivables en HLP también son derivables en LP (Sección 4.3) y también para la Preservación de Tipos.

Con respecto a las reglas de la suma, comentamos la regla PlusL, siendo el caso de PlusR análogo. Informalmente, el testigo de prueba $s + t$ testifica que o bien s o t es un testigo de la verdad de A *sin* brindar detalles sobre cuál de los dos lo es. Notar que t puede ser cualquier testigo de prueba. Puede incluso contener variables libres que no estén presentes en Θ, Γ ni Δ (ver también el ejemplo 4.2.7). El motivo de esto es que deseamos preservar los *teoremas* de LP

en HLP, en particular $[\![s]\!]A \supset [\![s+t]\!]A$, el cual no impone restricciones sobre t.

Finalmente, las reglas Name y NAbs fueron introducidas por Parigot en el cálculo $\lambda\mu$. La intuición detrás de ellas es que las hipótesis en Δ deben considerarse negadas. Bajo esta lectura, la regla Name indica que a partir de $\neg A$ y A podemos deducir $\bot$. De modo similar, NAbs es la regla clásica de la negación que indica que si llegamos a una contradicción a partir de la hipótesis $\neg A$, podemos descargar esta hipótesis y obtener A.

La relación entre LP y HLP y las respectivas traducciones pueden verse en la Sección 4.3.

A.4.3 Asignación de Términos

Si bien los testigos de prueba contienen suficiente información para garantizar que una fórmula es derivable, no contienen suficiente información para razonar sobre la derivación. Por ejemplo, el testigo $v^A + w^A$ asegura que A es verdadera si asumimos v^A y w^A, pero no nos indica cuál de las hip'otesis se utilizó para derivarlo. De hecho, hay derivaciones posibles, una usando v^A y PlusL, y la otra usando w^A y PlusR. De manera similar, $!v^A$ puede utilizarse para verificar que $[\![v^A]\!]A$ es verdadera asumiendo v^A como hipótesis de validez, pero esto puede derivarse de infinitas maneras, usando $\Box$I con cualquier testigo que sea equivalente a v^A (por ejemplo, el propio v^A, $(\lambda x^A.x^A)\cdot v^A$, $\mu\alpha^A.[\alpha^A]v^A$, etc.). Para poder razonar sobre las derivaciones y obtener resultados sobre la metateoría, es conveniente codificar las pruebas como *términos* que brinden mayor información, permitiéndonos razonar sobre pruebas y fórmulas a través de su representación como términos y tipos. Para esto definimos el sistema λ^{LP}, la asignación de términos para HLP.

Definición A.4.6 (Términos de λ^{LP}) El conjunto de términos de λ^{LP} se define como:

$$
\begin{array}{rcl}
M, N & ::= & x^A \\
& | & v^A \\
& | & (\lambda x^A.M^B)^{A\supset B} \\
& | & (M^{A\supset B}N^A)^B \\
& | & (!M^A)^{[\![s]\!]A} \\
& | & (N^B\langle v^A{:=}r, M^{[\![r]\!]A}\rangle)^{B\{v^A\leftarrow r\}} \\
& | & ([\alpha^A]M^A)^{\bot} \\
& | & (\mu\alpha^A.M^{\bot})^A \\
& | & (M^A +_{\mathrm{L}} s)^A \mid (s +_{\mathrm{R}} N^B)^B
\end{array}
$$

Volviendo a nuestros ejemplos, el término $(v^A +_{\mathrm{L}} w^A)^A$ codifica una prueba de A usando T-PlusL y v^A, y no la alternativa (la cual se codificaría como $(v^A +_{\mathrm{R}} w^A)^A$). Y el término $(!((\lambda x^A.x^A)^{A\supset A}v^A)^A)^{[\![v^A]\!]A}$ codifica una prueba de $[\![v^A]\!]A$ que usa $(\lambda x^A.x^A)\cdot v^A$ como testigo para las premisas, y no v^A, $\mu\alpha^A.[\alpha^A]v^A$ nni otro testigo equivalente. De modo similar, los términos $(!v^A)^{[\![v^A]\!]A}$ y $(!v^A)^{[\![(\lambda x^A.x^A)\cdot v^A]\!]A}$ codifican distintas derivaciones, que se utilizan para probar distintas fórmulas. Es por esto que las anotaciones sobre un ! son importantes.

No toda la información sobre la derivación está codificada en los términos, ya que los mismos no contienen información sobre las reglas de equivalencia utilizadas para derivar la segunda

premisa de T-$\Box\mathsf{I}$, ni los contextos usados en las derivaciones (pueden haberse asumido hipótesis adicionales que no se utilizaron). Sin embargo, estos términos proveen suficiente información para razonar sobre el proceso de normalización de pruebas y otras propiedades de la metateoría.

Los juicios de tipado en λ^{LP} tienen la forma $\Theta; \Gamma; \Delta \vdash M^A \,|\, s$, donde M es un término que codifica la derivación en HLP de $\Theta; \Gamma; \Delta \vdash A \,|\, s$, y A es a su vez el tipo de M en los contextos Θ, Γ, Δ. Los axiomas y reglas de tipado pueden verse en la Figura 4.6, en la Sección 4.4.

Definición A.4.7 (Reducción en λ^{LP}) La reducción en λ^{LP} se define como la clausura compatible de las siguientes reglas. El primer conjunto de reglas proviene de los casos principales de normalización de derivaciones, y el segundo de los casos permutativos.

Reglas principales:

$$
\begin{array}{lllll}
\beta: & (\lambda x^A.M^B)N^A & \to & M^B\{x^A \leftarrow N^A\}, & \\
\gamma: & M^B\langle v^A{:=}r,\ !N^A\rangle & \to & M^B\{v^A \leftarrow N^A, r\}, & \text{si } \mathsf{FVT}(N^A) = \mathsf{FVF}(N^A) = \emptyset \\
\mu: & [\beta^A]\mu\alpha^A.M^\perp & \to & M^\perp\{\alpha^A \leftarrow \beta^A\} & \\
\zeta: & (\mu\alpha^{A\supset B}.M^\perp)N^A & \to & \mu\beta^B.M^\perp(\!|\, [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)N^A \,|\!) & \\
\theta: & \mu\alpha^A.[\alpha^A]M^A & \to & M^A, & \text{si } \alpha^A \notin \mathsf{FVF}(M^A)
\end{array}
$$

Reglas permutativas:

$$
\begin{array}{lll}
\psi_L: & (M^{A\supset B} +_{\mathrm{L}} t)^{A\supset B} N^A & \to \quad (M^{A\supset B} N^A)^B +_{\mathrm{L}} t \\
\psi_R: & (s +_{\mathrm{R}} M^{A\supset B})^{A\supset B} N^A & \to \quad s +_{\mathrm{R}} (M^{A\supset B} N^A)^B \\
\phi_L: & N^B\langle v^A{:=}r, (M^{[\![r]\!]A} +_{\mathrm{L}} t)\rangle & \to \quad (N^B\langle v^A{:=}r, M\rangle)^{B\{v^A \leftarrow r^A\}} +_{\mathrm{L}} t \\
\phi_R: & N^B\langle v^A{:=}r, (s +_{\mathrm{R}} M^{[\![r]\!]A})\rangle & \to \quad s +_{\mathrm{R}} (N^B\langle v^A{:=}r, M\rangle)^{B\{v^A \leftarrow r^A\}} \\
\chi_L: & [\beta^A](M^A +_{\mathrm{L}} t)^A & \to \quad ([\beta^A]M^A)^\perp +_{\mathrm{L}} t \\
\chi_R: & [\beta^B](s +_{\mathrm{R}} N^B)^B & \to \quad s +_{\mathrm{R}} ([\beta^B]N^B)^\perp \\
\iota_L: & \mu\alpha^A.(M^\perp +_{\mathrm{L}} t)^\perp & \to \quad (\mu\alpha^A.M^\perp) +_{\mathrm{L}} t, \qquad \text{si } \alpha^A \notin \mathsf{FVF}(t) \\
\iota_R: & \mu\alpha^A.(s +_{\mathrm{R}} N^\perp)^\perp & \to \quad s +_{\mathrm{R}} (\mu\alpha^A.N^\perp), \qquad \text{si } \alpha^A \notin \mathsf{FVF}(s)
\end{array}
$$

Los resultados principales obtenidos sobre HLP y λ^{LP} son los siguientes.

- Correspondencia entre LP y HLP.
  - Si $\rhd_{\mathsf{LP}} \Gamma \vdash A$, entonces $\exists s$. t.q. $\rhd_{\mathsf{HLP}} \cdot; \underline{\Gamma}; \cdot \vdash \underline{A} \,|\, s$.
  - Si $\rhd_{\mathsf{HLP}} \cdot; \cdot; \cdot \vdash A \,|\, s$, entonces $\rhd_{\mathsf{LP}} \cdot \vdash A^\star$
- Preservación de tipos:
  - Si $\rhd_{\lambda^{\mathsf{LP}}} \Theta; \Gamma; \Delta \vdash M^A \,|\, s$ y $M \to N$, entonces $\rhd_{\lambda^{\mathsf{LP}}} \Theta; \Gamma; \Delta \vdash N^A \,|\, s'$ para un s' t.q. $\rhd_{\mathsf{HLP}} \Theta; \Gamma; \Delta \vdash s \equiv s' : A$.
- Confluencia de $\to$ para términos tipables.
- Normalización fuerte de $\to$ para términos tipables.

Estudiamos también algunas extensiones y variantes de HLP, por ejemplo la introducción de nuevas reglas permutativas y el tipo de datos de los números naturales.

A.5 Lógica Hipotética de Pruebas de Primer Orden

Esta sección (correspondiente al Capítulo 5 de esta tesis) extiende el análisis de la teoría de pruebas de HLP al caso de primer orden. Para esto, el primer paso es formular una presentación adecuada en Deducción Natural para FOLP, llamada FOHLP. Luego sigue un análisis de la normalización de derivaciones, probando la Normalización Fuerte de este proceso mediante una asignación de términos.

A.5.1 Lógica de Pruebas de Primer Orden

A continuación exponemos la lógica FOLP [AY11], incluyendo en nuestra presentación dos diferencias menores. La primera es que asumimos dados símbolos de función en el lenguaje de primer orden $\mathcal{L}$, ya que no se requiere un esfuerzo mayor para incorporarlos en nuestra presentación en Deducción Natural. La segunda es que, al igual que para LP, asumimos que las variables de prueba están decoradas con fórmulas. Es decir las variables toman la forma x^A en lugar de x.

Definition A.5.1 (Lenguaje de FOLP) Dado $\mathcal{L}$, el lenguaje de FOLP, $\mathcal{L}_{\mathsf{LP}}$, es la extensión de $\mathcal{L}$ que incluye:

- Variables de prueba $x_1, x_2, \ldots$
- Constantes de prueba $c_1, c_2, \ldots$
- Símbolos de función para operaciones sobre pruebas:
  - Los provenientes de LP: $+$ (binario), $\cdot$ (binario), $!$ (unario).
  - El operador $\mathsf{gen}_i()$ unario para cada variable individual i.

Los polinomios de prueba de LP proposicional pasan a llamarse "términos de prueba" [AY11].

Definition A.5.2 (Términos de prueba y fórmulas) El conjunto de términos de prueba, expresiones de primer orden y fórmulas de FOLP se define como:

$$\begin{array}{rcl} s,t & ::= & x^A \mid c \mid s \cdot t \mid !s \mid s + t \mid \mathsf{gen}_i(s) \\ E & ::= & i \mid f(E_1, \ldots, E_n) \\ A, B & ::= & P(E_1, \ldots, E_n) \mid \bot \mid A \supset B \mid [\![s]\!]_{\Xi} A \mid \forall i.A \end{array}$$

donde Ξ es un conjunto de variables individuales (o variables de individuos). Podemos abreviar $[\![s]\!]_{\emptyset} A$ como $[\![s]\!]A$.

Un término de prueba tiene una variable individual libre i solo si esta ocurre en la fórmula que decora una variable de prueba y no ocurre en una expresión de la forma $\mathsf{gen}_i(s)$. Las variables individuales que están libres en $[\![t]\!]_{\Xi} A$ son exactamente las contenidas en Ξ. Todas las otras variables individuales se asumen ligadas. Por ejemplo, en la fórmula $[\![(c^{\mathbf{A1a}}_{P(i),Q(j)})]\!]_{\{j\}}(P(i) \supset Q(j) \supset P(i))$, la variable j está libre, mientras que i está ligada. Asumimos la siguiente convención de variables: los nombres de todas las variables individuales ligadas son distintos entre sí, y distintos de los nombres de todas las variables de individuos libres. Al conjunto de variables de individuos libres en una fórmula A lo denotamos $\mathsf{FIV}(A)$.

Definition A.5.3 Los axiomas y reglas de inferencia de FOLP son los siguientes (los axiomas **A1** y **B3** se separan en sub-axiomas, uno por cada cláusula, con el fin de facilitar su tratamiento):

A1.	Axiomas de la lógica de primer orden en el lenguaje de FOLP	
A2.	$([\![t]\!]_{\Xi,i}A) \supset [\![t]\!]_{\Xi}A$,	si $i \notin \mathsf{FIV}(A)$
A3.	$([\![t]\!]_{\Xi}A) \supset [\![t]\!]_{\Xi,i}A$	
B1.	$([\![t]\!]_{\Xi}A) \supset A$	
B2.	$([\![s]\!]_{\Xi}(A \supset B)) \supset ([\![t]\!]_{\Xi}A) \supset [\![(s \cdot t)]\!]_{\Xi}B$	
B3a.	$([\![s]\!]_{\Xi}A) \supset [\![(s+t)]\!]_{\Xi}A$	
B3b.	$([\![t]\!]_{\Xi}A) \supset [\![(s+t)]\!]_{\Xi}A$	
B4.	$([\![t]\!]_{\Xi}A) \supset [\![!t]\!]_{\Xi}[\![t]\!]_{\Xi}A$	
B5.	$([\![t]\!]_{\Xi}A) \supset [\![\mathsf{gen}_i(t)]\!]_{\Xi}\forall i.A$,	si $i \notin \Xi$
MP.	$\vdash A \supset B$ y $\vdash A \;\Rightarrow \vdash B$	
Gen.	$\vdash A \;\Rightarrow \vdash \forall i.A$	
Nec.	A un axioma $\;\Rightarrow \vdash [\![c]\!]A$	

Los axiomas de la lógica de primer orden con los que trabajaremos son:

A1a.	$A \supset B \supset A$	
A1b.	$(A \supset B \supset C) \supset (A \supset B) \supset A \supset C$	
A1c.	$\neg\neg A \supset A$	
A1d.	$(\forall i.A) \supset A\{i \leftarrow E\}$	
A1e.	$(\forall i.(A \supset B)) \supset (\forall i.A) \supset \forall i.B$	
A1f.	$A \supset \forall i.A$,	si $i \notin \mathsf{FIV}(A)$

Remark A.5.4 Si bien todos los axiomas de LP son también axiomas en FOLP, sus nombres difieren en ambas presentaciones. Los axiomas de LP **A0**, **A1**, **A2**, **A2**, **A4**, **A5** tienen como contrapartida en primer orden a los axiomas **A1a,b,c**, **B1**, **B2**, **B4**, **B3a** y **B3b** respectivamente. Por otro lado, las reglas de inferencia **MP** y **Nec** en LP se corresponden con las reglas **MP** y **Nec** en FOLP.

Para enunciar el resultado de internalización en primer orden, fijaremos la siguiente notación: sea $\Gamma = A_1, \ldots A_n$, denotamos a un contexto de la forma $[\![u_1]\!]_{\Xi_1}A_1, \ldots, [\![u_n]\!]_{\Xi_n}A_n$ como $[\![\vec{u}]\!]_{\vec{\Xi}}\Gamma$.

Lema A.5.5 (Internalización) Si $[\![\vec{u}]\!]_{\vec{\Xi}}\Gamma \vdash D$ es derivable en FOLP, existe un término de prueba r tal que $[\![\vec{u}]\!]_{\vec{\Xi}}\Gamma \vdash [\![r]\!]_{\vec{\Xi}}D$ es derivable en FOLP.

Corolario A.5.1 Si $[\![\vec{u}]\!]_{\vec{\Xi}}\Gamma \vdash D$ es derivable en FOLP, existe un término de prueba $r^{\vec{\Xi},D}$ tal que $[\![\vec{u}]\!]_{\vec{\Xi}}\Gamma \vdash [\![r^{\vec{\Xi},D}]\!]_{\vec{\Xi}\cap\mathsf{FIV}(D)}D$ es derivable en FOLP.

A.5.2 Lógica Hipotética de Pruebas de Primer Orden

Definition A.5.6 (Testigos de prueba y fórmulas) El conjunto de expresiones de primer orden, fórmulas y testigos de prueba de FOHLP se define según la siguiente gramática:

$$
\begin{aligned}
r, s, t &::= x^A \mid v^A_\Xi \mid \lambda x^A.s \mid s \cdot t \\
&\mid\ !s \mid t\langle v^A_{\Xi'} := r, s\rangle \\
&\mid\ [\alpha^A]s \mid \mu\alpha^A.s \\
&\mid\ s + t \\
&\mid\ \mathsf{gen}_i(s) \mid \mathsf{ins}^E_i(s) \\
E_1, \ldots, E_n &::= i \mid f(E_1, \ldots, E_n) \\
A, B &::= P(E_1, \ldots, E_n) \mid \bot \mid A \supset B \mid [\![s]\!]_\Xi A \mid \forall i.A
\end{aligned}
$$

Incorporamos testigos adicionales para poder tratar con la introducción y eliminación del operador $\forall$. Estos nuevos testigos son de la forma $\mathsf{gen}_i(s)$ (*generalización*) e $\mathsf{ins}^E_i(s)$ (*instanciación*).

Los contextos y juicios en FOHLP se definen de manera análoga a los de FOHLP.

La definición del conjunto de variables individuales libres en una fórmula o testigo puede verse en la Sección 5.2 (Def. 5.2.3). De la misma, cabe destacar que las variables libres del testigo v^A_Ξ son exactamente las contenidas en Ξ, al igual que en la fórmula $[\![s]\!]_\Xi A$.

Trabajaremos módulo α-equivalencia sobre variables individuales (Def. 5.2.4), renombrando adecuadamente para mantener la siguiente convención: los nombres de todas las variables individuales ligadas son distintos entre sí, y distintos de los nombres de todas las variables de individuos libres, en todo testigo de prueba, fórmula, sentencia o prueba.

Definición A.5.7 (Axiomas y reglas de inferencia) Los axiomas y reglas de inferencia de FOHLP pueden verse en la Fig. A.6. Decimos que un juicio $\Theta; \Gamma; \Delta \vdash A \mid s$ es derivable si puede construirse una derivación del mismo utilizando los esquemas de inferencia de la Fig. A.6, y en ese caso escribimos $\rhd_{\mathsf{FOHLP}} \Theta; \Gamma; \Delta \vdash A \mid s$. Al igual que para HLP, escribimos $\mathcal{H} \vdash A \mid s$ para abreviar $\Theta; \Gamma; \Delta \vdash A \mid s$.

Notar que las únicas reglas que se modifican con respecto a HLP (Fig. A.5) son VarM, $\Box$I, $\Box$E, y las nuevas reglas $\forall$I y $\forall$E. Para beneficio del lector, incluimos todas las reglas en la Fig. 5.1. A continuación comentaremos sobre las nuevas reglas de introducción y eliminación de la modalidad.

