On the Conservativity of Leibniz Equality

M.P.A. Sellink

Programming Research Group, University of Amsterdam

Kruislaan 403, 1098 SJ Amsterdam, The Netherlands

email: `alex@fwi.uva.nl`

Abstract

We embed a first order theory with equality in the Pure Type System λL that is a subsystem of the well-known type system λPRED2. The embedding is based on the Curry-Howard isomorphism, i.e. $\Rightarrow$ and $\forall$ coincide with $\rightarrow$ and Π. Formulas of the form $t_1 \stackrel{s}{=} t_2$ are treated as Leibniz equalities. That is, $t_1 \stackrel{s}{=} t_2$ is identified with the second order formula $\forall P.\, P(t_1) \Rightarrow P(t_2)$, which contains only $\Rightarrow$'s and $\forall$'s and can hence be embedded straightforwardly. We give a syntactic proof for the equivalence between derivability in the logic and inhabitance in λL. The idea of the proof is to introduce extra reduction steps, that reduce those proofterms that do not correspond to derivations in the logic to ones that do correspond to derivations in the logic.

Introduction

Many logics can be interpreted in type systems. For instance *Implicational Propositional Logic* can be interpreted in $\lambda^{\rightarrow}$. In [Ber89], Berardi designed a type system (λPREDω) in which *Minimal Higher Order Many Sorted Intuitionistic Predicate Logic* (PREDω) can be interpreted, as worked out ingreat detail in [TF92]. All these interpretations are based on the *Curry-Howard-isomorphism* [How80].

In this paper we focus on the interpretation in type theory of a *first order theory with equality*. We interpret this theory, which is a subsystem of *second order predicate logic* (PRED2), in a Pure Type System that is a subsystem of λPRED2. The interpretation uses the correspondence between $\Rightarrow$ and $\forall$ on the logical side with $\rightarrow$ and Π on the type theoretic side (i.e. $\forall$-Introduction coincides with Π-Introduction, etc.). The derivation rules for equality do not have their counterpart in the type system. This means that – assuming that we have binary relation symbols eq_A of type $A \rightarrow A \rightarrow *^p$ in the logic — we should add proofterms (variables) $refl_A$ and $subst_A$ to capture the inference rules on equality (see Table 1.2 on page 4):

$$refl_A \quad : \quad \Pi x{:}A.\, (eq_A\, x\, x) \tag{1}$$
$$subst_A \quad : \quad \Pi x, y{:}A.\, \Pi P{:}A \rightarrow *^p.\, (P\, x) \rightarrow (eq_A\, x\, y) \rightarrow P\, y \tag{2}$$

However, this raises a question: the intended meaning of the universal quantification over P in $subst_A$ is to quantify over interpretations of expressable properties (formulas with a hole in it, see Definition 1.1). Only if the variable P in the type of $subst_A$ is instantiated with the interpretation of such an expresssable property there is a natural correspondence between derivations for ϕ in the logic on the one hand and inhabitants of the corresponding type on the other hand. Such a correspondence is interesting because it means that inhabitance in the type system implies derivability in the logic — namely, the inhabitant itself encodes a derivation in the logic.

In general, however, we can instantiate P with a term that is not of the required form. We *can* but we *don't have* to. In other words: each λ-term that does not correspond to a derivation can be replaced by another λ-term (of the same type) that *does* correspond to a derivation. Thus, giving the conservativity of λL over the fragment that corresponds to the logic.

In this paper the conservativity of a type system over the fragment that corresponds to a first order theory with equality is proved for a rather artificial type system (λL) which was actually designed to capture this problem. That is, it is just strong enough to enable the formation of types $\Pi P{:}A{\rightarrow}*^p.\, P\,a \rightarrow P\,b$ used for the encoding of equality, and it is (almost) the weakest type system (PTS) with this property. We expect, however, that the ideas of the proof apply to a large variety of type systems. In [Sel96] we give the criterion (for PTS's) which should be fulfilled in order to apply the ideas of the proof. This criterion is fulfilled for all type systems of the logic cube [Geu93].

The rest of this paper is organised as follows: in Section 1 we introduce the first order theory and its accompanying proof theory. Then, in Section 2, the Pure Type System λL is introduced. We assume that the reader is familiar with the notion of Pure Type Systems (PTS's). The mapping from formulas to types is given in Section 3. Furthermore this section contains some elementairy properties of the mapping.

Then, in Section 4, we introduce η_L- and β_L-reductions. This new reductions give rise to so-called *super normal forms*, which directly correspond to derivations in the logic. The main difficulty is to show that β_L-reduction normalises. This normalisation is proved by associating a complexity $\gamma(M)$ to λ-term M and show that this complexity decreases with respect to some well-founded ordering after each β_L-reduction step.

Acknowledgements. Erik Barendsen, Marc Bezem, Tonny Hurkens, Jaco van de Pol, Jan Springintveld.

1 Signatures, terms and formulas

In this section we introduce the first order logic with equality.

Let S be a non-empty finite set of *sorts* (set symbols) and let $S^\star$ be the set of words over S. An S-sorted signature Σ is a family of sets $\langle \Sigma_{w,s} \mid w \in S^\star, s \in S\rangle$ such that the $\Sigma_{w,s}$ are pairwise disjoint. $f \in \Sigma_{w,s}$ is called a function symbol with domain w and co-domain s. If $w \equiv \varepsilon$ then f is called a constant. We write Σ_s for $\bigcup_{w\in S^\star} \Sigma_{w,s}$.

It is convenient to denote signatures by enumerating its function symbols and add the domain and codomain of each function symbol in a superscript. Thus, $\langle \ldots, f^{w,s}, \ldots\rangle$

denotes a signature containing a function symbol $f \in \Sigma_{w,s}$. Troughout this section, assume that Σ is an S-sorted signature.

We assume that, for every sort s, we have an infinite set $\mathbb{V}_s$ of variables at our disposal. All these sets are assumed to be pairwise disjoint. We write $\mathbb{V}$ for the set $\bigcup_s \mathbb{V}_s$ off *all* variables. We let $x, x', y, z, \ldots$ range over variables.

The set $\mathcal{T}_s(\Sigma)$ of Σ-terms of sort s is inductively defined by

- $\mathbb{V}_s \subseteq \mathcal{T}_s(\Sigma)$.
- $f(t_1, \ldots, t_n) \in \mathcal{T}_s(\Sigma)$ if $f \in \Sigma_{s_1 \cdots s_n, s}$ and $t_i \in \mathcal{T}s_i(\Sigma)$ for all $1 \leq i \leq n$.