- $\Box$I. La restricción $\mathsf{FIV}(\Theta) \cap \mathsf{FIV}(A) \subseteq \Xi$ es necesaria para evitar ligar variables individuales que se utilicen como libres en las premisas, al igual que la restricción de que $i \notin \mathsf{FIV}(\Theta, \Gamma, \Delta)$ en $\forall$I. Sin esta restricción, podríamos derivar juicios que no deberían ser derivables, como $v^{P(i)}_i; \cdot; \cdot \vdash [\![v^{P(i)}_i]\!]_\emptyset P(i) \mid !v^{P(i)}_i$, lo cual indicaría que $\forall i.P(i)$ es verdadera – e incluso válida – si asumimos que $P(i)$ es válida para un i dado. Esto implica que podríamos probar $[\![w^{P(i)}_i]\!]_i P(i) \supset \forall i.P(i)$ (ver Proposición 5.2.19 para los detalles sobre por qué $[\![v^{P(i)}_i]\!]_\emptyset P(i)$ implica $\forall i.P(i)$, en particular los axiomas **B5** y **B1**.

- $\Box$E. La restricción en $\Box$E previene que una prueba de una fórmula con variables individuales libres se utilice como prueba de una fórmula donde esas variables están ligadas. La conversa puede hacerse sin riesgos (al igual que una prueba de $\forall i.P(i)$ puede usarse para probar $P(i)$). Este es el motivo por el cual la inclusión está orientada en un solo sentido.

Las nuevas reglas de equivalencia de testigos pueden verse en la Fig. 5.2 y la Fig. 5.3, en la Sección 5.2.1.

$$\frac{}{\mathcal{H}, x^A \vdash A \,|\, x^A}\ \mathsf{Var}$$

$$\frac{\mathcal{H}, x^A \vdash B \,|\, s}{\mathcal{H} \vdash A \supset B \,|\, \lambda x^A.s}\ \supset\mathsf{I} \qquad \frac{\mathcal{H} \vdash A \supset B \,|\, s \qquad \mathcal{H} \vdash A \,|\, t}{\mathcal{H} \vdash B \,|\, s \cdot t}\ \supset\mathsf{E}$$

$$\frac{}{\mathcal{H}, v^A_\Xi \vdash A \,|\, v^A_\Xi}\ \mathsf{VarM}$$

$$\frac{\Theta; \cdot; \cdot \vdash A \,|\, s \quad \Theta; \cdot; \cdot \vdash\ s \equiv t : A \quad \mathsf{FIV}(\Theta) \cap \mathsf{FIV}(A) \subseteq \Xi}{\Theta; \Gamma; \Delta \vdash [\![t]\!]_\Xi A \,|\, !t}\ \Box\mathsf{I}$$

$$\frac{\mathcal{H} \vdash [\![r]\!]_\Xi A \,|\, s \quad \mathcal{H}, v^A_{\Xi'} \vdash C \,|\, t \quad \Xi \cap \mathsf{FIV}(A) \subseteq \Xi'}{\mathcal{H} \vdash C\{v^A_{\Xi'} \leftarrow r\} \,|\, t\langle v^A_{\Xi'} := r, s\rangle}\ \Box\mathsf{E}$$

$$\frac{\mathcal{H} \vdash A \,|\, s}{\mathcal{H} \vdash A \,|\, s + t}\ \mathsf{PlusL} \qquad \frac{\mathcal{H} \vdash A \,|\, t}{\mathcal{H} \vdash A \,|\, s + t}\ \mathsf{PlusR}$$

$$\frac{\mathcal{H}, \alpha^A \vdash A \,|\, s}{\mathcal{H}, \alpha^A \vdash \bot \,|\, [\alpha^A]s}\ \mathsf{Name} \qquad \frac{\mathcal{H}, \alpha^A \vdash \bot \,|\, s}{\mathcal{H} \vdash A \,|\, \mu\alpha^A.s}\ \mathsf{NAbs}$$

$$\frac{\mathcal{H} \vdash A \,|\, s \quad i \notin \mathsf{FIV}(\mathcal{H})}{\mathcal{H} \vdash \forall i.A \,|\, \mathsf{gen}_i(s)}\ \forall\mathsf{I} \qquad \frac{\mathcal{H} \vdash \forall i.A \,|\, s}{\mathcal{H} \vdash A\{i \leftarrow E\} \,|\, \mathsf{ins}^E_i(s)}\ \forall\mathsf{E}$$

Figura A.6: Axiomas y reglas de inferencia de FOHLP

Al igual que en HLP, si $\rhd_{\mathsf{FOHLP}} \Theta; \Gamma; \Delta \vdash s \equiv t : D$, tanto $\Theta; \Gamma; \Delta \vdash D \,|\, s$ como $\Theta; \Gamma; \Delta \vdash D \,|\, t$ son derivables.

La relación entre FOLP y FOHLP y las respectivas traducciones pueden verse en las Secciones 5.2.2 y 5.2.3.

A.5.3 Asignación de Términos

Presentamos a continuación el cálculo λ^{FOLP}, la asignación de términos para FOHLP.

Definition A.5.8 (Terms of λ^{FOLP}) Los términos de λ^{FOLP} se definen de acuerdo con la siguiente gramática:

$$
\begin{array}{rcl}
M, N & ::= & x^A \\
& | & v^A_{\Xi} \\
& | & (\lambda x^A.M^B)^{A \supset B} \\
& | & (M^{A \supset B} N^A)^B \\
& | & (!M^A)^{[\![s]\!]_{\Xi} A} \\
& | & (M^B \langle v^A_{\Xi'} := r, N^{[\![r]\!]_{\Xi} A} \rangle)^{B\{v^A_{\Xi'} \leftarrow r\}} \\
& | & ([\alpha^A] M^A)^{\bot} \\
& | & (\mu \alpha^A.M^{\bot})^A \\
& | & (M^A +_{\mathrm{L}} s)^A \\
& | & (s +_{\mathrm{R}} N^B)^B \\
& | & (\mathsf{gen}_i(M^A))^{\forall i.A} \\
& | & (\mathsf{ins}^E_i(M^{\forall i.A}))^{A\{i \leftarrow E\}}
\end{array}
$$

Los términos son testigos decorados con anotaciones que brindasn información adicional sobre las derivaciones. La relación entre los términos de λ^{FOLP} y las derivaciones en FOHLP es la misma que la relación entre los términos de λ^{LP} y las derivaciones en HLP. Al igual que en λ^{LP}, el término $(M^A +_{\mathrm{L}} s)^A$ codifica una prueba de A que concatena el testigo s a una prueba previa de A – codificada por M^A – usando PlusL. Análogamente, $(s +_{\mathrm{R}} M^A)^A$ codifica la prueba que resulta de concatenar s a una prueba de A por PlusR. También como en λ^{LP}, pero teniendo ahora en cuenta las variables individuales, los términos $(!v^A_{\Xi})^{[\![v^A_{\Xi}]\!]_{\Xi} A}$, $(!v^A_{\Xi})^{[\![(\lambda x^A.x^A) \cdot v^A_{\Xi}]\!]_{\Xi} A}$ y $(!v^A_{\Xi})^{[\![v^A_{\Xi}]\!]_{\Xi \cup \Xi'} A}$ codifican distintas derivaciones, en las cuales $\Box$I se usa de diferentes maneras para probar distintas fórmulas.

Nuevamente, nuestros términos no codifican las reglas de equivalencia utilizadas para derivar la segunda premisa de T-$\Box$I, ni los contextos utilizados en las derivaciones.

Las variables libres de validez, verdad y falsedad, al igual que las variables individuales libres de los términos se definen de manera análoga a las de los testigos, y las convenciones de notación para los testigos se extienden a términos de la manera esperada. Al igual que en λ^{LP}, las decoraciones pueden omitirse cuando no hay ambigüedad. Usaremos las letras M, N, L – con o sin superíndices – para referirnos a términos.

Las reglas de tipado pueden verse en la Fig. 5.4, en la Sección 5.3.

La reducción en λ^{FOLP} está dada por las siguientes reglas.

Reglas principales:

$$
\begin{array}{lll}
\beta: & (\lambda x^A.M^B)N^A & \to \; M^B\{x^A \leftarrow N^A\} \\
\mu: & [\beta^A]\mu\alpha^A.M^\perp & \to \; M^\perp\{\alpha^A \leftarrow \beta^A\} \\
\zeta: & (\mu\alpha^{A\supset B}.M^\perp)N^A & \to \; \mu\beta^B.M^\perp(\!|\, [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)N^A \,|\!) \\
\theta: & \mu\alpha^A.[\alpha^A]M^A & \to \; M^A \;\text{ si } \alpha^A \notin \mathsf{FVF}(M^A) \\
\gamma: & M^B\langle v^A_\Xi{:=}r, !N^A\rangle & \to \; M^B\{v^A_\Xi \leftarrow N^A, r\} \text{ si } \mathsf{FVT}(N^A) = \mathsf{FVF}(N^A) = \emptyset \\
\xi: & \mathsf{ins}^E_i(\mathsf{gen}_i(M^A)) & \to \; (M^A)\{i \leftarrow E\}
\end{array}
$$

Reglas permutativas:

$$
\begin{array}{lll}
\psi_L: & (M^{A\supset B} +_{\mathrm{L}} t)^{A\supset B} N^A & \to \; (M^{A\supset B} N^A)^B +_{\mathrm{L}} t \\
\psi_R: & (s +_{\mathrm{R}} M^{A\supset B})^{A\supset B} N^A & \to \; s +_{\mathrm{R}} (M^{A\supset B} N^A)^B \\
\chi_L: & [\beta^A](M^A +_{\mathrm{L}} t)^A & \to \; ([\beta^A]M^A)^\perp +_{\mathrm{L}} t \\
\chi_R: & [\beta^B](s +_{\mathrm{R}} N^B)^B & \to \; s +_{\mathrm{R}} ([\beta^B]N^B)^\perp \\
\phi_L: & M^B\langle v^A_{\Xi'}{:=}r, (N^{[\![r]\!]_{\Xi^A}} +_{\mathrm{L}} t)\rangle & \to \; (M^B\langle v^A_{\Xi'}{:=}r, N^{[\![r]\!]_{\Xi^A}}\rangle)^{B\{v^A_{\Xi'} \leftarrow r^A\}} +_{\mathrm{L}} t \\
\phi_R: & M^B\langle v^A_{\Xi'}{:=}r, (s +_{\mathrm{R}} N^{[\![r]\!]_{\Xi^A}})\rangle & \to \; s +_{\mathrm{R}} (M^B\langle v^A_{\Xi'}{:=}r, N^{[\![r]\!]_{\Xi^A}}\rangle)^{B\{v^A_{\Xi'} \leftarrow r^A\}} \\
\iota_L: & \mu\alpha^A.(M^\perp +_{\mathrm{L}} t)^\perp & \to \; (\mu\alpha^A.M^\perp) +_{\mathrm{L}} t, \text{ si } \alpha^A \notin \mathsf{FVF}(t) \\
\iota_R: & \mu\alpha^A.(s +_{\mathrm{R}} N^\perp)^\perp & \to \; s +_{\mathrm{R}} (\mu\alpha^A.N^\perp), \text{ si } \alpha^A \notin \mathsf{FVF}(s) \\
\epsilon_L: & \mathsf{ins}^E_i(M^{\forall i.A} +_{\mathrm{L}} t) & \to \; \mathsf{ins}^E_i(M^{\forall i.A}) +_{\mathrm{L}} t \\
\epsilon_R: & \mathsf{ins}^E_i(s +_{\mathrm{R}} M^{\forall i.A}) & \to \; s +_{\mathrm{R}} \mathsf{ins}^E_i(M^{\forall i.A})
\end{array}
$$

Las reglas γ, ϕ_L y ϕ_R son versiones modificadas de las reglas correspondientes en λ^{LP}. Las reglas ξ, ϵ_L y ϵ_R son exclusivas de esta extensión. Todas las otras reglas son idénticas a las de λ^{LP}.

Notar que las nuevas reglas no introducen pares críticos. Esto significa que FOHLP tiene los mismos pares críticos que HLP, y todos ellos pueden cerrarse como se muestra en la Sección 4.6.

Los resultados principales obtenidos sobre FOHLP y λ^{FOLP} son los siguientes.

- Correspondencia entre FOLP y FOHLP.
  - Si $\rhd_{\mathsf{FOLP}} \Gamma \vdash A$, entonces $\exists s.$ t.q. $\rhd_{\mathsf{FOHLP}} \cdot; \underline{\Gamma}; \cdot \vdash \underline{A} \,|\, s$.
  - Si $\rhd_{\mathsf{FOHLP}} \cdot; \cdot; \cdot \vdash A \,|\, s$, entonces $\rhd_{\mathsf{FOLP}} \cdot \vdash A^\star$
- Preservación de tipos:
  - Si $\rhd_{\lambda^{\mathsf{FOLP}}} \Theta; \Gamma; \Delta \vdash M^A \,|\, s$ y $M \to N$, entonces $\rhd_{\lambda^{\mathsf{FOLP}}} \Theta; \Gamma; \Delta \vdash N^A \,|\, s'$ para un s' t.q. $\rhd_{\mathsf{FOHLP}} \Theta; \Gamma; \Delta \vdash s \equiv s' : A$.
- Confluencia de $\to$ para términos tipables.
- Normalización fuerte de $\to$ para términos tipables.

A.6 Conclusiones y Trabajo Futuro

Esta tesis abordó dos temas en reescritura.

En la primera parte presentamos un sistema de combinadores (CL_P) para un cálculo lambda (λP) en el cual las funciones abstraen una noción generalizada de patrón que incluye aplicaciones y abstracciones. Hemos probado que la noción asociada de *pattern-matching* puede ser capturada por medio de reescritura de primer orden, en el marco de un sistema de reescritura de términos (TRS). Para esto, enfrentamos dos desafíos:

1. Simular el proceso de matching de λP por medio de combinadores; y
2. Computar la sustitución resultante, también por medio de combinadores, los cuales deben descomponer aplicaciones para alcanzar el resultado esperado.

Encontramos además condiciones suficientes para evitar patrones mal formados que pueden romper la confluencia, y presentamos caracterizaciones sintácticas de las restricciones correspondientes. Estas restricciones pueden verificarse eficientemente dado un término o un patrón. Presentamos también un sistema de tipos simple, para el cual se probó la normalización de términos tipables. Finalmente, abordamos algunas extensiones, entre las cuales se destaca $CL_\star$. En contraste con CL_P, donde las reglas de reducción son finitas pese a estar representadas mediante un conjunto finito de esquemas, $CL_\star$ es un TRS finito. Esto se logró codificando el matching dentro del propio cálculo, al precio de obtener una sintaxis más complicada para los combinadores.

En la segunda parte de esta tesis desarrollamos una presentación de LP basada en razonamiento hipotético, llamada HLP. Este trabajo se nutrió, por un lado, de la Deducción Natural Clásica de Parigot, y por otro, de presentaciones previas con razonamiento hipotético del fragmento intuicionista de LP [AB07, BF12, BB10]. El resultado es un sistema de Deducción Natural capaz de probar los teoremas de LP. Se propuso una asignación de términos, la cual se utilizó para realizar un análisis detallado de la normalización de derivaciones en HLP: las derivaciones se representan como términos, y los pasos de normalización de derivaciones se codifican como pasos de reducción sobre términos. La extensión del Cálculo Lambda resultante, el cálculo λ^{LP}, satisface Preservación de Tipos, Normalización Fuerte y Confluencia. Se introdujo también una noción de equivalencia de pruebas para abordar el problema de la falla de Preservación de Tipos cuando el operador responsable de la internalización de pruebas (y su regla de introducción $\Box$I) causa modificaciones en los testigos ante un paso de reducción. También fue necesario reformular las reglas de inferencia relacionadas con el operador "+" – el cual introduce indeterminismo en las pruebas – para poder expresar todos los teoremas de LP, ya que en LP existen teoremas con variables libres: las mismas están permitidas siempre y cuando se ubiquen del lado del "+" que no se utiliza en la derivación. Finalmente desarrollamos una extensión de primer orden de HLP, llamada FOHLP, y probamos la correspondencia entre FOHLP y la lógica FOLP de Artemov e Yavorskaya. Presentamos una asignación de términos para λ^{FOLP}, incluyendo contracciones principales y conversiones permutativas, y probamos las propiedades fundamentales de Confluencia de la reducción, Preservación de Tipos y Normalización Fuerte de términos tipables.

Surgieron algunas dificultades al tratar con las variables ligadas, en particular las variables de individuos ligadas por el operador "$\Box$" de primer orden. De hecho, las reglas $\Box\mathsf{I}$ y $\Box\mathsf{E}$ resultaron especialmente elusivas.

Como posibles *líneas de trabajo futuro*, mencionamos las siguientes.

Combinadores para patrones dinámicos. El *Pure Pattern Calculus* o PPC [Jay06, JK05, JK09] es un cálculo de patrones que permite la creación de patrones en tiempo de ejecución. Por ejemplo, sea M el siguiente término de PPC:

$$x \hookrightarrow_x (x \hookrightarrow \texttt{true} | y \hookrightarrow \texttt{false})$$

El símbolo "$\hookrightarrow$" permite construir abstracciones, siendo su primer argumento el patrón, y el segundo el cuerpo. Un conjunto de variables denotado como subíndice, como x en "$\hookrightarrow_x$", indica qué variables del patrón se utilizarán para el matching (x en este ejemplo) y cuáles se consideran variables libres (las mismas deberán sustituirse desde "afuera"). El orden entre los patrones es importante, ya que la reducción procederá de acuerdo al primer matching exitoso. El término $M(\texttt{Succ 0})(\texttt{Succ 0})$ reduce en dos pasos a `true`. Notar que, luego del primer paso, $(\texttt{Succ 0})$ se transforma en un nuevo patrón, obteniendo:

$$\texttt{Succ 0} \hookrightarrow \texttt{true} | y \hookrightarrow \texttt{false}$$

De manera similar, $M(\texttt{Succ 0})\texttt{0}$ reduce a `false`. Resultaría interesante obtener una representación combinatoria de PPC.

Unificación de alto oren de λP mediante combinadores. Estamos interesados en estudiar la unificacíon con respecto a la igualdad de CL_P [Dou93]. La formulación de un algoritmo de unificación de alto orden resulta dificultosa (aun si nos restringimos a términos del Cálculo Lambda simplemente tipado), debido a la presencia de variables ligadas. Por este motivo, [Dou93] enfatiza la importancia de considerar la conversión a Lógica Combinatoria como paso intermedio, eliminando así las variables ligadas. De este modo la formulación del algoritmo, según [Dou93], se simplifica considerablemente. Es importante notar que la ausencia de variables ligadas es el único requiremiento para que un cálculo sirva para este propósito. Nuestro sistema de combinadores parece ser un lenguaje adecuado, debido a que sus propiedades extienden a las clásicas.

Detección de fallas de matching mediante chequeo de tipos. Resulta de interés el estudio de posibles sistemas de tipos para CL_P, en particular sistemas de tipos dependientes en los cuales los patrones jueguen un rol en la detección de fallas de matching. Las reglas deberían ser elegidas cuidadosamente para considerar el matching aún no decidido. Por ejemplo, deberían poder tipar correctamente el término $S_P M N P^\sigma$, teniendo en cuenta que la forma normal de NP^σ no puede conocerse de antemano.

LP mediante Teoría de Tipos Modal Contextual. Típicamente, las reglas de inferencia mantienen el invariante de que todas las variables libres están declaradas en las hipótesis. La regla PlusL:

$$\frac{\Theta; \Gamma; \Delta \vdash A \mid s}{\Theta; \Gamma; \Delta \vdash A \mid s + t}\ \mathsf{PlusL}$$

falla en este sentido. Si bien argumentamos que esto es necesario para probar todos los teoremas de LP en HLP, parece razonable explorar modalidades dependientes [NPP08] al posible precio de capturar un subconjunto de los teoremas de LP(por ejemplo, reemplazando el axioma **A4** por $[\![s]\!]A \supset [\![t]\!]B \supset [\![s+t]\!]A$, y de manera análoga para **A5**). Como ya se mencionó, la modalidad $\Box A$ se reemplaza por $[\Gamma]A$, que se puede leer informalmente como $\Box(\Gamma \supset A)$. Esto es, la validez de A depende de la verdad de las hipótesis en Γ. Tres ejemplos de reglas de inferencia de la *Lógica Modal Contextual* [NPP08] resultante son:

$$\frac{\Theta; \Gamma_1 \vdash A}{\Theta; \Gamma_2 \vdash [\Gamma_1]A}\ \Box\mathsf{I} \qquad \frac{\Theta; \Gamma_1 \vdash [\Gamma_2]A \quad \Theta, v :: A[\Gamma_2]; \Gamma_1 \vdash C}{\Theta; \Gamma_1 \vdash C}\ \Box\mathsf{E}^v$$

$$\frac{\Theta_1, v :: A[\Gamma_1], \Theta_2; \Gamma_2 \vdash \Gamma_1}{\Theta_1, v :: A[\Gamma_1], \Theta_2; \Gamma_2 \vdash A}\ \mathsf{mvar}$$

En lugar de probar **A4** ($[\![t]\!]A \supset [\![s+t]\!]A$), probaríamos una fórmula de la forma:

$$[\![\Gamma, \Delta.s]\!]A \supset [\![\Gamma, \Delta.s+t]\!]A$$

Notar que t solo podría tener variables libres en Γ, Δ.

Marco Lógico Basado en FOHLP. Consideramos que sería interesante desarrollar un marco lógico (*logical framework*) basado en HLP de primer orden. Creemos que Beluga [Pie10] puede proveer inspiración relevante para este propósito, ya que Beluga mismo está basado en [NPP08], y permite a la vez el uso de múltiples contextos y de tipos dependientes.

Cuantificación sobre variables de pruebas. Un sistema basado en HLP con la habilidad de cuantificar sobre variables de pruebas, en el espíritu de [Fit05b], merece ser estudiado. Esto incluiría el análisis del polimorfismo inducido sobre la asignación de términos.

Técnicas de inferencia de tipos para realización. Creemos que una mirada fresca a la realización de S4 en el marco de HLP podría ser un camino interesante para explorar. Cabe notar que este es un problema no trivial ante la presencia de reglas de inferencia con polaridades mixtas como $\supset$ E, de ahí el motivo por el cual la primera prueba [Art95, Art01] se basó en una presentación de LP como cálculo de secuentes sin cortes. De hecho, todas las pruebas de realización conocidas (por la autora de esta tesis) se basan en presentaciones donde ocurrencias relacionadas[2] de un $\Box$ *no* ocurren simultáneamente en posiciones positivas y negativas. Creemos

[2] Ver noción de "familia" en [Art01].

que puede ser interesante utilizar la tecnología ya existente para inferencia de tipos, pero inferir, en lugar de tipos, las decoraciones de las cajas. En el proceso podrían surgir relaciones con la unificación de alto orden.

Appendix B

Additional proofs and results

B.1 Combinatory Logics for Lambda Calculi with Patterns

Proof of Lemma 3.7.7

Proof.- By induction on P, using the syntax for matching-safe patterns defined in Section 3.6.1. We will not consider constructors in this proof.

- If $P = x$: $[P] \sim [P] = \star \sim \star \to_{W_\star} I$
- If $P = K_{P_1}$: $[P] \sim [P] = \mathsf{K}\langle [P_1] \rangle \sim \mathsf{K}\langle [P_1] \rangle \to_{W_\star} P_1 \sim P_1 \twoheadrightarrow_{W_\star (IH)} I$
- If $P = S_{P_1}$, the proof is analogous to the previous case.
- If $P = K_{P_1 P_2}$:

$$\begin{array}{ll} [P] \sim [P] & = \\ \mathsf{K}\langle [P_1] \rangle [P_2] \sim (\mathsf{K}\langle [P_1] \rangle [P_2]) & \to_{W_\star} \\ [P_1] \sim [P_1]([P_2] \sim [P_2]) & \twoheadrightarrow_{W_\star (IH)} \\ I([P_2] \sim [P_2]) & \twoheadrightarrow_{W_\star (IH)} \\ II & \twoheadrightarrow_{W_\star} \quad I. \end{array}$$

- If $P = \mathsf{S}\langle P_1 \rangle P_2$, the proof is analogous to the previous case.
- If $P = \mathsf{S}\langle P_1 \rangle P_2 P_3$, the proof is also analogous, with one more comparison to take into account.
- If $P = \Pi^1_{QR}$:

$$\begin{array}{ll} [P] \sim [P] & = \\ \mathsf{S}\langle [QR] \rangle (\mathsf{K}\langle \star \rangle \Pi^1)(\mathsf{S}\langle \star \rangle \mathsf{K}\langle \star \rangle \star) \sim \mathsf{S}\langle [QR] \rangle (\mathsf{K}\langle \star \rangle \Pi^1)(\mathsf{S}\langle \star \rangle \mathsf{K}\langle \star \rangle \star) & \to_{W_\star} \\ [QR] \sim [QR](\mathsf{K}\langle \star \rangle \Pi^1 \sim \mathsf{K}\langle \star \rangle \Pi^1)((\mathsf{S}\langle \star \rangle \mathsf{K}\langle \star \rangle \star) \sim (\mathsf{S}\langle \star \rangle \mathsf{K}\langle \star \rangle \star)) & \twoheadrightarrow_{W_\star (IH)} \\ I(\mathsf{K}\langle \star \rangle \Pi^1 \sim \mathsf{K}\langle \star \rangle \Pi^1)((\mathsf{S}\langle \star \rangle \mathsf{K}\langle \star \rangle \star) \sim (\mathsf{S}\langle \star \rangle \mathsf{K}\langle \star \rangle \star)) & \twoheadrightarrow_{W_\star} \quad I. \end{array}$$

- If $P = \Pi^2_{QR}$, the proof is analogous to the previous case.

□

Proof of Lemma 3.7.8

Proof.- By induction on P, using the syntax for matching-safe patterns defined in Section 3.6.1, as well as the result of Lemma 3.7.7. We will not consider constructors in this proof.

- If $P = x$:
  1. $\mathsf{m}\langle[P]\rangle[P] = \mathsf{m}\langle[x]\rangle[x] = \mathsf{m}\langle\star\rangle\star \to_{W_\star} I$.
  2. $\mathsf{m}\langle[P]\rangle(P^\sigma)^\star = \mathsf{m}\langle[x]\rangle(x^\sigma)^\star = \mathsf{m}\langle\star\rangle(x^\sigma)^\star \to_{W_\star} I$.
- If $P = K_Q$:
  1. $\mathsf{m}\langle[P]\rangle[P] = \mathsf{m}\langle[K_Q]\rangle[K_Q] = \mathsf{m}\langle\mathsf{K}\langle[Q]\rangle\rangle\mathsf{K}\langle[Q]\rangle \to_{W_\star} [Q]\sim[Q] \overset{L.\ 3.7.7}{\twoheadrightarrow_{W_\star}} I$.
  2. $\mathsf{m}\langle[P]\rangle(P^\sigma)^\star = \mathsf{m}\langle[K_Q]\rangle((K_Q)^\sigma)^\star = \mathsf{m}\langle[K_Q]\rangle(K_Q)^\star = \mathsf{m}\langle[K_Q]\rangle[K_Q] \twoheadrightarrow_{W_\star} I$, as we have seen.
- If $P = S_Q$, the proof is analogous to the previous case.
- If $P = K_Q P_1$:
  1.
  $$\begin{array}{ll}
  \mathsf{m}\langle[P]\rangle[P] & = \\
  \mathsf{m}\langle[K_Q P_1]\rangle[K_Q P_1] & = \\
  \mathsf{m}\langle\mathsf{K}\langle[Q]\rangle[P_1]\rangle(\mathsf{K}\langle[Q]\rangle[P_1]) & \to_{W_\star} \\
  [Q]\sim[Q](\mathsf{m}\langle[P_1]\rangle[P_1]) & \overset{L.\ 3.7.7}{\twoheadrightarrow_{W_\star}} \\
  I(\mathsf{m}\langle[P_1]\rangle[P_1]) & \twoheadrightarrow_{IH(i)} \\
  II & \twoheadrightarrow_{W_\star} \quad I.
  \end{array}$$
  2.
  $$\begin{array}{ll}
  \mathsf{m}\langle[P]\rangle(P^\sigma)^\star & = \\
  \mathsf{m}\langle[K_Q P_1]\rangle((K_Q)^\sigma P_1^\sigma)^\star & = \\
  \mathsf{m}\langle[K_Q][P_1]\rangle((K_Q)^\star(P_1^\sigma)^\star) & = \\
  \mathsf{m}\langle\mathsf{K}\langle[Q]\rangle[P_1]\rangle(\mathsf{K}\langle[Q]\rangle(P_1^\sigma)^\star) & \to_{W_\star} \\
  [Q]\sim[Q](\mathsf{m}\langle[P_1]\rangle(P_1^\sigma)^\star) & \overset{L.\ 3.7.7}{\twoheadrightarrow_{W_\star}} \\
  I(\mathsf{m}\langle[P_1]\rangle(P_1^\sigma)^\star) & \twoheadrightarrow_{IH(ii)} \\
  II & \twoheadrightarrow_{W_\star} \quad I.
  \end{array}$$
- If $P = S_Q P_1$ or $P = S_Q P_1 P_2$, the proof is analogous to the previous case.
- If $P = \Pi^1_{QR}$:
  1.
  $$\begin{array}{ll}
  \mathsf{m}\langle[P]\rangle[P] & = \\
  \mathsf{m}\langle[\Pi^1_{QR}]\rangle[\Pi^1_{QR}] & = \\
  \mathsf{m}\langle\mathsf{S}\langle[QR]\rangle(\mathsf{K}\langle\star\rangle\Pi^1)(\mathsf{S}\langle\star\rangle\mathsf{K}\langle\star\rangle\star)\rangle(\mathsf{S}\langle[QR]\rangle(\mathsf{K}\langle\star\rangle\Pi^1)(\mathsf{S}\langle\star\rangle\mathsf{K}\langle\star\rangle\star)) & \to_{W_\star} \\
  [QR]\sim[QR](\mathsf{m}\langle\mathsf{K}\langle\star\rangle\Pi^1\rangle(\mathsf{K}\langle\star\rangle\Pi^1))(\mathsf{m}\langle(\mathsf{S}\langle\star\rangle\mathsf{K}\langle\star\rangle\star)\rangle((\mathsf{S}\langle\star\rangle\mathsf{K}\langle\star\rangle\star))) & \overset{L.\ 3.7.7}{\twoheadrightarrow_{W_\star}} \\
  I(\mathsf{m}\langle\mathsf{K}\langle\star\rangle\Pi^1\rangle(\mathsf{K}\langle\star\rangle\Pi^1))(\mathsf{m}\langle(\mathsf{S}\langle\star\rangle\mathsf{K}\langle\star\rangle\star)\rangle((\mathsf{S}\langle\star\rangle\mathsf{K}\langle\star\rangle\star))) & \twoheadrightarrow_{W_\star} \\
  \mathsf{m}\langle\mathsf{K}\langle\star\rangle\Pi^1\rangle(\mathsf{K}\langle\star\rangle\Pi^1)(\mathsf{m}\langle(\mathsf{S}\langle\star\rangle\mathsf{K}\langle\star\rangle\star)\rangle((\mathsf{S}\langle\star\rangle\mathsf{K}\langle\star\rangle\star))) & \to_{W_\star} \\
  \star\sim\star(\mathsf{m}\langle\Pi^1\rangle\Pi^1)(\mathsf{m}\langle(\mathsf{S}\langle\star\rangle\mathsf{K}\langle\star\rangle\star)\rangle((\mathsf{S}\langle\star\rangle\mathsf{K}\langle\star\rangle\star))) & \to_{W_\star} \\
  I(\mathsf{m}\langle\Pi^1\rangle\Pi^1)(\mathsf{m}\langle(\mathsf{S}\langle\star\rangle\mathsf{K}\langle\star\rangle\star)\rangle((\mathsf{S}\langle\star\rangle\mathsf{K}\langle\star\rangle\star))) & \twoheadrightarrow_{W_\star} \quad I.
  \end{array}$$
  2. $\mathsf{m}\langle[P]\rangle(P^\sigma)^\star = \mathsf{m}\langle[\Pi^1_{QR}]\rangle(\Pi^1_{QR})^\star = \mathsf{m}\langle[\Pi^1_{QR}]\rangle[\Pi^1_{QR}] \twoheadrightarrow_{W_\star} I$ as we have just seen.

- If $P = \Pi^2_{QR}$, the proof is analogous to the previous case.

□

Proof of Lemma 3.7.9

Proof.- Since PQ is a matching-safe application pattern, there are only three possibilities.

- $PQ = K_{P'}Q$:
  1. $$\begin{array}{ll} \Pi^1(((PQ)^\sigma)^\star) & = \\ \Pi^1((K_{P'})^\star(Q^\sigma)^\star) & = \\ \Pi^1(\mathsf{K}\langle[P']\rangle(Q^\sigma)^\star) & \to_{W_\star} \\ \mathsf{K}\langle[P']\rangle = (K_{P'})^\star = ((K_{P'})^\sigma)^\star & = \quad (P^\sigma)^\star. \end{array}$$
  2. $\Pi^2(((PQ)^\sigma)^\star) = \Pi^2(\mathsf{K}\langle[P']\rangle(Q^\sigma)^\star) \to_{W_\star} (Q^\sigma)^\star$
- $PQ = S_{P'}Q$: the proof is analogous to the previous case, replacing K by S.
- $PQ = S_{P'}RQ$:
  1. $$\begin{array}{ll} \Pi^1(((PQ)^\sigma)^\star) & = \\ \Pi^1((S_{P'}R^\sigma Q^\sigma)^\star) & = \\ \Pi^1(\mathsf{S}\langle[P']\rangle(R^\sigma)^\star(Q^\sigma)^\star) & \to_{W_\star} \\ \mathsf{S}\langle[P']\rangle(R^\sigma)^\star & = \\ (S_{P'}R^\sigma)^\star & = \quad (P^\sigma)^\star. \end{array}$$
  2. $\Pi^2(((PQ)^\sigma)^\star) = \Pi^2(\mathsf{S}\langle[P']\rangle(R^\sigma)^\star(Q^\sigma)^\star) \to_{W_\star} (Q^\sigma)^\star$

□

Proof of Proposition 3.7.10

Proof.- By induction on the reduction $M \to_{W_P} N$. There are 6 possibilities:

- $M = K_P N P^\sigma$:

$$\begin{array}{ll} M^\star & = \\ (K_P N P^\sigma)^\star & = \\ \mathsf{K}\langle[P]\rangle N^\star(P^\sigma)^\star & \to_{W_\star} \\ \mathsf{m}\langle[P]\rangle(P^\sigma)^\star K N^\star(P^\sigma)^\star & \overset{L.\ 3.7.8}{\twoheadrightarrow_{W_\star}} \\ I K N^\star(P^\sigma)^\star & \twoheadrightarrow_{W_\star} \\ K N^\star(P^\sigma)^\star & \to_{W_\star} \quad N^\star. \end{array}$$

- $M = S_P M_1 M_2 P^\sigma$ and $N = M_1 P^\sigma(M_2 P^\sigma)$:

$$\begin{array}{ll} M^\star & = \\ \mathsf{S}\langle[P]\rangle M_1^\star M_2^\star(P^\sigma)^\star & \to_{W_\star} \\ \mathsf{m}\langle[P]\rangle(P^\sigma)^\star S M_1^\star M_2^\star(P^\sigma)^\star & \overset{L.\ 3.7.8}{\twoheadrightarrow_{W_\star}} \\ I S M_1^\star M_2^\star(P^\sigma)^\star & \twoheadrightarrow_{W_\star} \\ S M_1^\star M_2^\star(P^\sigma)^\star & \to_{W_\star} \\ M_1^\star(P^\sigma)^\star(M_2^\star(P^\sigma)^\star) & = \quad N^\star. \end{array}$$

- $M = \Pi^1_{PQ}(P^\sigma Q^\sigma)$ and $N = P^\sigma$:

$$
\begin{array}{ll}
M^\star & = \\
\mathsf{S}\langle[PQ]\rangle(\mathsf{K}\langle\star\rangle\Pi^1)(\mathsf{S}\langle\star\rangle\mathsf{K}\langle\star\rangle\star)((PQ)^\sigma)^\star & \to_{W_\star} \\
\mathsf{m}\langle[PQ]\rangle((PQ)^\sigma)^\star S(\mathsf{K}\langle\star\rangle\Pi^1)(\mathsf{S}\langle\star\rangle\mathsf{K}\langle\star\rangle\star)((PQ)^\sigma)^\star & \overset{L.\ 3.7.8}{\twoheadrightarrow_{W_\star}} \\
IS(\mathsf{K}\langle\star\rangle\Pi^1)(\mathsf{S}\langle\star\rangle\mathsf{K}\langle\star\rangle\star)((PQ)^\sigma)^\star & \twoheadrightarrow_{W_\star} \\
\mathsf{K}\langle\star\rangle\Pi^1((PQ)^\sigma)^\star((\mathsf{S}\langle\star\rangle\mathsf{K}\langle\star\rangle\star)((PQ)^\sigma)^\star) & \to_{W_\star} \\
\mathsf{m}\langle\star\rangle((PQ)^\sigma)^\star K\Pi^1((PQ)^\sigma)^\star((\mathsf{S}\langle\star\rangle\mathsf{K}\langle\star\rangle\star)((PQ)^\sigma)^\star) & \to_{W_\star} \\
IK\Pi^1((PQ)^\sigma)^\star((\mathsf{S}\langle\star\rangle\mathsf{K}\langle\star\rangle\star)((PQ)^\sigma)^\star) & \twoheadrightarrow_{W_\star} \\
\Pi^1((\mathsf{S}\langle\star\rangle\mathsf{K}\langle\star\rangle\star)((PQ)^\sigma)^\star) & \twoheadrightarrow_{W_\star} \\
\Pi^1((PQ)^\sigma)^\star & \overset{L.\ 3.7.9}{\to_{W_\star}} \\
(P^\sigma)^\star & = \quad N^\star.
\end{array}
$$

- $M = \Pi^2_{PQ}(P^\sigma Q^\sigma)$ and $N = Q^\sigma$: the proof is analogous to the previous case.
- $M = M_1M_2$, $N = N_1M_2$ and $M_1 \to_{W_P} N_1$: $M^\star = M_1^\star M_2^\star \twoheadrightarrow_{W_\star(IH)} N_1^\star M_2^\star = N^\star$.
- $M = M_1M_2$, $N = M_1N_2$ and $M_2 \to_{W_P} N_2$: the proof is analogous to the previous case.

□

Definition B.1.1 A more efficient translation from λP to CL_P

$$
\begin{array}{lll}
1)\ \lambda^* x.x & \triangleq SKK & \\
2)\ \lambda^* P.M & \triangleq K_P M, & \text{if } \mathsf{FV}(P) \cap \mathsf{FV}(M) = \emptyset \\
3)\ \lambda^* PQ.x & \triangleq S_{PQ}(K\lambda^0 P.x)\Pi^1_{PQ}, & \text{if } x \in (\mathsf{FV}(P) \backslash \mathsf{FV}(Q)) \\
4)\ \lambda^* PQ.x & \triangleq S_{PQ}(K\lambda^0 Q.x)\Pi^2_{PQ}, & \text{if } x \in \mathsf{FV}(Q) \\
5)\ \lambda^* P.MN & \triangleq S_P(\lambda^0 P.M)(\lambda^0 P.N), & \text{if } \mathsf{FV}(P) \cap \mathsf{FV}(MN) \neq \emptyset
\end{array}
$$

with λ^0 defined as follows:

$$
\begin{array}{lll}
1)\ \lambda^0 x.x & \triangleq SKK & \\
2)\ \lambda^0 P.M & \triangleq KM, & \text{if } \mathsf{FV}(P) \cap \mathsf{FV}(M) = \emptyset \\
3)\ \lambda^0 PQ.x & \triangleq S(K\lambda^0 P.x)\Pi^1_{PQ}, & \text{if } x \in (\mathsf{FV}(P) \backslash \mathsf{FV}(Q)) \\
4)\ \lambda^0 PQ.x & \triangleq S(K\lambda^0 Q.x)\Pi^2_{PQ}, & \text{if } x \in \mathsf{FV}(Q) \\
5)\ \lambda^0 P.MN & \triangleq S(\lambda^0 P.M)(\lambda^0 P.N), & \text{if } \mathsf{FV}(P) \cap \mathsf{FV}(MN) \neq \emptyset
\end{array}
$$

The definition of $-_{CL}$ remains the same, using the new definition of λ^*.