The set $\bigcup_{s \in \mathcal{S}} \mathcal{T}_s(\Sigma)$ of *all* Σ-terms is denoted as $\mathcal{T}(\Sigma)$. We write $\mathrm{FV}(t)$ for the set of *free variables* of Σ-term t. This set is inductively defined by the following clauses:

$$\mathrm{FV}(x) = \{x\} \text{ for all } x \in \mathbb{V} \qquad\qquad \mathrm{FV}(f(t_1, \ldots, t_n)) = \bigcup_{i=1}^{n} \mathrm{FV}(t_i)$$

Next, we introduce the set $\mathcal{F}(\Sigma)$ of Σ-formulas. For each sort $s \in S$ we have a binary relation symbol $\stackrel{s}{=}$ in the language, from which the atomic formulas are constructed. Universal quantification is restricted to base domains.

- $t_1 \stackrel{s}{=} t_2 \ \in \mathcal{F}(\Sigma)$ for all $t_1, t_2 \in \mathcal{T}_s(\Sigma)$.
- $\phi \Rightarrow \psi \ \in \mathcal{F}(\Sigma)$ for all $\phi, \psi \in \mathcal{F}(\Sigma)$.
- $\forall x \,.\, \phi \ \in \mathcal{F}(\Sigma)$ for all $\phi \in \mathcal{F}(\Sigma)$ and for all $x \in \mathbb{V}$.

We assume that $\forall$ binds weaker then $\Rightarrow$.

The arity $\propto(\phi)$ of a Σ-formula ϕ is inductively defined by

$$\propto(t_1 \stackrel{s}{=} t_2) = 0 \qquad\qquad \propto(\psi \Rightarrow \phi) = \propto(\forall x \,.\, \phi) = 1 + \propto(\phi)$$

The set $\mathrm{FV}(\phi)$ of *free variables* of formula ϕ is defined in the usual way:

$$\begin{aligned} \mathrm{FV}(t_1 \stackrel{s}{=} t_2) &= \mathrm{FV}(t_1) \cup \mathrm{FV}(t_2) \\ \mathrm{FV}(\phi \Rightarrow \psi) &= \mathrm{FV}(\phi) \cup \mathrm{FV}(\psi) \\ \mathrm{FV}(\forall x \,.\, \phi) &= \mathrm{FV}(\phi) \setminus \{x\} \end{aligned}$$

Furthermore we write $\mathrm{FV}(\bigcup_{i=1}^{n} \phi_i)$ for $\bigcup_{i=1}^{n} \mathrm{FV}(\phi_i)$.

A function $Q : \mathcal{T}_s(\Sigma) \longrightarrow \mathcal{F}(\Sigma)$ is called a *property over s*.

Definition 1.1. *Let $s \in S$ and Σ an S-sorted signature. A property Q over s is called expressable iff there exists a formula $\phi \in \mathcal{F}(\Sigma)$ such that*

$$Q : t \longmapsto \phi[x := t]$$

for all $t \in \mathcal{T}_s(\Sigma)$. The expressable property that maps t to $\phi[x := t]$ is denoted as $\boldsymbol{\lambda} x \,.\, \phi$.

Definition 1.2. *Let* $\Phi \subseteq \mathcal{F}(\Sigma)$ *be a set of formulas and let* $\phi, \psi \in \mathcal{F}(\Sigma)$*. Furthermore let*

$$\begin{array}{c}\mathcal{D}\\ \Phi \vdash \phi\end{array}$$

denote the derivation $\mathcal{D}$ *with root* $\Phi \vdash \phi$*. The set* $\mathfrak{D}$ *of derivations is inductively defined by the inference rules of Table 1.2.*

$$\frac{}{\Phi \vdash \phi} \quad \text{if } \phi \in \Phi \qquad \text{Denotation: } \mathbf{P}\phi$$

$$\frac{}{\Phi \vdash t \stackrel{s}{=} t} \qquad \text{Denotation: } \mathbf{R}\, t$$

$$\frac{\begin{array}{c}\mathcal{D}_1\\ \Phi \vdash \phi[x_s := t_1]\end{array} \qquad \begin{array}{c}\mathcal{D}_2\\ \Phi \vdash t_1 \stackrel{s}{=} t_2\end{array}}{\Phi \vdash \phi[x_s := t_2]} \qquad \text{Denotation: } \mathbf{S}[\boldsymbol{\lambda} x_s.\phi]\,\mathcal{D}_1/\mathcal{D}_2$$

$$\frac{\begin{array}{c}\mathcal{D}\\ \Phi \cup \{\phi\} \vdash \psi\end{array}}{\Phi \vdash \phi \Rightarrow \psi} \qquad \text{Denotation: } \mathbf{I}\,\phi.\mathcal{D}$$

$$\frac{\begin{array}{c}\mathcal{D}_1\\ \Phi \vdash \phi \Rightarrow \psi\end{array} \qquad \begin{array}{c}\mathcal{D}_2\\ \Phi \vdash \phi\end{array}}{\Phi \vdash \psi} \qquad \text{Denotation: } \mathcal{D}_1\mathcal{D}_2$$

$$\frac{\begin{array}{c}\mathcal{D}\\ \Phi \vdash \phi\end{array}}{\Phi \vdash \forall x_s \,.\, \phi} \quad \text{provided } x_s \notin \mathrm{FV}(\Phi) \qquad \text{Denotation: } \mathbf{G}\, x_s.\mathcal{D}$$

$$\frac{\begin{array}{c}\mathcal{D}\\ \Phi \vdash \forall x_s \,.\, \phi\end{array}}{\Phi \vdash \phi[x_s := t]} \qquad \text{Denotation: } \mathcal{D}t$$

$$\frac{\begin{array}{c}\mathcal{D}\\ \Phi' \vdash \phi\end{array}}{\Phi \vdash \phi} \quad \text{provided } \Phi' \subseteq \Phi \qquad \text{Denotation: } \mathcal{D}$$

Table 1: Inference rules

Definition 1.3. *Let* $\Phi \subseteq \mathcal{F}(\Sigma)$ *and* $\phi \in \mathcal{F}(\Sigma)$*. Then we say that*

(i) $\mathcal{D}$ *proves* $\Phi \Vdash \phi$ *iff* $\mathcal{D}$ *is a derivation with root* $\Phi \Vdash \phi$.

(ii) $\Phi \Vdash \phi$ *iff there exists a derivation* $\mathcal{D}$ *such that* $\mathcal{D}$ *proves* $\Phi \Vdash \phi$.