The idea behind this optimized translation is roughly as follows: the abstraction translates into a term which will begin with a combinator (S or K) with a subindex corresponding to the pattern that must be matched by the argument. Once the argument is passed to this term and the resulting redex is reduced, the argument needs not be tested against the original pattern anymore. Thus, λ^0 simply removes the original pattern from the remaining appearances of K and S (note that the projectors' subindices cannot be removed, otherwise the result of the translation would not be a well-formed CL_P-term).

The following results hold for the new definition of λ^*:

Lemma B.1.2 $(\lambda^0 P.M)P^\sigma \twoheadrightarrow_{W_P} M^\sigma$ for every CL_P-term M, CL_P pattern P and CL_P substitution σ with
$dom(\sigma) \subseteq \mathsf{FV}(P)$.

Proof.- The proof is similar to that of Lemma 3.3.14, except that the induction is done using the definition of λ^0 rather than λ^*. □

Proposition B.1.3 The results obtained from the original formulation of λ^* are still valid in the new formulation.

Proof.- Lemma 3.3.8 still holds with a similar proof (the definition of λ^0 needs to be analyzed, but this analysis is analogous to that of λ^*).

To prove Lemma 3.3.9 we use that for every CL_P-term M, and pattern P, $|\lambda^0 P.M| > |M|$ (this can be proved with a simple induction on M using the definition of $\lambda^0 P.M$). We also use Lemma 3.3.8. The proof is similar to the original proof (see Appendix B.1), with some changes in the fifth case.

Now we have that $\lambda^* P_{CL}.(N^{\sigma'})_{CL}(T^{\sigma'})_{CL} = S_{P_{CL}}(\lambda^0 P_{CL}.(N^{\sigma'})_{CL})(\lambda^0 P_{CL}.(T^{\sigma'})_{CL})$, where $\lambda^0 P_{CL}.(N^{\sigma'})_{CL} =_{\mathrm{IH}} \lambda^0 P_{CL}.N_{CL}^{\sigma'_{CL}} = (\lambda^0 P_{CL}.N_{CL})^{\sigma'_{CL}}$. This requires proving that $\lambda^0 P.(M^\sigma) = (\lambda^0 P.M)^\sigma$, but the proof is straightforward. The same reasoning can be applied to $\lambda^0 P_{CL}.T_{CL}^{\sigma'}$ to prove that it is equal to $(\lambda^0 P_{CL}.T_{CL})^{\sigma'_{CL}}$. The rest of the proof is straightforward.

The proof of Lemmas 3.3.10 and 3.3.16, as well as Proposition 3.3.18 remain unchanged.

Lemma 3.3.11 is still a consequence of the previous lemmas.

The proof of Lemma 3.3.14 is done by using Lemma B.1.2 in a case-by-case analysis on the definition of λ^*.

Remarks 3.3.3 and 3.3.6 still hold for the same reasons as in the original formulation.

Proposition 3.3.20 also holds. The proof is similar to the original, except that whenever Lemma 3.3.14 is used, Lemma B.1.2 must be used instead. □

B.2 Substitution Lemmas for λ^{LP}

Proof of Lemma 4.4.12

Proof.- By induction on the derivation of $\Theta; \Gamma, y^A; \Delta \vdash M^B \,|\, s$. Keep in mind that, by Lemma 4.4.5, $\mathsf{w}(N^A) = t$. We consider the last step of the derivation.

- T-Var: in this case $s = x^B = M^B$ and $\Theta; \Gamma, y^A; \Delta \vdash M^B \,|\, s{=}\Theta; \Gamma', x^B; \Delta \vdash x^B \,|\, x^B$. There are two possibilities.
  - If $x^B = y^A$, then $\Gamma = \Gamma'$, $B = A$, $M\{y^A \leftarrow N\} = N$ and $s\{y^A \leftarrow t\}$. Then $\Theta; \Gamma; \Delta \vdash M\{y^A \leftarrow N\}^B \,|\, s\{y^A \leftarrow t\} = \Theta; \Gamma; \Delta \vdash N^A \,|\, t$, which is derivable by hypothesis.
  - If $x^B \neq y^A$, then $x^B \in \Gamma$ and $M\{y^A \leftarrow N\} = x^B = s\{y^A \leftarrow t\}$. Then, by T-Var, $\Theta; \Gamma; \Delta \vdash x^B \,|\, x^B$ is derivable.

- T-VarM: here $M = s = M\{y^A \leftarrow N\} = s\{y^A \leftarrow t\} = v^B$ and $v^B \in \Theta$. By T-VarM, $\Theta; \Gamma; \Delta \vdash v^B \,|\, v^B$ is derivable.
- T-$\supset$I: here $s = \lambda x^C.s'$, $B = C \supset D$ and $M = (\lambda x^C.M')^B$ for some term M', where $\Theta; \Gamma, y^A, x^C; \Delta \vdash M'^D \,|\, s'$ is derivable by hypothesis. Again, there are two possibilities:
  - If $x^C = y^A$, then $M\{y^A \leftarrow N\} = M$, $s\{y^A \leftarrow t\} = s$ and $\Gamma, y^A, x^C = \Gamma, x^C$. We can derive $\Theta; \Gamma; \Delta \vdash M^B \,|\, s$ from $\Theta; \Gamma, x^C; \Delta \vdash M'^D \,|\, s'$ by T-$\supset$I.
  - If $x^C \neq y^A$, then $M\{y^A \leftarrow N\} = \lambda x^C.(M'\{y^A \leftarrow N\})$ and $s\{y^A \leftarrow t\} = \lambda x^C.(s'\{y^A \leftarrow t\})$. By induction hypothesis, the judgment $\Theta; \Gamma, x^C; \Delta \vdash M'\{y^A \leftarrow N\}^D \,|\, s'\{y^A \leftarrow t\}$ is derivable. Thus, we can derive $\Theta; \Gamma; \Delta \vdash M\{y^A \leftarrow N\}^B \,|\, s\{y^A \leftarrow t\}$ by T-$\supset$I.
- T-$\supset$E: $M = (M_1 M_2)^B$ and $s = s_1 \cdot s_2$, where $\Theta; \Gamma, y^A; \Delta \vdash M_1^{C \supset B} \,|\, s_1$ and $\Theta; \Gamma, y^A; \Delta \vdash M_2^C \,|\, s_2$ are derivable by hypothesis. By IH, $\Theta; \Gamma; \Delta \vdash M_1\{y^A \leftarrow N\}^{C \supset B} \,|\, s_1\{y^A \leftarrow t\}$ and $\Theta; \Gamma; \Delta \vdash M_2\{y^A \leftarrow N\}^C \,|\, s_2\{y^A \leftarrow t\}$ are also derivable. We obtain $\Theta; \Gamma; \Delta \vdash M\{y^A \leftarrow N\}^B \,|\, s\{y^A \leftarrow t\}$ by T-$\supset$E.
- T-$\Box$I: $M = (!M')^{[\![s']\!]C}$, $s = !s'$, $B = [\![s']\!]C$ and there is a proof witness r s.t. $\Theta; \cdot; \cdot \vdash M'^C \,|\, r$ and $\Theta; \cdot; \cdot \vdash r \equiv s' : C$ are derivable by hypothesis. Since $M\{y^A \leftarrow N\} = M$ and $s\{y^A \leftarrow t\} = s$, then $\Theta; \Gamma; \Delta \vdash M\{y^A \leftarrow N\}^B \,|\, s\{y^A \leftarrow t\}$ is already derivable by hypothesis.
- T-$\Box$E: $M = (N'\langle v^D{:=}r, M'\rangle)^{C\{v^D \leftarrow r\}}$, $s = s_2\langle v^D{:=}r, s_1\rangle$, $B = C\{v^D \leftarrow r\}$ and, by hypothesis, both $\Theta; \Gamma, y^A; \Delta \vdash M'^{[\![r]\!]D} \,|\, s_1$ and $\Theta, v^D; \Gamma, y^A; \Delta \vdash N'^C \,|\, s_2$ are derivable. From these two judgments, we can obtain $\Theta; \Gamma; \Delta \vdash M\{y^A \leftarrow N\}^B \,|\, s\{y^A \leftarrow t\}$ by IH and T-$\Box$E.
- T-Name: $M = ([\alpha^C]M')^\perp$, $s = [\alpha^C]s'$, $B = \perp$, $\Delta = \Delta', \alpha^C$ and the judgment $\Theta; \Gamma, y^A; \Delta', \alpha^C \vdash M'^C \,|\, s'$ is derivable by hypothesis. By IH, the judgment $\Theta; \Gamma; \Delta', \alpha^C \vdash M'\{y^A \leftarrow N\}^C \,|\, s'\{y^A \leftarrow t\}$ is also derivable, and thus we can derive $\Theta; \Gamma; \Delta \vdash M\{y^A \leftarrow N\}^B \,|\, s\{y^A \leftarrow t\}$ by T-Name.
- T-NAbs: $M = (\mu\alpha^B.M')^B$, $s = \mu\alpha^B.s'$ and $\Theta; \Gamma, y^A; \Delta, \alpha^B \vdash M'^\perp \,|\, s'$ is derivable by hypothesis.
  We can obtain $\Theta; \Gamma; \Delta \vdash M\{y^A \leftarrow N\}^B \,|\, s\{y^A \leftarrow t\}$ by IH and T-NAbs.
- T-PlusL: $M = (M_1 +_{\mathrm{L}} s_2)^B$, $s = s_1 + s_2$ and $\Theta; \Gamma, y^A; \Delta \vdash M_1^B \,|\, s_1$ is derivable by hypothesis. By IH, $\Theta; \Gamma; \Delta \vdash M_1\{y^A \leftarrow N\}^B \,|\, s_1\{y^A \leftarrow t\}$ is also derivable, and we can obtain $\Theta; \Gamma; \Delta \vdash M_1\{y^A \leftarrow N\} +_{\mathrm{L}} s_2\{y^A \leftarrow t\}^B \,|\, s_1\{y^A \leftarrow t\} + s_2\{y^A \leftarrow t\}$ by T-PlusL.
- T-PlusR: this case is analogous to the previous one.

$\Box$

Proof of Lemma 4.4.13

Proof.- By mutual induction on the derivations of $\Theta, v^A; \Gamma; \Delta \vdash M^B \,|\, s$ and $\Theta, v^A; \Gamma; \Delta \vdash s \equiv r : B$. We consider the last step of each derivation.

For (1):

- T-Var: $M = s = x^B$, $M\{v^A \leftarrow N^A, t\} = s\{v^A \leftarrow t\} = x^B$ and $x^B \in \Gamma$. Note that, due to the freshness condition and since $x^B \in \Gamma$, then $v^A \notin \mathsf{FVV}(B)$, and thus $B\{v^A \leftarrow t\} = B$. $\Theta; \Gamma; \Delta \vdash x^B \,|\, s$ is derivable by T-Var.

- T-VarM: $M = s = w^B$. there are now two cases:
  - $w^B = v^A$: here $A = B$, $M\{v^A \leftarrow N^A, t\} = N$ and $s\{v^A \leftarrow t\} = t$. Note that, since $A = B$, then $v^A \notin \mathsf{FVV}(B)$. Then $\Theta; \Gamma; \Delta \vdash M\{v^A \leftarrow N^A, t\}^{B\{v^A \leftarrow t\}} \mid s\{v^A \leftarrow t\} = \Theta; \Gamma; \Delta \vdash N^A \mid t$, which is derivable by hypothesis.
  - $w^B \neq v^A$: here $M = s = M\{v^A \leftarrow N^A, t\} = s\{v^A \leftarrow t\} = w^B$ and $w^B \in \Theta$. By freshness condition, $v^A \notin \mathsf{FVV}(B)$ and thus $B\{v^A \leftarrow t\} = B$. By T-VarM, $\Theta; \Gamma; \Delta \vdash w^B \mid w^B$ is derivable.
- T-$\supset$I: $s = \lambda x^C.s'$, $B = C \supset D$, $M = (\lambda x^C.M')^B$ and $\Theta, v^A; \Gamma, x^C; \Delta \vdash M'^D \mid s'$ is derivable by hypothesis. By IH (1), the judgment $\Theta; \Gamma, x^C; \Delta \vdash M'\{v^A \leftarrow N^A, t\}^{D\{v^A \leftarrow t\}} \mid s'\{v^A \leftarrow t\}$ is derivable, and we obtain $\Theta; \Gamma; \Delta \vdash M\{v^A \leftarrow N^A, t\}^{B\{v^A \leftarrow t\}} \mid s\{v^A \leftarrow t\}$ by T-$\supset$I. Note that $v^A \notin \mathsf{FVT}(C)$, since x^C and v^A are both present in the contexts involved in the judgement $\Theta, v^A; \Gamma, x^C; \Delta \vdash M'^D \mid s'$; thus $(C \supset D)\{v^A \leftarrow t\} = C \supset (D\{v^A \leftarrow t\})$.
- T-$\supset$E: $s = s_1 \cdot s_2$, $M = M_1 M_2$, and both $\Theta, v^A; \Gamma; \Delta \vdash M_1^{C \supset B} \mid s_1$ and $\Theta, v^A; \Gamma; \Delta \vdash M^C \mid s_2$ are derivable by hypothesis. The result holds by induction hypothesis and T-$\supset$E.
- T-$\Box$I: $M = (!M')^{[\![s_1]\!]C}$, $s = !s_1$, $B = [\![s_1]\!]C$ and both $\Theta, v^A; \cdot; \cdot \vdash M'^C \mid s_2$ and $\Theta, v^A; \cdot; \cdot \vdash s_2 \equiv s_1 : C$ are derivable by hypothesis. By induction hypothesis (1), $\Theta; \cdot; \cdot \vdash M'\{v^A \leftarrow N^A, t\}^{C\{v^A \leftarrow t\}} \mid s_2\{v^A \leftarrow t\}$ is also derivable and, by IH (2), so is $\Theta; \cdot; \cdot \vdash s_2\{v^A \leftarrow t\} \equiv s_1\{v^A \leftarrow t\} : C\{v^A \leftarrow t\}$. We can obtain $\Theta; \Gamma; \Delta \vdash M\{v^A \leftarrow N^A, t\}^{([\![s_1]\!]C)\{v^A \leftarrow t\}} \mid s\{v^A \leftarrow t\}$ by T-$\Box$I.
- T-$\Box$E: $M = (N'\langle w^D{:=}r, M'\rangle)^{C\{w^D \leftarrow r\}}$, $s = s_2\langle w^D{:=}r, s_1\rangle$, $B = C\{w^D \leftarrow r\}$, and the judgments $\Theta, v^A; \Gamma; \Delta \vdash M'^{[\![r]\!]D} \mid s_1$ and $\Theta, v^A, w^D; \Gamma; \Delta \vdash N'^C \mid s_2$ are derivable. There are now two cases to consider:
  - $w^D = v^A$: here $M\{v^A \leftarrow N, t\} = M$, $s\{v^A \leftarrow t\} = s$, $\Theta, v^A, w^D = \Theta, v^A$ and $B\{v^A \leftarrow t\} = B$ (since $w^D = v^A$ and $B = C\{w^D \leftarrow r\}$, this means that $v^A \notin \mathsf{FVV}(B)$). We can derive $\Theta, v^A; \Gamma; \Delta \vdash M^B \mid s$ from $\Theta, v^A; \Gamma; \Delta \vdash M'^{[\![r]\!]D} \mid s_1$ and $\Theta, v^A, w^D; \Gamma; \Delta \vdash N'^C \mid s_2$ by T-$\Box$E, and obtain $\Theta; \Gamma; \Delta \vdash M^B \mid s$ by Strengthening (since v^A is not free in M^B nor in s).
  - $w^D \neq v^A$: in this case $M\{v^A \leftarrow N, t\} = (N'\{v^A \leftarrow N, t\}$ $\langle w^D{:=}r\{v^A \leftarrow t\}, M'\{v^A \leftarrow N, t\}\rangle)^{C\{w^D \leftarrow r\}\{v^A \leftarrow t\}}$, and $s\{v^A \leftarrow t\} =$ $s_2\{v^A \leftarrow t\}\langle w^D{:=}r\{v^A \leftarrow t\}, s_1\{v^A \leftarrow t\}\rangle$. Note that, by freshness convention, $v^A \notin \mathsf{FVV}(D)$, and thus $([\![r]\!]D)\{v^A \leftarrow t\} = [\![r\{v^A \leftarrow t\}]\!]D$. Also, by the variable convention, $w^D \notin \mathsf{FVV}(t)$ (nor $\mathsf{FVV}(N^A)$).
    By induction hypothesis (1), both $\Theta; \Gamma; \Delta \vdash M'\{v^A \leftarrow N^A, t\}^{([\![r]\!]D)\{v^A \leftarrow t\}} \mid s_1\{v^A \leftarrow t\}$ and $\Theta, w^D; \Gamma; \Delta \vdash N'\{v^A \leftarrow N^A, t\}^{C\{v^A \leftarrow t\}} \mid s_2\{v^A \leftarrow t\}$ are derivable. We can obtain $\Theta; \Gamma; \Delta \vdash M\{v^A \leftarrow N^A, t\}^{C\{v^A \leftarrow t\}\{w^D \leftarrow r\{v^A \leftarrow t\}\}} \mid s\{v^A \leftarrow t\}$ by T-$\Box$E.
    All that remains is to prove that $C\{v^A \leftarrow t\}\{w^D \leftarrow r\{v^A \leftarrow t\}\} = C\{w^D \leftarrow r\}\{v^A \leftarrow t\}$, but this holds by Lemma 4.4.9, since $w^D \notin \mathsf{FVV}(t)$.
- T-Name: $M = ([\alpha^C]M')^\perp$, $s = [\alpha^C]s$, $B = \perp$ and $\Theta, v^A; \Gamma; \Delta, \alpha^C \vdash M'^C \mid s'$ is derivable by hypothesis. By IH (1), $\Theta; \Gamma; \Delta, \alpha^C \vdash M'\{v^A \leftarrow N, t\}^C \mid s'\{v^A \leftarrow t\}$ and, by freshness convention, $v^A \notin \mathsf{FVV}(C)$. Then $C\{v^A \leftarrow t\} = C$ and $\Theta; \Gamma; \Delta \vdash M\{v^A \leftarrow N^A, t\}^{B\{v^A \leftarrow t\}} \mid s\{v^A \leftarrow t\}$ is derivable by T-Name.