Consider the following example of a derivation for the equality of two vairables z and z'. Let $\Phi = \{\forall x \,.\, f(x) \stackrel{s}{=} x \,,\, \forall x \,.\, f(x) \stackrel{s}{=} x \rightarrow z \stackrel{s'}{=} z'\} \subset \mathcal{F}(\Sigma)$ where Σ is the $\{s, s'\}$-sorted signature $\langle f^{s,s} \rangle$. Now consider the following derivation for $z \stackrel{s'}{=} z'$:

$$\dfrac{\dfrac{\overline{\Phi \Vdash \forall x \,.\, f(x) \stackrel{s}{=} x}}{\Phi \Vdash f(y) \stackrel{s}{=} y} \qquad \dfrac{\overline{\Phi \Vdash \forall x \,.\, f(x) \stackrel{s}{=} x \rightarrow z \stackrel{s'}{=} z'}}{\Phi \Vdash f(y) \stackrel{s}{=} y \rightarrow z \stackrel{s'}{=} z'}}{\Phi \Vdash z \stackrel{s'}{=} z'}$$

The free variables z and z', occurring in this derivation, are also occurring as free variables in the root $\Phi \Vdash z \stackrel{s'}{=} z'$ of this derivation. However, the free variable y — that is essential in that sense that we can not derive $z \stackrel{s'}{=} z'$ in Φ without introducing such a variable — is not occurring in the root of the derivation. Thus, it is significant to speak about the free variables of a derivation.

Definition 1.4. *The set* $\mathrm{FV}(\mathcal{D})$ *of free variables of derivation* $\mathcal{D}$ *is defined by induction over the structure of* $\mathcal{D}$*:*

$$\begin{aligned}
\mathrm{FV}(\mathbf{P}\phi) &= \mathrm{FV}(\phi) \\
\mathrm{FV}(\mathbf{R}\,t) &= \mathrm{FV}(t) \\
\mathrm{FV}(\mathbf{S}[Q]\,\mathcal{D}_1/\mathcal{D}_2) &= \mathrm{FV}(\mathcal{D}_1) \cup \mathrm{FV}(\mathcal{D}_2) \\
\mathrm{FV}(\mathbf{I}\,\phi.\mathcal{D}) &= \mathrm{FV}(\mathcal{D}) \cup \mathrm{FV}(\phi) \\
\mathrm{FV}(\mathcal{D}_1\mathcal{D}_2) &= \mathrm{FV}(\mathcal{D}_1) \cup \mathrm{FV}(\mathcal{D}_2) \\
\mathrm{FV}(\mathbf{G}\,x.\phi) &= \mathrm{FV}(\mathcal{D}) \setminus \{x\} \\
\mathrm{FV}(\mathcal{D}t) &= \mathrm{FV}(\mathcal{D}) \cup \mathrm{FV}(t)
\end{aligned}$$

The following proposition states that we can restrict the use of the substitution rule to atomic formulas. The contribution of this proposition is that the idea of β_L-reduction (see Section 4) is actually the "type-theoretic version" of the transformation presented in the proof of 1.5.

Lemma 1.5. *Substitution can be restricted to atomic formulas. Id est, the substitution rule of Table 1.2 can be replaced by*

$$\frac{t_1'[x := t_1] \stackrel{s'}{=} t_2'[x := t_1] \qquad t_1 \stackrel{s}{=} t_2}{t_1'[x := t_2] \stackrel{s'}{=} t_2'[x := t_2]}$$

Proof. By induction to the structure of formulas. An application of the substitution rule on a implication formula, say

$$\frac{\psi(t_1)\to\phi(t_1) \qquad t_1 \stackrel{s}{=} t_2}{\psi(t_2)\to\phi(t_2)}$$

can be replaced by

$$\dfrac{\dfrac{\dfrac{\psi(t_1)\to\phi(t_1) \qquad \dfrac{[\ \psi(t_2)\ ]^1 \qquad \dfrac{\dfrac{}{t_1 \stackrel{s}{=} t_1} \qquad t_1 \stackrel{s}{=} t_2}{t_2 \stackrel{s}{=} t_1}\,(*)}{\psi(t_1)}\,(*)}{\phi(t_1)} \qquad t_1 \stackrel{s}{=} t_2}{\phi(t_2)}\,(*)}{\psi(t_2)\to\phi(t_2)}\,1$$

The applications of the substitution rule, marked with a $(*)$, are either on an atomic formula or replaceble (by the induction hypothesis) by a derivation of the right shape. A similar argument holds for universaly quantified formulas. ❑

2 A PTS for Leibniz equality

In this section we introduce the Pure Type System λL and prove some meta results of this PTS. Then, in Section 3, we define a mapping $[\![\ ]\!]$ that maps terms, formulas and derivations of the logic introduced in Section 1, to terms of λL. The system λL is designed such that it is just strong enough to allow this mapping. It might help the reader to take a brief look at the first part of Section 3, where the mapping is introduced, before reading this section.

We give a short intuition behind the sorts, axioms and rules of the PTS-specification for λL.

In λL we have the following six sorts:

$*^p$	:	the type of all propositions
$*^s$	:	the type of all base domains
$*^f$	:	the type of all functions spaces
$\Box^p$	:	the type of $*^p$
$\Box^s$	:	the type of $*^s$
$\triangle$	:	the type of all unary predicates

As one can see, $*^p$ and $*^s$ are the only sorts for which a type is arranged in the specification. From G1 and Lemma 2.4, on page 8, it follows that the other sorts (that do not have a type by declaration) do not have a type at all in any context Γ. In general, there can be two reasons to arrange a type for a sort ς:

(i) One wants to introduce variables of type ς. According to the Start rule or the Weakening rule, ς must have a type.

(ii) One wants to form $A \to \varsigma$ or $\varsigma \to A$ for some term A. According to the Π-Formation rule, ς must have a type. (Moreover, there must be a rule in $\mathcal{R}$, assigning a type to this new formed type.)

Adding sort $\Box^s$ is motivated by (i). Actually, we can not even form $A \to *^s$ or $*^s \to A$ since $\mathcal{R}$ does not contain any rule of the form $\langle \_, *^s, \_\rangle$ or $\langle *^s, \_, \_\rangle$. The sort $\Box^p$ is added to allow the formation of a type $A \to *^p$ with A of type $*^s$ for unary predicates.

Definition 2.1. *The Pure Type System* λL *is defined by the following specification.*

$$\begin{array}{lcl} \mathcal{S} & = & \{*^p, *^s, *^f, \Box^p, \Box^s, \triangle\} \\ \mathcal{A} & = & \{(*^p, \Box^p), (*^s, \Box^s)\} \\ \mathcal{R} & = & \{(*^p, *^p), (*^s, *^p), (\triangle, *^p), (*^s, *^f), (*^s, *^s, *^f), (*^s, \Box^p, \triangle)\} \end{array}$$

Table 2: The PTS-specification for λL

It is immediately clear from the specification of λL, that this system is a sub-system of the more standard PTS known as λPRED2 (see e.g. [Geu93], p.85), which is isomorphic to second order predicate logic and well understood. In order to explain the rules for λL, we compare them with the rules of λPRED2.

$(*^s, *^s, *^f)$	:	this rule corresponds to the λPRED2 rule $(\mathsf{Set}, \mathsf{Set})$. However, parking the result in $*^f$ cuts out the possibility to iteratively apply this rule in order to form higher order types like $((s_1 \to s_2 \to s_3) \to (s_4 \to s_5)) \to s_6$ with s_i of type $*^s$ for all $1 \le i \le 6$.
$(*^s, *^f)$	:	this rule weakens the restriction introduced in the previous rule. By iteratively applying this rule, we can form the type $s_1 \to \ldots \to s_n$, with s_i of type $*^s$ for all $1 \le i \le n$.
$(*^s, \Box^p, \triangle)$	:	this rule corresponds to the λPRED2 rule $(\mathsf{Set}, \mathsf{Type}^p)$. However, parking the result in $\triangle$ cuts out the possibility to iteratively apply this rule in order to form $s_1 \to \ldots \to s_n \to *^p$.
$(*^p, *^p)$	:	this rule corresponds to the rule $(\mathsf{Prop}, \mathsf{Prop})$ of λPRED2.
$(*^s, *^p)$	:	this rule corresponds to the rule $(\mathsf{Set}, \mathsf{Prop})$ of λPRED2.
$(\triangle, *^p)$	:	this rule corresponds to the rule $(\mathsf{Type}^p, \mathsf{Prop})$ of λPRED2.

A typical derivation (fragment) for λL is the formation of the term

$$\Pi P{:}A \to *^p .\, P\,a \to P\,b$$

of type $*^p$ in a context Γ such that $\Gamma \vdash A : *^s$ and $\Gamma \vdash a, b : A$. (In contexts we omit the types)

$$\dfrac{\dfrac{\begin{array}{c}\vdots\\ \Gamma\vdash A:*^s\end{array}\qquad\begin{array}{c}\vdots\\ \Gamma,x\vdash *^p:\Box^p\end{array}}{\Gamma\vdash A\to *^p:\triangle}\,(*^s,\Box^p,\triangle)\qquad\dfrac{\begin{array}{c}\vdots\\ \Gamma,P\vdash P\,a:*^p\end{array}\qquad\begin{array}{c}\vdots\\ \Gamma,P,h\vdash P\,b:*^p\end{array}}{\Gamma,P\vdash P\,a\to P\,b:*^p}\,(*^p,*^p)}{\Gamma\vdash \Pi P\!:\!A\to *^p.\,P\,a\to P\,b:*^p}\,(\triangle,*^p)$$

2.1 General properties of λL

We state some general properties of λL. Most of these properties rely on the well-known *Generation Lemma*, which holds for PTS's in general.

Lemma 2.2. (Generation Lemma). *Let $c\in\mathbb{C}$, $x\in\mathbb{V}$ then,*

(i)	$\Gamma\vdash c:C$	$\Longrightarrow$	*there exists a sort* ς *such that* $C=_\beta\varsigma$ *and* $(c,\varsigma)\in\mathcal{A}$
(ii)	$\Gamma\vdash x:C$	$\Longrightarrow$	*there exists a sort* ς *and a term* B *such that* $B=_\beta C$, $\Gamma\vdash B:\varsigma$ *and* $x{:}B\in\Gamma$
(iii)	$\Gamma\vdash Fa:C$	$\Longrightarrow$	*there exist terms* A *and* B *such that* $C=_\beta B[x:=a]$, $\Gamma\vdash F:\Pi x{:}A.\,B$ *and* $\Gamma\vdash a:A$
(iv)	$\Gamma\vdash\Pi x{:}A.\,B:C$	$\Longrightarrow$	*there exists a rule* $\langle\varsigma_1,\varsigma_2,\varsigma_3\rangle\in\mathcal{R}$ *such that* $C=_\beta\varsigma_3$, $\Gamma\vdash A:\varsigma_1$ *and* $\Gamma,x{:}A\vdash B:\varsigma_2$
(v)	$\Gamma\vdash\lambda x{:}A.\,b:C$	$\Longrightarrow$	*there exists a sort* ς *and a term* B *such that* $C=_\beta\Pi x{:}A.\,B$, $\Gamma\vdash\Pi x{:}A.\,B:\varsigma$ *and* $\Gamma,x{:}A\vdash b:B$

The Generation Lemma is frequently used in the coming (sub)sections. Therefore, in order to improve the readability, we introduce the more significant names G1,... ,G5 for the lemmas 2.2(i),... ,2.2(v) respectively.

Another important lemma states that derivability is preserved under β-reductions in the subject, i.e. if $\Gamma\vdash A:B$ and $A\twoheadrightarrow_\beta A'$, then $\Gamma\vdash A':B$. We refer to this so-called *Subject Reduction Lemma* by SR.

Let $\mathbb{T}_\Gamma=\{M\in\mathbb{T}\mid\exists\sigma.\,\Gamma\vdash M:\sigma\}$. A term $M\in\mathbb{T}_\Gamma$ is called Γ-legal. Let $\mathbb{T}_\ell=\bigcup_\Gamma\mathbb{T}_\Gamma$. A term $M\in\mathbb{T}_\ell$ is called legal[1].

We write $\Gamma\vdash A:B:C$ as a shorthand for $\Gamma\vdash A:B$ and $\Gamma\vdash B:C$. Furthermore $\Gamma\vdash A::C$ means that $\Gamma\vdash A:B:C$ for some B.

Lemma 2.3. *Let $\sigma\in\Pi\mathbb{V}{:}\,\mathbb{T}.\,\mathbb{T}$ and $M=_\beta\sigma$, then $\textsc{Nf}(M)\in\Pi\mathbb{V}{:}\,\mathbb{T}.\,\mathbb{T}$.*

Proof. From $M=_\beta\sigma$ it follows that $\textsc{Nf}(M)\equiv\textsc{Nf}(\sigma)$. Furthermore $\textsc{Nf}(\sigma)\in\Pi\mathbb{V}{:}\,\mathbb{T}.\,\mathbb{T}$ because $\Pi\mathbb{V}{:}\,\mathbb{T}.\,\mathbb{T}$ is closed under β-reduction, so we are done. ❑

[1] according to this definition, topsorts (sorts that do not occur as the left component of an axiom) are not legal. We excluded them here only because it simplifies the definition

Lemma 2.4. *For all* $M \in \mathbb{T}_\ell$ *and all* $\varsigma \in \mathcal{S}$

$$M =_\beta \varsigma \Longrightarrow M \equiv \varsigma$$

Proof. It suffices to show that all reduction paths $M \twoheadrightarrow_\beta \varsigma$ are empty. Let $M \twoheadrightarrow_\beta \varsigma$ be a non-empty reduction path and let $(\lambda x{:}A.\,B)\,C \to_\beta B[x := C]$ be the last step of this reduction path. Now either $B \equiv \varsigma$ or $B \equiv x$ and $C \equiv \varsigma$. Both cases are ruled out by the Generation Lemma (Lemma 2.2 on page 8). ❑

Lemma 2.5. $\Gamma \vdash M :: A \Longrightarrow A \in \mathcal{S}$.