- T-NAbs: $M = (\mu\alpha^B.M')^B$, $s = \mu\alpha^B.s'$ and $\Theta, v^A; \Gamma; \Delta, \alpha^B \vdash M'^{\perp} \,|\, s'$ is derivable by hypothesis. By freshness convention, $v^A \notin \mathsf{FVV}(B)$, and therefore $B\{v^A \leftarrow t\} = B$. By induction hypothesis (1), $\Theta; \Gamma; \Delta, \alpha^B \vdash M'\{v^A \leftarrow N^A, t\}^{\perp} \,|\, s'\{v^A \leftarrow t\}$ is also derivable ($\perp\{v^A \leftarrow t\} = \perp$), and thus we can derive $\Theta; \Gamma; \Delta \vdash (\mu\alpha^B.M')^B\{v^A \leftarrow N, t\}^B \,|\, \mu\alpha^B.s'\{v^A \leftarrow t\}$ by T-NAbs.
- T-PlusL: $M = (M_1 +_{\mathsf{L}} s_2)^B$, $s = s_1 + s_2$ and $\Theta, v^A; \Gamma; \Delta \vdash M_1^B \,|\, s_1$ is derivable by hypothesis. By induction hypothesis (1), $\Theta; \Gamma; \Delta \vdash M_1\{v^A \leftarrow N, t\}^{B\{v^A \leftarrow t\}} \,|\, s_1\{v^A \leftarrow t\}$ is also derivable, and we can derive $\Theta; \Gamma; \Delta \vdash M\{v^A \leftarrow N^A, t\}^{B\{v^A \leftarrow t\}} \,|\, s\{v^A \leftarrow t\}$ by T-PlusL.
- T-PlusR: this case is analogous to the previous one.

For (2):

- Eq-β: $s = (\lambda x^C.s_1) \cdot s_2$, $r = s_1\{x^C \leftarrow s_2\}$, and both $\Theta, v^A; \Gamma, x^C; \Delta \vdash B \,|\, s_1$ and $\Theta, v^A; \Gamma; \Delta \vdash C \,|\, s_2$ are derivable by hypothesis.

  By IH (1), $\Theta; \Gamma, x^C; \Delta \vdash B\{v^A \leftarrow t\} \,|\, s_1\{v^A \leftarrow t\}$ and $\Theta; \Gamma; \Delta \vdash C\{v^A \leftarrow t\} \,|\, s_2\{v^A \leftarrow t\}$ are also derivable. Note that, by freshness convention, $C\{v^A \leftarrow t\} = C$.

  By Eq-β, we can derive $\Theta; \Gamma; \Delta \vdash (\lambda x^C.s_1\{v^A \leftarrow t\}) \cdot (s_2\{v^A \leftarrow t\}) \equiv s_1\{v^A \leftarrow t\}\{x^C \leftarrow s_2\{v^A \leftarrow t\}\} : B\{v^A \leftarrow t\}$, which, by Lemma 4.4.9, is the same as $\Theta; \Gamma; \Delta \vdash ((\lambda x^C.s_1) \cdot s_2)\{v^A \leftarrow t\} \equiv s_1\{x^C \leftarrow s_2\}\{v^A \leftarrow t\} : B\{v^A \leftarrow t\}$.
- Eq-γ: $B = C\{w^D \leftarrow s_1\}$, $s = s_2\langle w^D{:=}s_1, !s_1\rangle$, $r = s_2\{w^D \leftarrow s_1\}$ and both $\Theta, v^A; \cdot; \cdot \vdash D \,|\, s_1$ and $\Theta, v^A, w^D; \Gamma; \Delta \vdash C \,|\, s_2$ are derivable by hypothesis.

  There are two possibilities:

  - $v^A = w^D$: in this case $s\{v^A \leftarrow t\} = s$, $r\{v^A \leftarrow t\} = r$ and $B\{v^A \leftarrow t\} = B$. $\Theta, v^A; \Gamma; \Delta \vdash s \equiv r : B$ is already derivable by hypothesis and, by Strengthening (since $v^A \notin \mathsf{FVV}(s)$), so is $\Theta; \Gamma; \Delta \vdash s \equiv r : B$.
  - $v^A \neq w^D$: here, by IH (1), we can derive both $\Theta; \cdot; \cdot \vdash D\{v^A \leftarrow t\} \,|\, s_1\{v^A \leftarrow t\}$ and $\Theta, w^D; \Gamma; \Delta \vdash C\{v^A \leftarrow t\} \,|\, s_2\{v^A \leftarrow t\}$.
    By Eq-γ, $\Theta; \Gamma; \Delta \vdash s_2\{v^A \leftarrow t\}\langle w^D{:=}s_1\{v^A \leftarrow t\}, !s_1\{v^A \leftarrow t\}\rangle \equiv s_2\{v^A \leftarrow t\}\{w^D \leftarrow s_1\{v^A \leftarrow t\}\} : C\{v^A \leftarrow t\}\{w^D \leftarrow s_1\{v^A \leftarrow t\}\}$ is derivable. And this is the same as $\Theta; \Gamma; \Delta \vdash (s_2\langle w^D{:=}s_1, !s_1\rangle)\{v^A \leftarrow t\} \equiv (s_2\{w^D \leftarrow s_1\})\{v^A \leftarrow t\} : C\{w^D \leftarrow s_1\}\{v^A \leftarrow t\}$ by Lemma 4.4.9.
- Eq-μ: $B = B\{v^A \leftarrow t\} = \perp$, $s = [\beta^C]\mu\alpha^C.s'$, $r = s'\{\alpha^C \leftarrow \beta^C\}$, $\Delta = \Delta', \alpha^C, \beta^C$ and the judgement $\Theta, v^A; \Gamma; \Delta', \alpha^C, \beta^C \vdash \perp \,|\, s'$ is derivable by hypothesis.

  By induction hypothesis (1), we can derive $\Theta; \Gamma; \Delta', \alpha^C, \beta^C \vdash \perp \,|\, s'\{v^A \leftarrow t\}$. And, by Eq-$\mu$, $\Theta; \Gamma; \Delta \vdash [\beta^C]\mu\alpha^C.s'\{v^A \leftarrow t\} \equiv s'\{v^A \leftarrow t\}\{\alpha^C \leftarrow \beta^C\} : \perp$, which is the same as $\Theta; \Gamma; \Delta \vdash s\{v^A \leftarrow t\} \equiv s'\{\alpha^C \leftarrow \beta^C\}\{v^A \leftarrow t\} : \perp$, since $\alpha^C \notin \mathsf{FVF}(t)$ by the variable convention.
- Eq-ζ: $s = (\mu\alpha^{C \supset B}.s_1) \cdot s_2$, $r = \mu\beta^B.s_1(\!|_{[\alpha^{C\supset B}](\bullet) \leftarrow [\beta^B](\bullet)s_2}|\!)$ and both $\Theta, v^A; \Gamma; \Delta, \alpha^{C\supset B} \vdash \perp \,|\, s_1$ and $\Theta, v^A; \Gamma; \Delta \vdash C \,|\, s_2$ are derivable by hypothesis. By IH (1), we can derive $\Theta; \Gamma; \Delta, \alpha^{C\supset B} \vdash \perp \,|\, s_1\{v^A \leftarrow t\}$ and $\Theta; \Gamma; \Delta \vdash C\{v^A \leftarrow t\} \,|\, s_2\{v^A \leftarrow t\}$. Additionally, by freshness convention, $B\{v^A \leftarrow t\} = B$ and $C\{v^A \leftarrow t\} = C$. By Eq-$\zeta$, $\Theta; \Gamma; \Delta \vdash$

$(\mu\alpha^{C\supset B}.s_1\{v^A \leftarrow t\}) \cdot s_2\{v^A \leftarrow t\} \equiv \mu\beta^B.s_1\{v^A \leftarrow t\}(\!|\, [\alpha^{C\supset B}](\bullet) \leftarrow [\beta^B](\bullet)s_2\{v^A \leftarrow t\} \,|\!) : B$ is derivable, and this is the same as $\Theta;\Gamma;\Delta \vdash s\{v^A \leftarrow t\} \equiv r\{v^A \leftarrow t\} : B$ by Lemma 4.4.9.

- Eq-θ: $s = \mu\alpha^B.[\alpha^B]r$ and $\Theta, v^A;\Gamma;\Delta \vdash B \,|\, r$ is derivable by hypothesis. By IH (1), $\Theta;\Gamma;\Delta \vdash B \,|\, r\{v^A \leftarrow t\}$ is also derivable ($B\{v^A \leftarrow t\} = B$ by freshness convention).
  By Eq-θ, we obtain $\Theta;\Gamma;\Delta \vdash \mu\alpha^B.[\alpha^B]r\{v^A \leftarrow t\} \equiv r\{v^A \leftarrow t\} : B$.
- Eq-ψ_L: $s = (s_1 + s_2) \cdot s_3$, $r = (s_1 \cdot s_3) + s_2$ and both $\Theta, v^A;\Gamma;\Delta \vdash C \supset B \,|\, s_1$ and $\Theta, v^A;\Gamma;\Delta \vdash C \,|\, s_3$ are derivable by hypothesis.
  By IH (1), we can derive $\Theta;\Gamma;\Delta \vdash C\{v^A \leftarrow t\} \supset B\{v^A \leftarrow t\} \,|\, s_1\{v^A \leftarrow t\}$ and $\Theta;\Gamma;\Delta \vdash C\{v^A \leftarrow t\} \,|\, s_3\{v^A \leftarrow t\}$. The result follows by Eq-ψ_L.
- Eq-ψ_R: This case is analogous to the previous one.
- Eq-ϕ_L: $B = C\{w^D \leftarrow s_2\}$, $s = s_1\langle w^D{:=}s_2, (s_3 + s_4)\rangle$, $r = s_1\langle w^D{:=}s_2, s_3\rangle + s_4$, both $\Theta, v^A;\Gamma;\Delta \vdash [\![s_2]\!]D \,|\, s_3$ and $\Theta, w^D, v^A;\Gamma;\Delta \vdash C \,|\, s_1$ are derivable by hypothesis and $w^D \notin \mathsf{FVV}(s_4)$. There are two possibilities:
  - $v^A = w^D$: in this case $s\{v^A \leftarrow t\} = s$, $r\{v^A \leftarrow t\} = r$ and $B\{v^A \leftarrow t\} = B$. $\Theta, v^A;\Gamma;\Delta \vdash s \equiv r : B$ is already derivable by hypothesis and, by Strengthening (since $v^A \notin \mathsf{FVV}(s)$), so is $\Theta;\Gamma;\Delta \vdash s \equiv r : B$.
  - $v^A \neq w^D$: here, by IH (1), we can derive both $\Theta;\Gamma;\Delta \vdash [\![s_2\{v^A \leftarrow t\}]\!]D \,|\, s_3\{v^A \leftarrow t\}$ and $\Theta, w^D;\Gamma;\Delta \vdash C\{v^A \leftarrow t\} \,|\, s_1\{v^A \leftarrow t\}$. Note that $D\{v^A \leftarrow t\} = D$ by freshness convention.
    By Eq-ϕ_L, we can derive:
    $\Theta;\Gamma;\Delta \vdash s\{v^A \leftarrow t\} \equiv r\{v^A \leftarrow t\} : C\{v^A \leftarrow t\}\{w^D \leftarrow s_2\{v^A \leftarrow t\}\}$, which is the same as $\Theta;\Gamma;\Delta \vdash s\{v^A \leftarrow t\} \equiv r\{v^A \leftarrow t\} : B\{v^A \leftarrow t\}$ by Lemma 4.4.9.
- Eq-ϕ_R: This case is analogous to the previous one.
- Eq-χ_L: $s = [\alpha^C]s_1 + s_2$, $r = ([\alpha^C]s_1) + s_2$, $\Delta = \Delta', \alpha^C$, $B = B\{v^A \leftarrow t\} = \bot$ and $\Theta, v^A;\Gamma;C \vdash \Delta', \alpha^C \,|\, s_1$ is derivable by hypothesis. Also, by freshness convention, $C\{v^A \leftarrow t\} = C$. By IH (1), we can derive $\Theta;\Gamma;\Delta \vdash C \,|\, s_1\{v^A \leftarrow t\}$. The result follows by Eq-χ_L.
- Eq-χ_R: This case is analogous to the previous one.
- Eq-$\lightning_L$: $s = \mu\alpha^B.s_1 + s_2$, $r = (\mu\alpha^B.s_1) + s_2$ and $\Theta, v^A;\Gamma;\bot \vdash \Delta, \alpha^B \,|\, s_1$ is derivable by hypothesis. By freshness convention, $B\{v^A \leftarrow t\} = B$.
  By IH (1), we can derive $\Theta;\Gamma;\Delta, \alpha^B \vdash \bot \,|\, s_1\{v^A \leftarrow t\}$. The result follows by Eq-$\lightning_L$.
- Eq-$\lightning_R$: This case is analogous to the previous one.
- Eq-Refl: this case is straightforward.
- Eq-Symm: in this case $\Theta, v^A;\Gamma;\Delta \vdash r \equiv s : B$ is derivable by hypothesis. By IH (2), we can derive $\Theta;\Gamma;\Delta \vdash r\{v^A \leftarrow t\} \equiv s\{v^A \leftarrow t\} : B\{v^A \leftarrow t\}$, and then obtain $\Theta;\Gamma;\Delta \vdash s\{v^A \leftarrow t\} \equiv r\{v^A \leftarrow t\} : B\{v^A \leftarrow t\}$ by Eq-Symm.
- Eq-Trans: analogously to the previous case, the result is obtained by IH (2) and Eq-Trans.
- Eq-$\langle\rangle$: $s = s_1\langle w^D{:=}t', s_2\rangle$, $r = r_1\langle w^D{:=}t', r_2\rangle$, $B = C\{w^D \leftarrow t'\}$ and both $\Theta, v^A;\Gamma;\Delta \vdash s_2 \equiv r_2 : [\![t']\!]D$ and $\Theta, w^D, v^A;\Gamma;\Delta \vdash s_1 \equiv r_1 : C$ are derivable by hypothesis. By freshness convention, $D\{v^A \leftarrow t\} = D$.
  There are two possibilities:

- $v^A = w^D$: in this case $s\{v^A \leftarrow t\} = s$, $r\{v^A \leftarrow t\} = r$ and $B\{v^A \leftarrow t\} = B$. $\Theta, v^A; \Gamma; \Delta \vdash s \equiv r : B$ is already derivable by hypothesis and, by Strengthening (since $v^A \notin \mathsf{FVV}(s)$), so is $\Theta; \Gamma; \Delta \vdash s \equiv r : B$.
- $v^A \neq w^D$: here, by IH (2), we can derive $\Theta; \Gamma; \Delta \vdash s_2\{v^A \leftarrow t\} \equiv r_2\{v^A \leftarrow t\} : [\![t'\{v^A \leftarrow t\}]\!]D$ and $\Theta, w^D; \Gamma; \Delta \vdash s_1\{v^A \leftarrow t\} \equiv r_1\{v^A \leftarrow t\} : C\{v^A \leftarrow t\}$. By Eq-$\langle\rangle$, we obtain $\Theta; \Gamma; \Delta \vdash s\{v^A \leftarrow t\} \equiv r\{v^A \leftarrow t\} : C\{v^A \leftarrow t\}\{W^D \leftarrow t'\{v^A \leftarrow t\}\}$, which is the same as $\Theta; \Gamma; \Delta \vdash s\{v^A \leftarrow t\} \equiv r\{v^A \leftarrow t\} : B\{v^A \leftarrow t\}$ by Lemma 4.4.9.

- Eq-λ: $B = C \supset D$, $s = \lambda x^C.s'$, $r = \lambda x^C.r'$ and $\Theta, v^A; \Gamma; \Delta \vdash r \equiv s : D$ is derivable by hypothesis. Note that $C\{v^A \leftarrow t\} = C$ by freshness condition. The result follows by IH (2) and Eq-λ.
- All other cases (Eq-$\cdot$, Eq-$[\alpha]$, Eq-$\mu\alpha$, Eq-$+_{\mathrm{L}}$ and Eq-$+_{\mathrm{R}}$) are analogous to the previous one, using the IH (2) and the corresponding inference rule. In the cases of T-Name and T-NAbs, $B\{v^A \leftarrow t\} = B$ by freshness convention.

□

Proof of Lemma 4.4.14

Proof.- By induction on the derivation of $\Theta; \Gamma; \Delta \vdash M^B \mid s$.

- T-Var: $M^B = s = M\{v^A \leftarrow N^A, t\} = s\{v^A \leftarrow t\} = x^B$ and $x^B \in \Gamma$. Note that, due to the freshness condition and since $x^B \in \Gamma$, then $v^A \notin \mathsf{FVV}(B)$, and thus $B\{v^A \leftarrow t\} = B$. We know by hypothesis that $\Theta, v^A; \Gamma; \Delta \vdash x^B \mid s$ is derivable. By Strengthening (since $v^A \notin \mathsf{FVV}(x^B)$), so is $\Theta; \Gamma; \Delta \vdash x^B \mid x^B$. Take $s' = x^B$.
- T-VarM: $M^B = s = w^B$. There are now two cases:
  - $w^B = v^A$: here $A = B$, $M\{v^A \leftarrow N^A, t\} = N^A$ and $s\{v^A \leftarrow t\} = t$. Note that, since $A = B$, then $v^A \notin \mathsf{FVV}(B)$. Take $s' = r$. Then $\Theta; \Gamma; \Delta \vdash M\{v^A \leftarrow N^A, t\}^{B\{v^A \leftarrow t\}} \mid s' = \Theta; \Gamma; \Delta \vdash N^A \mid r$, which is derivable by hypothesis.
  - $w^B \neq v^A$: here $M = s = M\{v^A \leftarrow N^A, t\} = s\{v^A \leftarrow t\} = w^B$ and $w^B \in \Theta$. Take $s' = w^B$. By freshness condition, $v^A \notin \mathsf{FVV}(B)$ and thus $B\{v^A \leftarrow t\} = B$. By T-VarM, $\Theta; \Gamma; \Delta \vdash w^B \mid w^B$ is derivable.
- T-$\supset$I: $s = \lambda x^C.s'$, $B = C \supset D$, $M = (\lambda x^C.M_1)^B$ and $\Theta, v^A; \Gamma, x^C; \Delta \vdash M_1^D \mid s_1$ is derivable by hypothesis. By induction hypothesis, the judgements $\Theta; \Gamma, x^C; \Delta \vdash M_1\{v^A \leftarrow N^A, t\}^{D\{v^A \leftarrow t\}} \mid s_1'$ and $\Theta; \Gamma, x^C; \Delta \vdash s_1' \equiv s_1\{v^A \leftarrow t\} : D\{v^A \leftarrow t\}$ are both derivable. Take $s' = \lambda x^C.s_1'$. We obtain $\Theta; \Gamma; \Delta \vdash M\{v^A \leftarrow N^A, t\}^{B\{v^A \leftarrow t\}} \mid \lambda x^C.s_1'$ by T-$\supset$I, and $\Theta; \Gamma; \Delta \vdash s' \equiv s\{v^A \leftarrow t\} : A$ by Eq-λ.

  Note that $v^A \notin \mathsf{FVT}(C)$, since x^C and v^A are both present in the contexts involved in the judgement $\Theta, v^A; \Gamma, x^C; \Delta \vdash M'^D \mid s'$; thus $(C \supset D)\{v^A \leftarrow t\} = C \supset (D\{v^A \leftarrow t\})$.
- T-$\supset$E: $s = s_1 \cdot s_2$, $M = M_1 M_2$, and both $\Theta, v^A; \Gamma; \Delta \vdash M_1^{C \supset B} \mid s_1$ and $\Theta, v^A; \Gamma; \Delta \vdash M^C \mid s_2$ are derivable by hypothesis. By induction hypothesis, the following judgements are derivable:
  1. $\Theta; \Gamma; \Delta \vdash M_1\{v^A \leftarrow N^A, t\}^{(C\{v^A \leftarrow t\}) \supset (B\{v^A \leftarrow t\})} \mid s_1'$

2. $\Theta;\Gamma;\Delta \vdash M_2\{v^A \leftarrow N^A, t\}^{C\{v^A \leftarrow t\}} \mid s_2'$
3. $\Theta;\Gamma;\Delta \vdash s_1' \equiv s_1\{v^A \leftarrow t\} : (C\{v^A \leftarrow t\}) \supset (B\{v^A \leftarrow t\})$
4. $\Theta;\Gamma;\Delta \vdash s_2' \equiv s_2\{v^A \leftarrow t\} : C\{v^A \leftarrow t\}$

Take $s' = s\{v^A \leftarrow r\} = s_1\{v^A \leftarrow r\} \cdot s_2\{v^A \leftarrow t\}$.
By T-$\supset$E from 1. and 2., we can derive:
$\Theta;\Gamma;\Delta \vdash M_1\{v^A \leftarrow N^A, t\} M_2\{v^A \leftarrow N^A, t\}^{B\{v^A \leftarrow t\}} \mid s_1' \cdot s_2'$.
And by Eq-$\cdot$ from 3. and 4.:
$\Theta;\Gamma;\Delta \vdash s_1' \cdot s_2' \equiv s_1\{v^A \leftarrow r\} \cdot s_2\{v^A \leftarrow t\} : B\{v^A \leftarrow t\}$.