Proof. Suppose $\Gamma \vdash M : \sigma$ and $\Gamma \vdash \sigma : A$. By Correctness of types [Bar92] there exists a sort $\varsigma \in \mathcal{S}$ such that $\Gamma \vdash \sigma : \varsigma$ or $\sigma \in \mathcal{S}$. In the first case we know, by Uniqueness of Types [Bar92], that $A =_\beta \varsigma$ and hence $A \equiv \varsigma$ by Lemma 2.4, so $A \in \mathcal{S}$ and we are done. In the second case we apply G1 on $\Gamma \vdash \sigma : A$ and conclude that there exists a sort ς such that $A =_\beta \varsigma$. Now $A \equiv \varsigma$ so, again by Lemma 2.4, we are done. ❑

Lemma 2.6. $\Gamma, x{:}A \vdash M :: B \Longrightarrow \Gamma \vdash M[x := a] :: B$ *for all* $M \in \mathbb{T}_{\Gamma, x:A}$.

Proof. Suppose $\Gamma, x{:}A \vdash M : \sigma$ and $\Gamma, x{:}A \vdash \sigma : B$. By the Substitution Lemma [Bar92] we know that $\Gamma \vdash M[x := a] : \sigma[x := a]$ and $\Gamma \vdash \sigma[x := a] : B[x := a]$, so $\Gamma \vdash M[x := a] :: B[x := a]$. However, supertypes are sorts (Lemma 2.5) so $B[x := a] \equiv B$. ❑

Lemma 2.7. *Let* $\Gamma \vdash \Pi x{:}A.\,B : \varsigma$, *then*

(i) $\Gamma \vdash A : *^p \Longrightarrow \Gamma, x{:}A \vdash B : *^p$

(ii) $\Gamma \vdash A : \triangle \Longrightarrow \Gamma, x{:}A \vdash B : *^p$

Proof. Follows immediately from G4, Lemma 2.4 and the shape of $\mathcal{R}$. ❑

Lemma 2.8. *Let*

$$\Gamma \vdash \Pi x_0{:}A_0.\ \ldots \Pi x_k{:}A_k.\,\varsigma : \varsigma'. \tag{3}$$

Then $k = 0$, $\varsigma \equiv *^p$, $\varsigma' \equiv \triangle$ *and* $\Gamma \vdash A_0 : *^s$.

Proof. Assume that $k = 0$. From G4 and Lemma 2.4 it follows that $\mathcal{R}$ must contain a rule of the form $\langle \varsigma_1, \varsigma_2, \varsigma' \rangle$ such that $\Gamma, x_0{:}A_0 \vdash \varsigma : \varsigma_2$ and $\Gamma \vdash A_0 : \varsigma_1$. Now G1 gives that $\varsigma_2 =_\beta \Box$ and hence, by Lemma 2.4, $\varsigma_2 \equiv \Box$. By inspecting $\mathcal{R}$ we find that $\langle \varsigma_1, \varsigma_2, \varsigma' \rangle \equiv \langle *^s, \Box^p, \triangle \rangle$ so in case $k = 0$ indeed $\varsigma \equiv *^p$, $\varsigma' \equiv \triangle$ and $\Gamma \vdash A_0 : *^s$.

Next, we prove by induction to k that $k > 0$ is impossible. Let $k = 1$. Again, from G4 it follows that there exists an $\varsigma'' \in \mathcal{S}$ such that $\mathcal{R}$ contains a rule of the form $\langle \_, \varsigma'', \varsigma' \rangle$ and $\Gamma, x_0{:}A_0 \vdash \Pi x_1{:}A_1 . \varsigma : \varsigma''$. As we just proved in the $k = 0$ case, this implies that $\varsigma'' \equiv \triangle$, so $\mathcal{R}$ contains a rule of the form $\langle \_, \triangle, \varsigma' \rangle$. Contradiction.

Assume that for $0, \ldots, k-1$ there are no judgements of the form (3) derivable. Now if $\Gamma \vdash \Pi x_0{:}A_0 . \ldots \Pi x_k{:}A_k . \varsigma : \varsigma'$, then G4 states that there is a derivation

$$\Gamma, x_0{:}A_0 \vdash \Pi x_1{:}A_1 . \ldots \Pi x_k{:}A_k . \varsigma' : \varsigma'',$$

contradicting the induction hypothesis. ❑

Lemma 2.9. *For all contexts* Γ *and for all* $\sigma \in \mathbb{T}$ *we have*

(i) $\Gamma \vdash \sigma : *^s \Longrightarrow \sigma =_\beta X$ *for some* $X{:}*^s \in \Gamma$

(ii) $\Gamma \vdash \sigma : \Box^p \Longrightarrow \sigma \equiv *^p$

(iii) $\Gamma \vdash \sigma : \triangle \Longrightarrow \sigma =_\beta X \to *^p$ *for some* $X{:}*^s \in \Gamma$

Proof. By SR we know that there is a derivation for $\mathrm{NF}(\sigma)$ whenever there is a derivation for σ. Let $\mathrm{NF}(\sigma) \equiv \Pi y_1{:}B_1 . \ldots \Pi y_m{:}B_m . v M_1 \ldots M_k$.

(i) By G4 we have that there must be a rule of the form $\langle \_, \_, *^s \rangle$ in $\mathcal{R}$. Since this is not the case we know that $m = 0$. Let $k > 0$. Now G3 states that there are terms A and B such that

$$\Gamma \vdash v M_1 \ldots M_{k-1} : \Pi x{:}A . B \tag{4}$$

and $B[x := M_k] =_\beta *^s$. From Lemma 2.4 it follows that $B[x := M_k] \equiv *^s$. If $B \equiv *^s$ then (4) and Correctness of Types [Bar92] imply $\Gamma \vdash A \to *^s : \varsigma$ (for some ς) contradicting Lemma 2.8. The only other possibility, i.e. $B \equiv x$ and $M_k \equiv *^s$, is ruled out by G4, so $k = 0$ and we are done.