- T-$\Box$I: $M = (!M_1)^{[\![s_1]\!]C}$, $s = !s_1$, $B = [\![s_1]\!]C$ and both $\Theta, v^A; \cdot; \cdot \vdash M_1^C \mid s_2$ and $\Theta, v^A; \cdot; \cdot \vdash s_2 \equiv s_1 : C$ are derivable by hypothesis.
  By IH, we can derive both $\Theta; \cdot; \cdot \vdash M_1\{v^A \leftarrow N^A, t\}^{C\{v^A \leftarrow t\}} \mid s_2'$ and $\Theta; \cdot; \cdot \vdash s_2' \equiv s_2\{v^A \leftarrow t\} : C\{v^A \leftarrow t\}$.
  Since $\Theta, v^A; \cdot; \cdot \vdash s_2 \equiv s_1 : C$ is derivable, then by Lemma 4.4.13 so is $\Theta; \cdot; \cdot \vdash s_2\{v^A \leftarrow t\} \equiv s_1\{v^A \leftarrow t\} : C\{v^A \leftarrow t\}$. And, by Eq-Trans, $\Theta; \cdot; \cdot \vdash s_2' \equiv s_1\{v^A \leftarrow t\} : C\{v^A \leftarrow t\}$.
  Take $s' = s\{v^A \leftarrow t\} = !(s_1\{v^A \leftarrow t\})$. By T-$\Box$I, we obtain:
  $\Theta;\Gamma;\Delta \vdash !(M_1\{v^A \leftarrow N^A, t\})^{[\![s_1\{v^A \leftarrow t\}]\!]C\{v^A \leftarrow t\}} \mid !(s_1\{v^A \leftarrow t\})$.
- T-$\Box$E: $M = (N'\langle w^D := r', M'\rangle)^{C\{w^D \leftarrow r'\}}$, $s = s_2\langle w^D := r', s_1\rangle$, $B = C\{w^D \leftarrow r'\}$, and the judgments $\Theta, v^A; \Gamma; \Delta \vdash M'^{[\![r']\!]D} \mid s_1$ and $\Theta, v^A, w^D; \Gamma; \Delta \vdash N'^C \mid s_2$ are derivable. Take $s' = s\{v^A \leftarrow t\}$. There are now two cases to consider:
  - $w^D = v^A$: here $M\{v^A \leftarrow N, t\} = M$, $s\{v^A \leftarrow t\} = s$, $\Theta, v^A, w^D = \Theta, v^A$ and $B\{v^A \leftarrow t\} = B$ (since $w^D = v^A$ and $B = C\{w^D \leftarrow r'\}$, this means that $v^A \notin \mathsf{FVV}(B)$). We can derive $\Theta, v^A; \Gamma; \Delta \vdash M^B \mid s$ from $\Theta, v^A; \Gamma; \Delta \vdash M'^{[\![r']\!]D} \mid s_1$ and $\Theta, v^A, w^D; \Gamma; \Delta \vdash N'^C \mid s_2$ by T-$\Box$E, and obtain $\Theta;\Gamma;\Delta \vdash M^B \mid s$ by Strengthening (since v^A is not free in either M or s).
  - $w^D \neq v^A$: in this case $M\{v^A \leftarrow N, t\} = (N'\{v^A \leftarrow N, t\}$ $\langle w^D := r\{v^A \leftarrow t\}, M'\{v^A \leftarrow N, t\}\rangle)^{C\{w^D \leftarrow r\}\{v^A \leftarrow t\}}$, and $s\{v^A \leftarrow t\} =$ $s_2\{v^A \leftarrow t\}\langle w^D := r\{v^A \leftarrow t\}, s_1\{v^A \leftarrow t\}\rangle$. Note that, by freshness convention, $v^A \notin \mathsf{FVV}(D)$, and thus $([\![r']\!]D)\{v^A \leftarrow t\} = [\![r'\{v^A \leftarrow t\}]\!]D$. Also, by the variable convention, $w^D \notin \mathsf{FVV}(t)$ (nor $\mathsf{FVV}(N)$).
    By IH, the following judgements are derivable for some witnesses s_3 and s_4:

    $$\begin{array}{l}\Theta;\Gamma;\Delta \vdash M'\{v^A \leftarrow N, t\}^{[\![r'\{v^A \leftarrow t\}]\!]D} \mid s_4\\ \Theta, w^D;\Gamma;\Delta \vdash N'\{v^A \leftarrow N, t\}^{C\{v^A \leftarrow t\}} \mid s_3\\ \Theta;\Gamma;\Delta \vdash s_3 \equiv s_2\{v^A \leftarrow t\} : C\{v^A \leftarrow t\}\\ \Theta, w^D;\Gamma;\Delta \vdash s_4 \equiv s_1\{v^A \leftarrow t\} : [\![r'\{v^A \leftarrow t\}]\!]D\end{array}$$

    We can use T-$\Box$E to derive:
    $\Theta;\Gamma;\Delta \vdash N'\{v^A \leftarrow N, t\}\langle w^D := r'\{v^A \leftarrow t\}, M'\{v^A \leftarrow N, t\}\rangle^{C\{v^A \leftarrow t\}\{w^D \leftarrow (r'\{v^A \leftarrow t\})\}}$
    $\mid s_4\langle w^D := r'\{v^A \leftarrow t\}, s_3\rangle$.
    All that remains is to prove that $C\{v^A \leftarrow t\}\{w^D \leftarrow r'\{v^A \leftarrow t\}\} = C\{w^D \leftarrow r'\}\{v^A \leftarrow t\}$, but this holds by Lemma 4.4.9, since $w^D \notin \mathsf{FVV}(t)$.
    Take $s' = s_4\langle w^D := r'\{v^A \leftarrow t\}, s_3\rangle$. We can derive $\Theta;\Gamma;\Delta \vdash s' \equiv s\{v^A \leftarrow t\} : C\{w^D \leftarrow r'\}\{v^A \leftarrow t\}$ by Eq-$\langle\rangle$.

- T-Name: $M = ([\alpha^C]M_1)^\perp$, $s = [\alpha^C]s$, $B = \perp$ and $\Theta, v^A; \Gamma; \Delta, \alpha^C \vdash M_1^C \mid s_1$ is derivable by hypothesis. Take $s' = s\{v^A \leftarrow t\} = [\alpha^C]s\{v^A \leftarrow t\}$. By induction hypothesis, $\Theta; \Gamma; \Delta, \alpha^C \vdash M_1\{v^A \leftarrow N, t\}^C \mid s_1\{v^A \leftarrow t\}$ and, by freshness convention, $v^A \notin \mathsf{FVV}(C)$. Then $C\{v^A \leftarrow t\} = C$ and $\Theta; \Gamma; \Delta \vdash M\{v^A \leftarrow N^A, t\}^{B\{v^A \leftarrow t\}} \mid s\{v^A \leftarrow t\}$ is derivable by T-Name.
- T-NAbs: $M = (\mu\alpha^B.M_1)^B$, $s = \mu\alpha^B.s_1$ and $\Theta, v^A; \Gamma; \Delta, \alpha^B \vdash M_1^\perp \mid s_1$ is derivable by hypothesis. By freshness convention, $v^A \notin \mathsf{FVV}(B)$, and therefore $B\{v^A \leftarrow t\} = B$. Take $s' = s\{v^A \leftarrow t\} = \mu\alpha^B.s_1\{v^A \leftarrow t\}$. By induction hypothesis, $\Theta; \Gamma; \Delta, \alpha^B \vdash M_1\{v^A \leftarrow N\}^\perp \mid s_1\{v^A \leftarrow t\}$ is also derivable ($\perp\{v^A \leftarrow t\} = \perp$), and thus we can derive $\Theta; \Gamma; \Delta \vdash (\mu\alpha^B.M_1)^B\{v^A \leftarrow N, t\}^B \mid \mu\alpha^B.s_1\{v^A \leftarrow t\}$ by T-NAbs.
- T-PlusL: $M = (M_1 +_{\mathsf{L}} s_2)^B$, $s = s_1 + s_2$ and $\Theta, v^A; \Gamma; \Delta \vdash M_1^B \mid s_1$ is derivable by hypothesis. Take $s' = s\{v^A \leftarrow t\}$. By induction hypothesis, $\Theta; \Gamma; \Delta \vdash M_1\{v^A \leftarrow N, t\}^{B\{v^A \leftarrow t\}} \mid s_1\{v^A \leftarrow t\}$ is derivable, and we can derive $\Theta; \Gamma; \Delta \vdash M\{v^A \leftarrow N^A, t\}^{B\{v^A \leftarrow t\}} \mid s\{v^A \leftarrow t\}$ by T-PlusL.
- T-PlusR: this case is analogous to the previous one.

□

Proof of Lemma 4.4.15

Proof.- By induction on the derivation of $\Theta; \Gamma; \Delta, \alpha^A, \beta^A \vdash M^B \mid s$. We consider the last step of the derivation.

- T-Var: in this case $s = s\{\alpha^A \leftarrow \beta^A\} = M = M\{\alpha^A \leftarrow \beta^A\} = x^B$. $\Theta; \Gamma; \Delta, \beta^A \vdash x^B \mid x^B$ is derivable by T-Var.
- T-VarM: this case is analogous to the previous one.
- T-$\supset$I: here $s = \lambda x^C.s'$, $B = C \supset D$ and $M = (\lambda x^C.M')^B$ for some term M', where $\Theta; \Gamma, x^C; \Delta, \alpha^A, \beta^A \vdash M'^D \mid s'$ is derivable by hypothesis. By IH, we can derive $\Theta; \Gamma, x^C; \Delta, \beta^A \vdash M'\{\alpha^A \leftarrow \beta^A\}^D \mid s'\{\alpha^A \leftarrow \beta^A\}$, and then obtain $\Theta; \Gamma; \Delta, \beta^A \vdash M\{\alpha^A \leftarrow \beta^A\}^B \mid s\{\alpha^A \leftarrow \beta^A\}$ by T-$\supset$I.
- T-$\supset$E: $M = (M_1 M_2)^B$ and $s = s_1 \cdot s_2$, where $\Theta; \Gamma; \Delta, \alpha^A, \beta^A \vdash M_1^{C \supset B} \mid s_1$ and $\Theta; \Gamma; \Delta, \alpha^A, \beta^A \vdash M_2^C \mid s_2$ are derivable by hypothesis. By IH, $\Theta; \Gamma; \Delta, \beta^A \vdash M_1\{\alpha^A \leftarrow \beta^A\}^{C \supset B} \mid s_1\{\alpha^A \leftarrow \beta^A\}$ and $\Theta; \Gamma; \Delta, \beta^A \vdash M_2\{\alpha^A \leftarrow \beta^A\}^C \mid s_2\{\alpha^A \leftarrow \beta^A\}$ are also derivable. We obtain $\Theta; \Gamma; \Delta, \beta^A \vdash M\{\alpha^A \leftarrow \beta^A\}^B \mid s\{\alpha^A \leftarrow \beta^A\}$ by T-$\supset$E.
- T-$\Box$I: $M = (!M')^{[\![s']\!]C}$, $s = !s'$and $B = [\![s']\!]C$. Since $M\{\alpha^A \leftarrow \beta^A\} = M$ and $s\{\alpha^A \leftarrow \beta^A\} = s$, then $\Theta; \Gamma; \Delta \vdash M\{\alpha^A \leftarrow \beta^A\}^B \mid s\{\alpha^A \leftarrow \beta^A\}$ is derivable by T-$\Box$I.
- T-$\Box$E: $M = (N'^C\langle v^D{:=}r, M'\rangle)^{C\{v^D \leftarrow r\}}$, $s = s_2\langle v^D{:=}r, s_1\rangle$, $B = C\{v^D \leftarrow r\}$, and both $\Theta; \Gamma; \Delta, \alpha^A, \beta^A \vdash M'^{[\![r]\!]D} \mid s_1$ and $\Theta, v^D; \Gamma; \Delta, \alpha^A, \beta^A \vdash N'^C \mid s_2$ are derivable by hypothesis. From these two judgments, we can obtain $\Theta; \Gamma; \Delta, \beta^A \vdash M\{\alpha^A \leftarrow \beta^A\}^B \mid s\{\alpha^A \leftarrow \beta^A\}$ by IH and T-$\Box$E.
- T-Name: $M = ([\gamma^C]M')^\perp$, $s = [\gamma^C]s'$, $B = \perp$, $\Delta = \Delta', \gamma^C$. $\Theta; \Gamma; \Delta', \gamma^C, \alpha^A, \beta^A \vdash M'^C \mid s'$ is derivable by hypothesis and, by IH, $\Theta; \Gamma; \Delta', \gamma^C, \beta^A \vdash M'\{\alpha^A \leftarrow \beta^A\}^C \mid s'\{\alpha^A \leftarrow \beta^A\}$ is also derivable. Whether or not $\gamma^C = \alpha^A$, we can derive $\Theta; \Gamma; \Delta', \gamma^C, \beta^A \vdash M\{\alpha^A \leftarrow \beta^A\}^\perp \mid s'\{\alpha^A \leftarrow \beta^A\}$ from $\Theta; \Gamma; \Delta', \gamma^C, \beta^A \vdash M'\{\alpha^A \leftarrow \beta^A\}^C \mid s\{\alpha^A \leftarrow \beta^A\}$ by T-Name.

- T-NAbs: $M = (\mu\gamma^B.M')^B$, $s = \mu\gamma^B.s'$ and $\Theta;\Gamma;\Delta,\alpha^A,\beta^A,\gamma^B \vdash M'^{\perp} \mid s'$ is derivable by hypothesis.

  There are two possibilities:

  - $\gamma^B = \alpha^A$: in this case $M\{\alpha^A \leftarrow \beta^A\} = M$ and $s\{\alpha^A \leftarrow \beta^A\} = s$. We can obtain $\Theta;\Gamma;\Delta,\beta^A \vdash M^A \mid s$ from $\Theta;\Gamma;\Delta,\beta^A,\alpha^A \vdash M'^{\perp} \mid s'$ by T-NAbs.
  - $\gamma^B \neq \alpha^A$: here $\Theta;\Gamma;\Delta,\beta^A,\gamma^B \vdash M'\{\alpha^A \leftarrow \beta^A\}^{\perp} \mid s'\{\alpha^A \leftarrow \beta^A\}$ is derivable by IH, and thus we can derive $\Theta;\Gamma;\Delta,\beta^A \vdash M\{\alpha^A \leftarrow \beta^A\}^B \mid s\{\alpha^A \leftarrow \beta^A\}$ by T-Name.

- T-PlusL: $M = (M_1 +_{\mathrm{L}} s_2)^B$, $s = s_1 + s_2$ and $\Theta;\Gamma;\Delta,\alpha^A,\beta^A \vdash M_1^B \mid s_1$ is derivable by hypothesis. By IH, we can derive $\Theta;\Gamma;\Delta,\beta^A \vdash M_1\{\alpha^A \leftarrow \beta^A\}^B \mid s_1\{\alpha^A \leftarrow \beta^A\}$, and thus we can obtain $\Theta;\Gamma;\Delta,\beta^A \vdash M_1\{\alpha^A \leftarrow \beta^A\} +_{\mathrm{L}} s_2\{\alpha^A \leftarrow \beta^A\}^B \mid s_1\{\alpha^A \leftarrow \beta^A\} + s_2\{\alpha^A \leftarrow \beta^A\}$ by T-PlusL.

- T-PlusR: this case is analogous to the previous one.

□

Proof of Lemma 4.4.16

Proof.- By induction on the derivation of $\Theta;\Gamma;\Delta \vdash M^D \mid s$. Keep in mind that, by Lemma 4.4.5, $\mathsf{w}(N^A) = t$. We consider the last step of the derivation.

- T-Var: in this case $M = M(\!|\,[\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)N^A\,|\!) = s = s(\!|\,[\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)t\,|\!) = x^D$ and $\Gamma = \Gamma', x^D$. The judgment $\Theta;\Gamma',x^D;\Delta' \vdash x^D \mid x^D$ is derivable by T-Var.

- T-VarM: this case is analogous to the previous one.

- T-$\supset$I: here $D = C \supset D$, $M = (\lambda x^C.L)^D$, $s = \lambda x^C.r$ and $\Theta;\Gamma,x^C;\Delta,\alpha^{A\supset B} \vdash L^D \mid r$ is derivable by hypothesis. By IH, we can derive $\Theta;\Gamma,x^C;\Delta' \vdash L(\!|\,[\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)N^A\,|\!)^D \mid r(\!|\,[\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)t\,|\!)$, and then obtain $\Theta;\Gamma;\Delta' \vdash M(\!|\,[\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)N^A\,|\!)^D \mid s(\!|\,[\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)t\,|\!)$ by T-$\supset$I.

- T-$\supset$E: $M = (M_1 M_2)^D$, $s = s_1 \cdot s_2$, and both $\Theta;\Gamma;\Delta,\alpha^{A\supset B} \vdash M_1^{C\supset D} \mid s_1$ and $\Theta;\Gamma;\Delta,\alpha^{A\supset B} \vdash M_2^C \mid s_2$ are derivable by hypothesis. By IH, we can derive $\Theta;\Gamma;\Delta' \vdash M_1(\!|\,[\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)N^A\,|\!)^{C\supset D} \mid s_1(\!|\,[\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)t\,|\!)$ and $\Theta;\Gamma;\Delta' \vdash M_2(\!|\,[\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)N^A\,|\!)^C \mid s_2(\!|\,[\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)t\,|\!)$, from which we obtain $\Theta;\Gamma;\Delta' \vdash M(\!|\,[\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)N^A\,|\!)^D \mid s(\!|\,[\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)t\,|\!)$ by T-$\supset$E.

- T-$\Box$I: $D = [\![r]\!]C$, $M = (!L)^{[\![r]\!]C}$ and $s = !r$. Since $M(\!|\,[\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)N^A\,|\!) = M$ and $s(\!|\,[\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)t\,|\!) = s$, then $\Theta;\Gamma;\Delta \vdash M(\!|\,[\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)N^A\,|\!)^B \mid$ $s(\!|\,[\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)t\,|\!)$ is already derivable by hypothesis.

- T-$\Box$E: $M = (M_1^D\langle v^C{:=}r, M_2\rangle)^{D\{v^C \leftarrow r\}}$, $D = D\{v^C \leftarrow r\}$, $s = s_1\langle v^C{:=}r, s_2\rangle$ and the judgments $\Theta;\Gamma;\Delta,\alpha^{A\supset B} \vdash M_2^{[\![r]\!]C} \mid s_1$ and $\Theta,v^C;\Gamma;\Delta,\alpha^{A\supset B} \vdash M_1^D \mid s_2$ are both derivable. By induction hypothesis, we can derive both $\Theta;\Gamma;\Delta' \vdash M_2^{[\![r]\!]C}(\!|\,[\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)N^A\,|\!)^{[\![r]\!]C} \mid s_1(\!|\,[\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)t\,|\!)$ and $\Theta,v^C;\Gamma;\Delta' \vdash M_1^D(\!|\,[\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)N^A\,|\!)^D \mid s_2(\!|\,[\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)t\,|\!)$. The result follows by T-$\Box$E.