(ii) Similar arguments. Use Lemma 2.4.

(iii) Let $m = 0$. Now $k = 0$ is ruled out by G2 (variables do not inhabit topsorts), so $k > 0$. From G3 it follows that there are terms A and B such that $\Gamma \vdash v M_1 \ldots M_{k-1} : \Pi x{:}A . B$ and $B[x := M_k] =_\beta \triangle$. By Lemma 2.4, $B[x := M_k] \equiv \triangle$ so there are two cases. Both cases are excluded by the Generation Lemma.

We conclude that $m > 0$. By G4 there are $\varsigma, \varsigma' \in \mathcal{S}$ such that $\Gamma \vdash B_1 : \varsigma$, $\Gamma, y_1{:}B_1 \vdash \Pi y_2{:}B_2 . \ldots \Pi y_m{:}B_m . v M_1 \ldots M_k : \varsigma'$ and $\langle \varsigma, \varsigma', \triangle \rangle \in \mathcal{R}$. Hence $\varsigma \equiv *^s$ and $\varsigma' \equiv \Box^p$. By (i) we have that $B_1 =_\beta X$ for some $X{:}*^s \in \Gamma$. Furthermore, by (ii), $\Pi y_2{:}B_2 . \ldots \Pi y_m{:}B_m . v M_1 \ldots M_k \equiv *^p$ and we are done.

❑

Lemma 2.10. *Let* $\Gamma \vdash P :: \triangle$ *then* $P \in \mathbb{V} \cup \lambda\mathbb{V}\colon\mathbb{T}.\,\mathbb{T}$

Proof. By Lemma 2.9(iii) we have that $\Gamma \vdash P : X \to *^p$ for some $X{:}*^s \in \Gamma$. Since $X \to *^p \notin \mathcal{S}$, we may exclude $P \in \mathbb{C} \cup \Pi\mathbb{V}\colon\mathbb{T}.\,\mathbb{T}$. We prove that P can not be an application. Let $P \equiv F\,a$ then, by G3, there exist terms A and B such that $\Gamma \vdash F : \Pi x{:}A.\,B$ and $B[x := a] =_\beta X \to *^p$. Now $\Gamma \vdash B[x := a] : \triangle$ and hence also, by Lemma 2.6, $\Gamma, x{:}A \vdash B : \triangle$. Furthermore $\Gamma \vdash \Pi x{:}A.\,B : \varsigma$ for some $\varsigma \in \mathcal{S}$. Applying G4 gives that we have a rule $\langle \varsigma', \triangle, \varsigma \rangle$. Contradiction. We conclude that P is not an application. ❑

Lemma 2.11. *Let* $\Gamma \vdash a :: *^s$ *for some context* Γ. *Then*

$$\textsc{Nf}(\sigma) \in \Pi\mathbb{V}\colon\mathbb{T}.\,\mathbb{T} \iff \textsc{Nf}(\sigma[x := a]) \in \Pi\mathbb{V}\colon\mathbb{T}.\,\mathbb{T}.$$

Proof. Let $\sigma \in \mathbb{T}$.
($\Longrightarrow$) Suppose $\textsc{Nf}(\sigma) \equiv \Pi y{:}A.\,B \in \Pi\mathbb{V}\colon\mathbb{T}.\,\mathbb{T}$. Then $\sigma[x := a] =_\beta \textsc{Nf}(\sigma)[x := a] \equiv (\Pi y{:}A.\,B)[x := a] \equiv \Pi y{:}A[x := a].\,B[x := a] \in \Pi\mathbb{V}\colon\mathbb{T}.\,\mathbb{T}$. Use Lemma 2.3.
($\Longleftarrow$) Suppose

$$\textsc{Nf}(\sigma) \equiv \lambda x_1{:}A_1.\,\ldots\lambda x_n{:}A_n.\,\Pi y_1{:}B_1.\,\ldots\Pi y_m{:}B_m.\,vM_1\ldots M_k$$

and let $\textsc{Nf}(\sigma[x := a]) \in \Pi\mathbb{V}\colon\mathbb{T}.\,\mathbb{T}$. We write θ for the substitution $x := \textsc{Nf}(a)$.

$$\begin{aligned}
&\textsc{Nf}(\sigma[x := a])\\
&\quad\equiv\ \textsc{Nf}(\textsc{Nf}(\sigma)[x := \textsc{Nf}(a)])\\
&\quad\equiv\ \textsc{Nf}(\textsc{Nf}(\sigma)[\theta])\\
&\quad\equiv\ \textsc{Nf}((\lambda x_1{:}A_1.\,\ldots\lambda x_n{:}A_n.\,\Pi y_1{:}B_1.\,\ldots\Pi y_m{:}B_m.\,vM_1\ldots M_k)[\theta])\\
&\quad\equiv\ \textsc{Nf}(\lambda x_1{:}A_1[\theta].\,\ldots\lambda x_n{:}A_n[\theta].\,\Pi y_1{:}B_1[\theta].\,\ldots\Pi y_m{:}B_m[\theta].\,v[\theta]M_1[\theta]\ldots M_k[\theta])\\
&\quad\equiv\ \lambda x_1{:}\textsc{Nf}(A_1[\theta]).\,\ldots\lambda x_n{:}\textsc{Nf}(A_n[\theta]).\\
&\qquad\quad \Pi y_1{:}\textsc{Nf}(B_1[\theta]).\,\ldots\Pi y_m{:}\textsc{Nf}(B_m[\theta]).\,\textsc{Nf}(v[\theta]M_1[\theta]\ldots M_k[\theta])
\end{aligned}$$

Now we distinguish between two cases:

(i) $v \not\equiv x$. Now $\textsc{Nf}(v[\theta]M_1[\theta]\ldots M_k[\theta]) \equiv v\textsc{Nf}(M_1[\theta])\ldots\textsc{Nf}(M_k[\theta])$ because the left-terms $vM_1[\theta]\ldots M_i[\theta]$ of the applications in $vM_1[\theta]\ldots M_k[\theta]$ are never a λ-abstraction term. Now it follows from $\textsc{Nf}(\sigma[x := a]) \in \Pi\mathbb{V}\colon\mathbb{T}.\,\mathbb{T}$ that $n = 0$ and $m > 0$, so $\textsc{Nf}(\sigma) \in \Pi\mathbb{V}\colon\mathbb{T}.\,\mathbb{T}$.

(ii) $v \equiv x$. Now $\textsc{Nf}(v[\theta]M_1[\theta]\ldots M_k[\theta]) \equiv \textsc{Nf}(\textsc{Nf}(a)M_1[\theta]\ldots M_k[\theta])$. Note however, that $\textsc{Nf}(a) \notin \lambda\mathbb{V}\colon\mathbb{T}.\,\mathbb{T}$ because — by Lemma 2.9(i) — the type of $\textsc{Nf}(a)$ is β-converible with a variable. Hence we can — as in the previous case — push the $\textsc{Nf}(..)$-operator inside: $\textsc{Nf}(\textsc{Nf}(a)M_1[\theta]\ldots M_k[\theta]) \equiv \textsc{Nf}(a)\textsc{Nf}(M_1[\theta])\ldots\textsc{Nf}(M_k[\theta])$ etc.

❑

3 The representation

We define a mapping $\llbracket\ \rrbracket$ that maps terms, formulas and derivations to terms of λL. Let $\Gamma_1 \uplus \Gamma_2$ denote the concatenation of Γ_1 and Γ_2', where Γ_2' is obtained from Γ_2 by α-conversion on the variables that occur in Γ_1.

Definition 3.1. *Let Σ be an S-sorted signature and suppose that $\llbracket\ \rrbracket : \Sigma \longrightarrow \mathbb{V}$ is an injection. Then $\Gamma(\Sigma)$ is the smallest context such that*

- $\Gamma(\Sigma) \vdash \llbracket s \rrbracket : *^s$ *for all* $s \in S$
- $\Gamma(\Sigma) \vdash \llbracket f \rrbracket : \llbracket s_1 \rrbracket \to \ldots \to \llbracket s_n \rrbracket \to \llbracket s \rrbracket$ *for all* $f \in \Sigma_{s_1 \cdots s_n, s}$

Definition 3.2. *Let $t \in \mathcal{T}_s(\Sigma)$ and $\mathrm{FV}(t) = \{x_1, \ldots, x_n\}$ with $x_i \in \mathbb{V}_{s_i}$ for all $1 \leq i \leq n$. Then Γ_t $\langle x_1{:}\llbracket s_1 \rrbracket, \ldots, x_n{:}\llbracket s_n \rrbracket \rangle$.*

If $n > 0$, then Γ_t is not necessarily a legal context. However, the context $\Gamma(\Sigma) \uplus \Gamma_t$, which is denoted as $\Gamma_t(\Sigma)$, is always legal. Analogous to Definition 3.2 we define Γ_ϕ for $\phi \in \mathcal{F}(\Sigma)$ and we write $\Gamma_\phi(\Sigma)$ for $\Gamma(\Sigma) \uplus \Gamma_\phi$. A mapping $\llbracket\ \rrbracket : \Sigma \longrightarrow \mathbb{T}$ canonically extends to a mapping $\llbracket\ \rrbracket : \mathcal{T}(\Sigma) \longrightarrow \mathbb{T}$ by putting

$$\begin{array}{rcl} x & \longmapsto & x \\ f(t_1, \ldots, t_n) & \longmapsto & \llbracket f \rrbracket \llbracket t_1 \rrbracket \ldots \llbracket t_n \rrbracket \end{array}$$

Note that there is a 1–1 correspondence between subterms of t and $\llbracket t \rrbracket$.

Lemma 3.3. *Let $t \in \mathcal{T}_s(\Sigma)$. Then $\Gamma_t(\Sigma) \vdash \llbracket t \rrbracket : \llbracket s \rrbracket$.*

Proof. By a straightforward induction on the structure of t. ❑

Definition 3.4. *We extend $\llbracket\ \rrbracket$ to a mapping on $\llbracket\ \rrbracket : \mathcal{F}(\Sigma) \longrightarrow \mathbb{T}$ by induction to the structure of the formulas.*

$$\begin{array}{rcl} t_1 \overset{s}{=} t_2 & \longmapsto & \Pi P{:}\llbracket s \rrbracket \to *^p . P\llbracket t_1 \rrbracket \to P\llbracket t_2 \rrbracket \\ \phi \Rightarrow \psi & \longmapsto & \llbracket \phi \rrbracket \to \llbracket \psi \rrbracket \\ \forall x_s . \phi & \longmapsto & \Pi x_s{:}\llbracket s \rrbracket . \llbracket \phi \rrbracket \end{array}$$

Lemma 3.5. *Let $\phi \in \mathcal{F}(\Sigma)$. Then $\Gamma_\phi(\Sigma) \vdash \llbracket \phi \rrbracket : *^p$.*

Proof. By a straightforward induction on the structure of ϕ. ❑

This definition implies that $[\![\phi]\!] \equiv \Pi z_1\!:\!B_1 . \ldots \Pi z_{\propto(\phi)}\!:\!B_{\propto(\phi)} . [\![t_1 \stackrel{s}{\doteq} t_2]\!]$ where $B_i : *$ for $1 \leq i \leq \propto(\phi)$. If $B_i : *^p$ then $z_i \notin \mathrm{FV}(\Pi z_{i+1}\!:\!B_{i+1} . \ldots \Pi z_m\!:\!B_m . [\![t_1 \stackrel{s}{\doteq} t_2]\!])$ and $B_i \equiv [\![\phi]\!]$ for some $\phi \in \mathcal{F}(\Sigma)$. If $B_i : *^s$ then $B_i \equiv [\![s]\!]$ for some $s \in S$. Other trivial remarks are that $[\![t]\!]$ and $[\![\phi]\!]$ are normal forms and

$$[\![\phi]\!] \equiv [\![\psi]\!] \iff \phi \equiv \psi$$

The 1–1 correspondence between subterms of t and $[\![t]\!]$ is preserved for formulas, i.e. there is a 1–1 correspondence between subterms of ϕ and $[\![\phi]\!]$.

Recall that we gave a meaning to expressions of the form Γ_x with $x \in \mathcal{F}(\Sigma)$. In the following definition we define contexts Γ_x for $x \subset \mathcal{F}(\Sigma)$. The reader is warned not to confuse Γ_ϕ with $\Gamma_{\{\phi\}}$.