- T-Name: $M = ([\gamma^C]M')^{\perp}$, $s = [\gamma^C]s'$, $D = \perp$, $\Delta = \Delta_1,\gamma^C$ and the judgment $\Theta;\Gamma;\Delta_1,\gamma^C,\alpha^{A\supset B} \vdash M'^C \mid s'$ is derivable by hypothesis. There are two possibilities:

– $\gamma^C = \alpha^{A \supset B}$: here $C = A \supset B$, $M(\!| [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)N^A |\!) = ([\beta^B]M'N^A)^\perp$ and $s(\!| [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)t |\!) = [\beta^B]s' \cdot t$. Since $\Theta; \Gamma; \Delta \vdash N^A \mid t$ is derivable by hypothesis, then by Weakening so is $\Theta; \Gamma; \Delta, \beta^B \vdash N^A \mid t$. We can thus construct the following derivation:

$$\dfrac{\dfrac{\Theta; \Gamma; \Delta, \beta^B \vdash M'^{A\supset B} \mid s' \quad \Theta; \Gamma; \Delta, \beta^B \vdash N^A \mid t}{\Theta; \Gamma; \Delta, \beta^B \vdash (M'N^A)^B \mid s' \cdot t}\,\mathsf{T\text{-}\supset E}}{\Theta; \Gamma; \Delta, \beta^B \vdash [\beta^B](M'N^A)^\perp \mid [\beta^B]s' \cdot t}\,\mathsf{T\text{-}Name}$$

– $\gamma^C \neq \alpha^{A \supset B}$: here $M(\!| [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)N^A |\!) = [\gamma^C](M'(\!| [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)N^A |\!))$ and $s(\!| [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)t |\!) = [\gamma^C](s'(\!| [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)t |\!))$. The result follows by IH and T-Name.

- T-NAbs: $M = (\mu\gamma^D.M')^D$, $s = \mu\gamma^D.s'$ and $\Theta; \Gamma; \Delta, \alpha^{A\supset B}, \gamma^D \vdash M'^\perp \mid s'$ is derivable by hypothesis. There are two cases to consider:
  – $\gamma^D = \alpha^{A\supset B}$: here $M(\!| [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)N^A |\!) = M$ and $s(\!| [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)t |\!)$ $= s$. Since $\Theta; \Gamma; \Delta, \alpha^{A\supset B} \vdash M^D \mid s$ is derivable and $\alpha^{A\supset B}$ is not free in either M or s, then, by Strengthening and Weakening, we can derive $\Theta; \Gamma; \Delta, \beta^B \vdash M^D \mid s$.
  – $\gamma^D \neq \alpha^{A\supset B}$: $M(\!| [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)N^A |\!) = (\mu\gamma^D.M'(\!| [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)N^A |\!))^D$ and $s(\!| [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)t |\!) = \mu\gamma^D.(s'(\!| [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)t |\!))$. By IH we can derive $\Theta; \Gamma; \Delta, \beta^B, \gamma^D \vdash M'(\!| [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)N^A |\!)^\perp$ $\mid s'(\!| [\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)t |\!)$.
  The result follows by T-NAbs.
- T-PlusL: $M = (M_1 \uplus_L s_2)^D$, $s = s_1 + s_2$ and $\Theta; \Gamma; \Delta, \alpha^{A\supset B} \vdash M_1^D \mid s_1$ is derivable by hypothesis. The result follows by IH and T-PlusL.
- T-PlusR: this case is analogous to the previous one.

□

B.3 Metatheoretical Results for λ^{LP} with Additional Rules

This section extends the proofs shown in Section 4.4 with the additional rules presented in Section 4.7.1.

Proof of Lemma 4.4.19

Proof.- Additional cases:

- υ_β: $M = M_1^{A\supset B}\langle v^C{:=}r, M_2^{[\![r]\!]C}\rangle N_1^A$ and $N = (M_1^{A\supset B}N_1^A)\langle v^C{:=}r, M_2^{[\![r]\!]C}\rangle$. By Inversion Lemma (3 times), $D = B\{v^C \leftarrow r\}$, $s = s_1\langle v^C{:=}r, s_2\rangle \cdot t$ and the judgements $\Theta, v^C; \Gamma; \Delta \vdash M_1^{A\supset B} \mid s_1$, $\Theta; \Gamma; \Delta \vdash M_2^{[\![r]\!]C} \mid s_2$ and $\Theta; \Gamma; \Delta \vdash N_1^A \mid t$ are derivable. Note that, by variable convention, $v^C \notin \mathsf{FVV}(N^A) \cup \mathsf{FVV}(A)$. The result holds by $\supset$E (using Weakening), □E and Eq-υ_β.
- υ_γ: $M = M_1^A\langle v^B{:=}r, N_1^{[\![r]\!]B}\langle u^C{:=}s, N_2^{[\![s]\!]C}\rangle\rangle$ and $N = M_1^A\langle v^B{:=}r, N_1^{[\![r]\!]B}\rangle\langle u^C{:=}s, N_2^{[\![s]\!]C}\rangle$. By Inversion Lemma (3 times), $D = A\{v^B \leftarrow r\}$, $s = t_1\langle v^B{:=}r, t_2\rangle\langle u^C{:=}s_1, t_3\rangle$ and the judgements $\Theta, v^B, u^C; \Gamma; \Delta \vdash M_1^A \mid t_1$, $\Theta, u^C; \Gamma; \Delta \vdash N_1^{[\![r]\!]B} \mid t_2$ and $\Theta; \Gamma; \Delta \vdash N_2^{[\![s_1]\!]C} \mid t_3$ are derivable. Note that, by variable convention, $u^C \notin \mathsf{FVV}(A) \cup \mathsf{FVV}([\![r]\!]B)$. The result holds by □E twice and Eq-υ_γ.

- υ_μ: $M = [\beta^B](M_1^B\langle v^A{:=}r, M_2^{[\![r]\!]A}\rangle)$ and $N = ([\beta^B]M_1^B)\langle v^A{:=}r, M_2^{[\![r]\!]A}\rangle$. By Inversion Lemma (twice), $D = \bot$, $s = [\beta^B](s_1\langle v^A{:=}r, t\rangle)$, $\Delta = \Delta', \beta^B$ and the judgements $\Theta, v^A; \Gamma; \Delta', \beta^B \vdash M_1^B \,|\, s_1$ and $\Theta; \Gamma; \Delta', \beta^B \vdash M_2^{[\![r]\!]A} \,|\, t$ are derivable. By variable convention, $v^A \notin \mathsf{FVV}(B)$. The result holds by Name, $\Box\mathsf{E}$ and Eq-$\upsilon\mu$.
- υ_θ: $M = \mu\alpha^A.(M_1^\bot\langle v^B{:=}r, M_2^{[\![r]\!]B}\rangle)$, $N = (\mu\alpha^A.M_1^\bot)\langle v^B{:=}r, M_2^{[\![r]\!]B}\rangle$ and $\alpha^A \notin \mathsf{FVF}(M_2^{[\![r]\!]B})$. By Inversion Lemma (twice), $s = \mu\alpha^A.(s_1\langle v^B{:=}r, t\rangle)$, $D = A$ and the judgements $\Theta, v^B; \Gamma; \Delta, \alpha^A \vdash M_1^\bot \,|\, s_1$ and $\Theta; \Gamma; \Delta \vdash M_2^{[\![r]\!]B} \,|\, t$ are derivable. By variable convention, $v^B \notin \mathsf{FVV}(A)$. The result holds by NAbs, $\Box\mathsf{E}$ and Eq-$\upsilon\theta$.

$\Box$

Proof of Lemma 4.2.15

Proof.- Additional cases:

- Eq-υ_β: $s = s_1\langle v^C{:=}r, s_2\rangle \cdot t'$, $t = (s_1 \cdot t')\langle v^C{:=}r, s_2\rangle$, $B = D\{v^C \leftarrow r\}$ and the judgements $\Theta, v^C; \Gamma; \Delta \vdash A \supset D \,|\, s_1$, $\Theta; \Gamma; \Delta \vdash [\![r]\!]C \,|\, s_2$ and $\Theta; \Gamma; \Delta \vdash A \,|\, t'$ are derivable by hypothesis. By Weakening, $\Theta, v^C; \Gamma; \Delta \vdash A \,|\, t'$ is also derivable. We can derive s by $\Box\mathsf{E}$ and $\supset\mathsf{E}$, and t by $\supset\mathsf{E}$ and $\Box\mathsf{E}$.
- Eq-υ_γ: analogous to the previous case, using $\Box\mathsf{E}$ twice in a different order on either side.
- Eq-υ_μ: analogous to previous cases, using $\Box\mathsf{E}$ and Name for s, and conversely Name and $\Box\mathsf{E}$ for t.
- Eq-υ_θ: analogous to previous cases, using $\Box\mathsf{E}$ and NAbs for s, and conversely NAbs and $\Box\mathsf{E}$ for t.

$\Box$

B.3.1 Confluence of HLP with additional rules

This section depicts (Fig. B.1 to Fig. B.3) how the critical pairs that arise from the additional permutation rules (Sec. 4.7.1) may be joined.

μ-υ_θ Critical pair:

$$\begin{tikzcd}
 & {[\beta^A]\mu\alpha^A.(M^\perp\langle v^B{:=}r, N^{[\![r]\!]B}\rangle)} \arrow[dl] \arrow[dr] & \\
M^\perp\{\alpha^A \leftarrow \beta^A\}\langle v^B{:=}r, N^{[\![r]\!]B}\rangle & & {[\beta^A]((\mu\alpha^A.M^\perp)\langle v^B{:=}r, N^{[\![r]\!]B}\rangle)} \arrow[dl] \\
 & {([\beta^A]\mu\alpha^A.M^\perp)\langle v^B{:=}r, N^{[\![r]\!]B}\rangle} \arrow[ul] &
\end{tikzcd}$$

ζ-υ_θ Critical pair:

$$\begin{tikzcd}
 & {(\mu\alpha^{A\supset B}.M_1^\perp\langle v^C{:=}r, N^{[\![r]\!]C}\rangle)M_2^A} \arrow[dl] \arrow[dr] & \\
{\mu\beta^B.(M_1^\perp(\!|\,[\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)M_2^A\,|\!)\langle v^C{:=}r, N^{[\![r]\!]C}\rangle)} \arrow[d] & & {(\mu\alpha^{A\supset B}.M_1^\perp)\langle v^C{:=}r, N^{[\![r]\!]C}\rangle M_2^A} \arrow[d] \\
{(\mu\beta^B.M_1^\perp(\!|\,[\alpha^{A\supset B}](\bullet) \leftarrow [\beta^B](\bullet)M_2^A\,|\!))\langle v^C{:=}r, N^{[\![r]\!]C}\rangle} \arrow[r, leftarrow] & {((\mu\alpha^{A\supset B}.M_1^\perp)M_2^A)\langle v^C{:=}r, N^{[\![r]\!]C}\rangle} &
\end{tikzcd}$$

υ_β-γ Critical pair:

$$\begin{tikzcd}
 & {(M_1^{A\supset B}\langle v^C{:=}r, !N^C\rangle)M_2^A} \arrow[dl] \arrow[dr] & \\
{(M_1^{A\supset B}M_2^A)\langle v^C{:=}r, !N^C\rangle} \arrow[rr, "\rightharpoonup"] & & {M_1^{A\supset B}\{v^C \leftarrow N^C, r\}M_2^A}
\end{tikzcd}$$

υ_β-ϕ_L Critical pair:

$$\begin{tikzcd}
 & {(M_1^{A\supset B}\langle v^C{:=}r, N^{[\![r]\!]C} +_{\mathrm{L}} t\rangle)M_2^A} \arrow[dl] \arrow[dr] & \\
{(M_1^{A\supset B}M_2^A)\langle v^C{:=}r, N^{[\![r]\!]C} +_{\mathrm{L}} t\rangle} \arrow[d] & & {(M_1^{A\supset B}\langle v^C{:=}r, N^{[\![r]\!]C}\rangle +_{\mathrm{L}} t)M_2^A} \arrow[d] \\
{(M_1^{A\supset B}M_2^A)\langle v^C{:=}r, N^{[\![r]\!]C}\rangle +_{\mathrm{L}} t} & {(M_1^{A\supset B}\langle v^C{:=}r, N^{[\![r]\!]C}\rangle M_2^A) +_{\mathrm{L}} t} \arrow[l]
\end{tikzcd}$$

Additional critical pairs associated to permutative reductions (1/3)

υ_β-υ_γ Critical pair:

$$
\begin{array}{ccc}
 & (M_1^{A\supset B}\langle v^C{:=}r, M_2^{[\![r]\!]C}\langle u^F{:=}s, M_3^{[\![s]\!]F}\rangle\rangle)N^A & \\
\swarrow & & \searrow \\
(M_1^{A\supset B}N^A)\langle v^C{:=}r, M_2^{[\![r]\!]C}\langle u^F{:=}s, M_3^{[\![s]\!]F}\rangle\rangle & & M_1^{A\supset B}\langle v^C{:=}r, M_2^{[\![r]\!]C}\rangle\langle u^F{:=}s, M_3^{[\![s]\!]F}\rangle N^A \\
\downarrow & & \downarrow \\
(M_1^{A\supset B}N^A)\langle v^C{:=}r, M_2^{[\![r]\!]C}\rangle\langle u^F{:=}s, M_3^{[\![s]\!]F}\rangle & \leftarrow & (M_1^{A\supset B}\langle v^C{:=}r, M_2^{[\![r]\!]C}\rangle)N^A\langle u^F{:=}s, M_3^{[\![s]\!]F}\rangle
\end{array}
$$

υ_γ-γ Critical pair:

$$
\begin{array}{ccc}
 & M_1^A\langle v^B{:=}r, M_2^{[\![r]\!]B}\langle u^C{:=}s, !N^C\rangle\rangle & \\
\swarrow & & \searrow \\
M_1^A\langle v^B{:=}r, M_2^{[\![r]\!]B}\rangle\langle u^C{:=}s, !N^C\rangle & \rightarrow & M_1^A\langle v^B{:=}r, M_2^{[\![r]\!]B}\{U^C \leftarrow N^C, s\}\rangle
\end{array}
$$

υ_γ-ϕ_L Critical pair:

$$
\begin{array}{ccc}
 & M_1^A\langle v^B{:=}r, M_2^{[\![r]\!]B}\langle u^C{:=}s, M_3^{[\![s]\!]C}+_{\mathrm{L}}t\rangle\rangle & \\
\swarrow & & \searrow \\
M_1^A\langle v^B{:=}r, M_2^{[\![r]\!]B}\rangle\langle u^C{:=}s, M_3^{[\![s]\!]C}+_{\mathrm{L}}t\rangle & & M_1^A\langle v^B{:=}r, M_2^{[\![r]\!]B}\langle u^C{:=}s, M_3^{[\![s]\!]C}\rangle+_{\mathrm{L}}t\rangle \\
\downarrow & & \downarrow \\
M_1^A\langle v^B{:=}r, M_2^{[\![r]\!]B}\rangle\langle u^C{:=}s, M_3^{[\![s]\!]C}\rangle+_{\mathrm{L}}t & \leftarrow & M_1^A\langle v^B{:=}r, M_2^{[\![r]\!]B}\langle u^C{:=}s, M_3^{[\![s]\!]C}\rangle\rangle+_{\mathrm{L}}t
\end{array}
$$

υ_γ-υ_γ Critical pair:

$$
\begin{array}{ccc}
 & M_1^A\langle v^B{:=}r, M_2^{[\![r]\!]B}\langle u^C{:=}s, M_3^{[\![s]\!]C}\langle w^F{:=}t, M_4^{[\![w]\!]F}\rangle\rangle\rangle & \\
\swarrow & & \searrow \\
M_1\langle v^B{:=}r, M_2\rangle\langle u^C{:=}s, M_3\langle w^F{:=}t, M_4\rangle\rangle & & M_1\langle v^B{:=}r, M_2\langle u^C{:=}s, M_3\rangle\langle w^F{:=}t, M_4\rangle\rangle \\
\downarrow & & \downarrow \\
M_1\langle v^B{:=}r, M_2\rangle\langle u^C{:=}s, M_3\rangle\langle w^F{:=}t, M_4\rangle & \leftarrow & M_1\langle v^B{:=}r, M_2\langle u^C{:=}s, M_3\rangle\rangle\langle w^F{:=}t, M_4\rangle
\end{array}
$$

υ_μ-γ Critical pair:

$$
\begin{array}{ccc}
 & [\alpha^A](M^A\langle v^B{:=}r, !N^B\rangle) & \\
\swarrow & & \searrow \\
([\alpha^A]M^A)\langle v^B{:=}r, !N^B\rangle & \rightarrow & [\alpha^A]M^A\{v^B \leftarrow r, N^B\}
\end{array}
$$

υ_μ-ϕ_L Critical pair:

$$
\begin{array}{ccc}
 & [\alpha^A](M^A\langle v^B{:=}r, N^{[\![r]\!]B}+_{\mathrm{L}}t\rangle) & \\
\swarrow & & \searrow \\
([\alpha^A]M^A)\langle v^B{:=}r, N^{[\![r]\!]B}+_{\mathrm{L}}t\rangle & & [\alpha^A](M^A\langle v^B{:=}r, N^{[\![r]\!]B}\rangle+_{\mathrm{L}}t) \\
\downarrow & & \downarrow \\
([\alpha^A]M^A)\langle v^B{:=}r, N^{[\![r]\!]B}\rangle+_{\mathrm{L}}t & \leftarrow & ([\alpha^A](M^A\langle v^B{:=}r, N^{[\![r]\!]B}\rangle))+_{\mathrm{L}}t
\end{array}
$$

Additional critical pairs associated to permutative reductions (2/3)

υ_μ-υ_γ Critical pair:

$$
\begin{tikzcd}
 & [\alpha^A](M_1^A\langle v^B{:=}r, M_2^{[\![r]\!]B}\langle u^C{:=}s, M_3^{[\![s]\!]C}\rangle\rangle) \arrow[dl] \arrow[dr] & \\
([\alpha^A]M_1^A)\langle v^B{:=}r, M_2\langle u^C{:=}s, M_3\rangle\rangle \arrow[d] & & [\alpha^A](M_1^A\langle v^B{:=}r, M_2\rangle\langle u^C{:=}s, M_3\rangle) \arrow[d] \\
([\alpha^A]M_1^A)\langle v^B{:=}r, M_2\rangle\langle u^C{:=}s, M_3\rangle & & ([\alpha^A](M_1^A\langle v^B{:=}r, M_2\rangle))\langle u^C{:=}s, M_3\rangle \arrow[ll]
\end{tikzcd}
$$

υ_θ-γ Critical pair:

$$
\begin{tikzcd}
 & \mu\alpha^A.(M^A\langle v^B{:=}r, !N^B\rangle) \arrow[dl] \arrow[dr] & \\
(\mu\alpha^A.M^A)\langle v^B{:=}r, !N^B\rangle \arrow[rr] & & \mu\alpha^A.M^A\{v^B \leftarrow r, N^B\}
\end{tikzcd}
$$

υ_θ-ϕ_L Critical pair:

$$
\begin{tikzcd}
 & \mu\alpha^A.(M^A\langle v^B{:=}r, N^{[\![r]\!]B}+_{\mathrm{L}} t\rangle) \arrow[dl] \arrow[dr] & \\
(\mu\alpha^A.M^A)\langle v^B{:=}r, N^{[\![r]\!]B}+_{\mathrm{L}} t\rangle \arrow[d] & & \mu\alpha^A.(M^A\langle v^B{:=}r, N^{[\![r]\!]B}\rangle+_{\mathrm{L}} t) \arrow[d] \\
(\mu\alpha^A.M^A)\langle v^B{:=}r, N^{[\![r]\!]B}\rangle+_{\mathrm{L}} t & & (\mu\alpha^A.(M^A\langle v^B{:=}r, N^{[\![r]\!]B}\rangle))+_{\mathrm{L}} t \arrow[ll]
\end{tikzcd}
$$

υ_θ-υ_γ Critical pair:

$$
\begin{tikzcd}
 & \mu\alpha^A.(M_1^A\langle v^B{:=}r, M_2^{[\![r]\!]B}\langle u^C{:=}s, M_3^{[\![s]\!]C}\rangle\rangle) \arrow[dl] \arrow[dr] & \\
(\mu\alpha^A.M_1^A)\langle v^B{:=}r, M_2\langle u^C{:=}s, M_3\rangle\rangle \arrow[d] & & \mu\alpha^A.(M_1^A\langle v^B{:=}r, M_2\rangle\langle u^C{:=}s, M_3\rangle) \arrow[d] \\
(\mu\alpha^A.M_1^A)\langle v^B{:=}r, M_2\rangle\langle u^C{:=}s, M_3\rangle & & (\mu\alpha^A.(M_1^A\langle v^B{:=}r, M_2\rangle))\langle u^C{:=}s, M_3\rangle \arrow[ll]
\end{tikzcd}
$$

Additional critical pairs associated to permutative reductions (3/3)

Bibliography

[AB05] Sergei N. Artëmov and Lev D. Beklemishev. Provability logic. In *Handbook of Philosophical Logic, 2nd ed*, number 13, pages 189–360. Springer, 2005.

[AB07] Sergei N. Artëmov and Eduardo Bonelli. The intensional lambda calculus. In *LFCS*, pages 12–25, 2007.

[AF03] Martin Abadi and Cedric Fournet. Access control based on execution history. In *In Proceedings of the 10th Annual Network and Distributed System Security Symposium*, pages 107–121, 2003.

[AI07] Sergei N. Artëmov and Rosalie Iemhoff. The basic intuitionistic logic of proofs. *Journal of Symbolic Logic*, 72(2):439–451, 2007.

[AMR06] Ariel Arbiser, Alexandre Miquel, and Alejandro Ríos. A lambda-calculus with constructors. In Frank Pfenning, editor, *Term Rewriting and Applications*, volume 4098 of *Lecture Notes in Computer Science*, pages 181–196. Springer Berlin Heidelberg, 2006.

[Art95] Sergei N. Artëmov. Operational modal logic. Technical Report Technical Report MSI 95-29, Cornell University, 1995.

[Art01] Sergei N. Artëmov. Explicit provability and constructive semantics. *Bulletin of Symbolic Logic*, 7(1):1–36, 2001.

[Art08a] Sergei N. Artëmov. Justification logic. In *Logics in Artificial Intelligence*, pages 1–4. Springer, 2008.

[Art08b] Sergei N. Artëmov. The logic of justification. *The Review of Symbolic Logic*, 1(04):477–513, 2008.

[Aug84] Lennart Augustsson. A compiler for lazy ml. In *Proceedings of the 1984 ACM Symposium on LISP and functional programming*, pages 218–227. ACM, 1984.

[AY01] Sergei N. Artëmov and Tatiana Yavorskaya (Sidon). On first order logic of proofs. *Moscow Mathematical Journal*, 1(4):475–490, October-December 2001.

[AY11] Sergei N. Artëmov and Tatiana Yavorskaya (Sidon). First-order logic of proofs. Technical report, Technical Report TR–2011005, CUNY Ph. D. Program in Computer Science, 2011.

[Bar84] Henk P. Barendregt. *The Lambda Calculus: Its Syntax and Semantics*. Sole Distributors for the U.S.A. And Canada, Elsevier Science Pub. Co., 1984.

[Bar92] Henk P. Barendregt. Handbook of logic in computer science (vol. 2). chapter Lambda Calculi with Types, pages 117–309. Oxford University Press, Inc., New York, NY, USA, 1992.

[BB10] Francisco Bavera and Eduardo Bonelli. Justification logic and history based computation. In *ICTAC*, pages 337–351, 2010.

[BB14] Francisco Bavera and Eduardo Bonelli. Justification logic and audited computation. To appear in Journal of Logic and Computation, 2014.

[BF12] Eduardo Bonelli and Federico Feller. Justification logic as a foundation for certifying mobile computation. *Ann. Pure Appl. Logic*, 163(7):935–950, 2012.

[BG66] Corrado Böhm and Wolf Gross. Introduction to the cuch. *Automata Theory*, pages 35–65, 1966.

[BKdV03] Mark Bezem, Jan Willem Klop, and Roel de Vrijer. *Term Rewriting Systems*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2003.

[BN98] Franz Baader and Tobias Nipkow. *Term Rewriting and All That*. Cambridge University Press, New York, NY, USA, 1998.

[BN05] Anindya Banerjee and David A. Naumann. History-based access control and secure information flow. In *Construction and Analysis of Safe, Secure, and Interoperable Smart Devices, International Workshop (CASSIS 2004), Revised Selected Papers, volume 3362 of Lecture Notes in Computer Science*, pages 27–48. Springer-Verlag, 2005.

[BPJR88] Geoffrey L Burn, Simon L. Peyton Jones, and John D. Robson. The spineless g-machine. In *Proceedings of the 1988 ACM conference on LISP and functional programming*, pages 244–258. ACM, 1988.

[BR14] Alexander Bates and Reuben N.S. Rowe. Structural types for the factorisation calculus. Master's thesis, University College London, 2014.

[Bra02] Julian C. Bradfield, editor. *Computer Science Logic, 16th International Workshop, CSL 2002, 11th Annual Conference of the EACSL, Edinburgh, Scotland, UK, September 22-25, 2002, Proceedings*, volume 2471 of *Lecture Notes in Computer Science*. Springer, 2002.

[Bro07] Luitzen Egbertus Jan Brouwer. On the foundations of mathematics. *Collected Works*, 1:11–101, 1907.

[BS14] Eduardo Bonelli and Gabriela Steren. Hypothetical logic of proofs. *Logica Universalis*, pages 1–38, 2014.

[CCM87] Guy Cousineau, Pierre-Louis Curien, and Michel Mauny. The categorical abstract machine. *Science of computer programming*, 8(2):173–202, 1987.

[CF58] Haskell Brooks Curry and Robert Feys. *Combinatory Logic*. Number v. 1 in Combinatory Logic. North-Holland Publishing Company, 1958.

[ÇH98] Naim Çağman and J. Roger Hindley. Combinatory weak reduction in lambda calculus. *Theoretical Computer Science*, 198(1):239–247, 1998.

[Chu36] Alonzo Church. A note on the entscheidungsproblem. *Journal of Symbolic Logic*, 1(01):40–41, 1936.

[Chu40] Alonzo Church. A formulation of the simple theory of types. *Journal of Symbolic Logic*, 5(2):56–68, 1940.

[Chu41] Alonzo Church. *The Calculi of Lambda Conversion*. Princeton University Press, Princeton, NJ, USA, 1941.

[CK01] Horatiu Cirstea and Claude Kirchner. The rewriting calculus — Part I. *Logic Journal of the Interest Group in Pure and Applied Logics*, 9(3):427–498, May 2001.

[CSH72] Haskell Brooks Curry, Jonathan P. Seldin, and J. Roger Hindley. *Combinatory Logic, 2*. Number v. 2 in Studies in logic and the foundations of mathematics. North-Holland, 1972.

[Cur30] Haskell Brooks Curry. *Grundlagen der kombinatorischen Logik (Foundations of Combinatory logic)*. PhD thesis, University of Göttingen, 1930.

[Cur34] Haskell Brooks Curry. Functionality in combinatory logic. *Proceedings of the National Academy of Sciences*, 20(11):584–590, 1934.

[Cur52] Haskell Brooks Curry. The logic of program composition. In *Applications Scientifiques de la Logique Mathématique, Actes du Deuxieme Colloque International de Logique Mathématique, Paris*, pages 97–102, 1952.

[Cur69] Haskell Brooks Curry. Modified basic functionality in combinatory logic. *Dialectica*, 23(2):83–92, 1969.

[Cur86] Pierre-Louis Curien. Categorical combinators, sequential algorithms, and functional programming, 1986.

[Cur93] Pierre-Louis Curien. *Categorical combinators, sequential algorithms, and functional programming*, volume 9. Springer, 1993.

[Das11] Evgenij Dashkov. Arithmetical completeness of the intuitionistic logic of proofs. *Journal of Logic and Computation*, 21(4):665–682, 2011.

[dG01] Philippe de Groote. Strong normalization of classical natural deduction with disjunction. In *Typed Lambda Calculi and Applications*, pages 182–196. Springer, 2001.

[DO90] Nachum Dershowitz and Mitsuhiro Okada. A rationale for conditional equational programming. *Theoretical Computer Science*, 75(1-2):111–138, September 1990.

[Dou93] Daniel J. Dougherty. Higher-order unification via combinators. *Theoretical Computer Science*, 114:273–298, 1993.

[DP96] Rowan Davies and Frank Pfenning. A modal analysis of staged computation. In *POPL*, pages 258–270, 1996.

[DP01a] Rowan Davies and Frank Pfenning. A judgmental reconstruction of modal logic. *Mathematical Structures in Computer Science*, 11(4):511–540, 2001.

[DP01b] Rowan Davies and Frank Pfenning. A modal analysis of staged computation. *Journal of the ACM*, 48(3):555–604, 2001.

[Dra68] Al'bert Grigor'evich Dragalin. A computability of primitive recursive terms of finite type and the primitive recursive realization. *Zapiski Nauchnykh Seminarov POMI*, 8:32–45, 1968.

[Fit58] Frederic B. Fitch. Representation of sequential circuits in combinatory logic. *Philosophy of science*, pages 263–279, 1958.

[Fit05a] Melvin Fitting. The logic of proofs, semantically. *Annals of Pure and Applied Logic*, 132(1):1–25, 2005.

[Fit05b] Melvin Fitting. A quantified logic of evidence. *WoLLIC 2005 Proceedings, Electronic Notes in Theoretical Computer Science, Amsterdam: Elsevier*, pages 59–70, 2005.

[Fit14] Melvin Fitting. Possible world semantics for first-order logic of proofs. *Ann. Pure Appl. Logic*, 165(1):225–240, 2014.

[Gan80] Robin O. Gandy. Proofs of strong normalization. *To HB Curry: essays on combinatory logic, lambda calculus and formalism*, pages 457–477, 1980.

[Gen35] Gerhard Gentzen. Untersuchungen über das logische schließen. i. *Mathematische zeitschrift*, 39(1):176–210, 1935.

[GJ02] Herman Geuvers and Gueorgui I. Jojgov. Open proofs and open terms: A basis for interactive logic. In Bradfield [Bra02], pages 537–552.

[Göd33] Kurt Gödel. Eine interpretation des intuitionistischen aussagenkalküls. *Ergebnisse eines mathematischen Kolloquiums*, 4:39–40, 1933.

[Gri90] Timothy G. Griffin. A formulae-as-type notion of control. In *Proceedings of the 17th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '90, pages 47–58, New York, NY, USA, 1990. ACM.

[Han97] Michael Hanus. A unified computation model for functional and logic programming. In *POPL*, pages 80–93, 1997.

[Hey30] Arend Heyting. *Die formalen Regeln der intuitionistischen Logik*. Deütsche Akademie der Wissenschaften zu Berlin, Mathematisch-Naturwissenschaftliche Klasse, 1930.

[Hin67] Shigeru Hinata. Calculability of primitive recursive functionals of finite type. *Science Reports of the Tokyo Kyoiku Daigaku*, A(9):218–235, 1967.

[HK84] Paul Hudak and David Kranz. A combinator-based compiler for a functional language. In *Proceedings of the 11th ACM SIGACT-SIGPLAN symposium on Principles of programming languages*, pages 122–132. ACM, 1984.

[HM66] Yoshito Hanatani and Shoji Maehara. Calculabilité des fonctionnels recursives primitives de type fini sur les nombres naturels. *Annals of the Japan Association for the Philosophy of Science*, 3:19–30, 1966.

[Hon06] Furio Honsell. A framework for defining logical frameworks. In *University of Udine*, 2006.

[How95] William A Howard. The formulae-as-types notion of construction, 1995.

[HS86] J. Roger Hindley and Jonathan P. Seldin. *Introduction to combinators and the lambda-calculus*. Cambridge University Press, 1986.

[Hue80] Gérard P. Huet. Confluent reductions: Abstract properties and applications to term rewriting systems: Abstract properties and applications to term rewriting systems. *Journal of the ACM*, 27(4):797–821, October 1980.

[Jay01] C Barry Jay. Distinguishing data structures and functions: the constructor calculus and functorial types. In *Typed Lambda Calculi and Applications*, pages 217–239. Springer, 2001.

[Jay06] C. Barry Jay. Typing first-class patterns. In *In Higher-Order Rewriting, proceedings*, 2006.

[Jay09] C. Barry Jay. *Pattern Calculus: Computing with Functions and Structures*. Springer, 2009.

[JGW11] C. Barry Jay and Thomas Given-Wilson. A combinatory account of internal structure. *Journal of Symbolic Logic*, 76(3):807–826, 09 2011.

[JK05] C. Barry Jay and Delia Kesner. Pure pattern calculus. In *ACM Transactions on Programming Languages and Systems (TOPLAS)*, pages 262–274, 2005.

[JK09] C. Barry Jay and Delia Kesner. First-class patterns. *J. Funct. Program.*, 19(2):191–225, March 2009.

[Joh83] Thomas Johnsson. The g-machine: An abstract machine for graph reduction. In *Declarative Programming Workshop*, pages 1–19, 1983.

[K+36] Stephen Cole Kleene et al. λ-definability and recursiveness. *Duke Mathematical Journal*, 2(2):340–353, 1936.

[KB70] Donald E. Knuth and Peter Bendix. Simple word problems in universal algebras. In John Leech, editor, *Computational Problems in Abstract Algebra*, pages 263 – 297. Pergamon, 1970.

[Kie85] Richard B. Kieburtz. The g-machine: A fast, graph-reduction evaluator. In *Functional Programming Languages and Computer Architecture*, pages 400–413. Springer, 1985.

[Klo76] Jan Willem Klop. A counterexample to the cr property for λ-calculus+ dmm m. *Typed note, Utrecht University*, 1976.

[Kra12] Simon Kramer. A logic of interactive proofs (formal theory of knowledge transfer). *arXiv preprint arXiv:1201.3667*, 2012.

[Kuz09] Roman Kuznets. A note on the use of sum in the logic of proofs. Proceedings of the 7th Panhellenic Logic Symposium, Patras University, Greece, July 1519, 2009, pages 99–103. Patras University Press, Costas Drossos, Pavlos Peppas, and Constantine Tsinakis, editors, 2009.

[KvOdV08] Jan Willem Klop, Vincent van Oostrom, and Roel de Vrijer. Lambda calculus with patterns. *Theoretical Computer Science*, 398(13):16 – 31, 2008. Calculi, Types and Applications: Essays in honour of M. Coppo, M. Dezani-Ciancaglini and S. Ronchi Della Rocca.

[Lév78] Jean-Jacques Lévy. *Réductions correctes et optimales dans le lambda-calcul.* PhD thesis, Paris 7, 1978.

[Mkr97] Alexey Mkrtychev. Models for the logic of proofs. In *Logical foundations of computer science*, pages 266–275. Springer, 1997.

[ML84] Per Martin-Löf. *An intuitionistic theory of types.* Bibliopolis, 1984.

[MT05] Dale Miller and Alwen Tiu. A proof theory for generic judgments. *ACM Transactions on Computational Logic (TOCL)*, 6(4):749–783, 2005.

[New42] Maxwell Herman Alexander Newman. On theories with a combinatorial definition of "equivalence". *Annals of Mathematics*, 43(2):223–243, 1942.

[Nip95] Tobias Nipkow. Higher-order rewrite systems. In *Rewriting Techniques and Applications*, pages 256–256. Springer, 1995.

[NPP08] Aleksandar Nanevski, Frank Pfenning, and Brigitte Pientka. Contextual modal type theory. *Transactions on Computational Logic (TOCL)*, 9(3), 2008.

[Orl25] Ivan Efimovich Orlov. The logic of compatibility of propositions. *Matematicheskii Sbornik*, 35(3-4):263–286, 1925.

[Par92] Michel Parigot. $\lambda\mu$-calculus: An algorithmic interpretation of classical natural deduction. In Andrei Voronkov, editor, *Logic Programming and Automated Reasoning*, volume 624 of *Lecture Notes in Computer Science*, pages 190–201. Springer Berlin Heidelberg, 1992.

[Par97] Michel Parigot. Proofs of strong normalisation for second order classical natural deduction. *The Journal of Symbolic Logic*, 62(4):pp. 1461–1479, 1997.

[Pie10] Brigitte Pientka. Beluga: programming with dependent types, contextual data, and contexts. In *Functional and Logic Programming*, pages 1–12. Springer, 2010.

[PJHH+93] Simon L. Peyton Jones, Cordy Hall, Kevin Hammond, Will Partain, and Philip Wadler. The glasgow haskell compiler: a technical overview. In *Proc. UK Joint Framework for Information Technology (JFIT) Technical Conference*, volume 93. Citeseer, 1993.

[PJS89] Simon L. Peyton Jones and Jon Salkild. The spineless tagless g-machine. In *Proceedings of the fourth international conference on Functional programming languages and computer architecture*, pages 184–201. ACM, 1989.

[Pra65] Dag Prawitz. Natural deduction: A proof theoretical study. *Almqvist and Wiksell, Stockholm*, 1965.

[Ros35] John Barkley Rosser. A mathematical logic without variables. i. *Annals of mathematics*, pages 127–150, 1935.

[Sau10] Alexis Saurin. Typing streams in the $\lambda\mu$-calculus. *ACM Transactions on Computational Logic (TOCL)*, 11(4):28, 2010.

[SB13] Gabriela Steren and Eduardo Bonelli. Intuitionistic hypothetical logic of proofs. *Electronic Notes in Theoretical Computer Science*, 300:89–103, 2013.

[Sch24] Moses Ilyich Schönfinkel. Über die bausteine der mathematischen logik. *Mathematische Annalen*, 92(3-4):305–316, 1924.

[Sol76] Robert M. Solovay. Provability interpretations of modal logic. *Israel journal of mathematics*, 25(3-4):287–304, 1976.

[SU06] Morten Heine Sørensen and Pawel Urzyczyn. *Lectures on the Curry-Howard isomorphism*, volume 149 of *Studies in Logic and the Foundations of Mathematics*. Elsevier Science, 2006.

[Tai65] William W. Tait. Infinitely long terms of transfinite type. *Studies in Logic and the Foundations of Mathematics*, 40:176–185, 1965.

[Tai67] William W. Tait. Intensional interpretations of functionals of finite type i. *Journal of Symbolic Logic*, 32(2):198–212, 06 1967.

[TS00] Anne Sjerp Troelstra and Helmut Schwichtenberg. Basic proof theory. (43), 2000.

[Tur37] Alan Mathison Turing. Computability and λ-definability. *The Journal of Symbolic Logic*, 2(04):153–163, 1937.

[TVD88] Anne Sjerp Troelstra and Dirk Van Dalen. *Constructivism in mathematics: an introduction*. North-Holland, 1988.

[vO90] Vincent van Oostrom. Lambda calculus with patterns. Technical report, Vrije Universiteit, 1990.

[vW64] Adriaan van Wijngaarden. *Algol 60 [sechzig]*. Vieweg & Sohn, 1964.

[WH08] Benjamin Wack and Clement Houtmann. Strong Normalization in two Pure Pattern Type Systems. *Mathematical Structures in Computer Science*, 18(3):431–465, 2008.

[WWdRZ02] Frank Wolter, Heinrich Wansing, Maarten de Rijke, and Michael Zakharyaschev, editors. *Advances in Modal Logic 3, papers from the third conference on "Advances in Modal logic", held in Leipzig (Germany) in October 2000*. World Scientific, 2002.

[Yav01] Rostislav E. Yavorsky. Provability logics with quantifiers on proofs. *Ann. Pure Appl. Logic*, 113(1-3):373–387, 2001.