Definition 3.6. *Let $\phi \in \mathcal{F}(\Sigma)$. Then $\Gamma_{\{\phi\}} \quad \Gamma_\phi \uplus \langle v_\phi : [\![\phi]\!]\rangle$. Furthermore $\Gamma_\Phi = \biguplus_{\phi\in\Phi} \Gamma_{\{\phi\}}$ and $\Gamma_\Phi(\Sigma)$ denotes $\Gamma(\Sigma) \uplus \Gamma_\Phi$.*

As usual, derivations are mapped to λ-terms (*proofs as objects*). However, the context in which we can represent a derivation $\mathcal{D}$ for $\Phi \Vdash \phi$ is not just $\Gamma_\Phi(\Sigma)$ since the derivation $\mathcal{D}$ may contain more free variables than those occurring in Φ or ϕ (as we explained on page 5). The natural solution is to define a context $\Gamma_\mathcal{D}$ that — analogously to Γ_t and Γ_ϕ — contains the free variables of derivation $\mathcal{D}$. Thus, for each derivation in a fixed theory Φ, the context $\Gamma_\Phi(\Sigma)$ should be extended with the free variables of that particular derivation in order to obtain a context in which we can represent that derivation (see Definition 3.8). We somewhat losely write $\Gamma_\Phi(\Sigma)^+$ for any context obtained from $\Gamma_\Phi(\Sigma)$ by adding a finite number of variables of supertype $*^s$, i.e. we do not explicitly specify the extra variables used in a particular derivation and simply write $\Gamma_\Phi(\Sigma)^+ \vdash M : \sigma$ where we actually mean $\Gamma_\Phi(\Sigma) \uplus \langle x_1 : [\![s_1]\!], \ldots, x_n : [\![s_n]\!]\rangle \vdash M : \sigma$ for some $x_1, \ldots, x_n \in \mathbb{V}$.

Definition 3.7. *We extend $[\![\ ]\!]$ to expressable properties. Let $\boldsymbol{\lambda} x_s.\phi$ be a property over s, then*

$$\boldsymbol{\lambda} x_s.\phi \quad \longmapsto \quad \lambda x_s\!:\![\![s]\!] . [\![\phi]\!]$$

Note that the types $\Pi x\!:\![\![s]\!] . [\![x \stackrel{s}{\doteq} x]\!]$ and $\Pi x, y\!:\![\![s]\!] . [\![x \stackrel{s}{\doteq} y]\!] \to [\![y \stackrel{s}{\doteq} x]\!]$, expressing reflexivity and symmetry of equality respectively, are inhabited. For a technical reason, however, we assume variables refl_s and sym_s for each $s \in S$:

$$\begin{aligned} \mathit{refl}_s &: \Pi x\!:\![\![s]\!] . [\![x \stackrel{s}{\doteq} x]\!] \\ \mathit{sym}_s &: \Pi x, y\!:\![\![s]\!] . [\![x \stackrel{s}{\doteq} y]\!] \to [\![y \stackrel{s}{\doteq} x]\!] \end{aligned}$$

Definition 3.8. *Let $\mathcal{D}$ be a derivation for $\Phi \Vdash \phi$. We define $[\![\mathcal{D}]\!] \in \mathbb{T}_{\Gamma_\Phi(\Sigma)^+}$ by induction to the structure of $\mathcal{D}$:*

$$\begin{array}{rcl}
\mathbf{P}\phi & \longmapsto & v_\phi \\
\mathbf{R}\, t_s & \longmapsto & \mathit{refl}_s\, [\![t_s]\!] \\
\mathbf{S}[\boldsymbol{\lambda} x.\phi]\, \mathcal{D}_1/\mathcal{D}_2 & \longmapsto & [\![\mathcal{D}_2]\!][\![\boldsymbol{\lambda} x.\phi]\!][\![\mathcal{D}_1]\!] \\
\mathbf{I}\, \phi.\mathcal{D} & \longmapsto & \lambda v_\phi : [\![\phi]\!] \,.\, [\![\mathcal{D}]\!] \\
\mathcal{D}_1\mathcal{D}_2 & \longmapsto & [\![\mathcal{D}_1]\!][\![\mathcal{D}_2]\!] \\
\mathbf{G}\, x_s.\mathcal{D} & \longmapsto & \lambda x_s : [\![s]\!] \,.\, [\![\mathcal{D}]\!] \\
\mathcal{D}t & \longmapsto & [\![\mathcal{D}]\!][\![t]\!]
\end{array}$$

We mentioned before that $[\![t]\!]$ and $[\![\phi]\!]$ are normal forms, for all $t \in \mathcal{T}(\Sigma)$ and for all $\phi \in \mathcal{F}(\Sigma)$. A term M, which is not in normal form, is never of the form $[\![t]\!]$. However, we want to view M as a representation for $t \in \mathcal{T}(\Sigma)$ whenever $M =_\beta [\![t]\!]$. And also, we want to view M as a representation for $\phi \in \mathcal{F}(\Sigma)$ whenever $M =_\beta [\![\phi]\!]$. Therefore the sets $[\![\mathcal{T}(\Sigma)]\!]_\beta$ and $[\![\mathcal{F}(\Sigma)]\!]_\beta$ of terms that are β-convertible to a representation of a term or a formula respectively will play a more important role than $[\![\mathcal{T}(\Sigma)]\!]$ and $[\![\mathcal{F}(\Sigma)]\!]$.

Let $\Gamma \vdash M : \sigma$. In the sequel we will see that $M =_\beta [\![\mathcal{D}]\!] \Longrightarrow \sigma \in [\![\mathcal{F}(\Sigma)]\!]_\beta$ and that the converse does not hold.

3.1 Properties of the representation

Lemma 3.9. *Let $t' \in \mathcal{T}_s(\Sigma)$, $x \in \mathbb{V}_s$ then*

$$[\![t\,[x := t']\,]\!] \equiv [\![t]\!][x := [\![t']\!]]$$

for all $t \in \mathcal{T}(\Sigma)$.

Proof. Induction on the structure of t.

- $t \in \mathbb{V}_s$. Distinguish between two cases.
  - $t \equiv x$. Now $[\![t[x := t']]\!] \equiv [\![x[x := t']]\!] \equiv [\![t']\!] \equiv x[x := [\![t']\!]] \equiv [\![x]\!][x := [\![t']\!]] \equiv [\![t]\!][x := [\![t']\!]]$.
  - $t \not\equiv x$. Now $[\![t[x := t']]\!] \equiv [\![t]\!] \equiv [\![t]\!][x := [\![t']\!]]$.
- $t \equiv f(t_1, \ldots, t_n)$. Now 3.9 holds for $t_1, \ldots, t_n$ by induction hypothesis.

$$\begin{array}{rcl}
[\![t[x := t']]\!] & \equiv & [\![f(t_1, \ldots, t_n)[x := t']]\!] \\
 & \equiv & [\![f(t_1[x := t'], \ldots, t_n[x := t'])]\!] \\
 & \equiv & [\![f]\!][\![t_1[x := t']]\!] \cdots [\![t_n[x := t']]\!] \\
 & \equiv & [\![f]\!][\![t_1]\!][x := [\![t']\!]] \cdots [\![t_n]\!][x := [\![t']\!]] \\
 & \equiv & [\![f]\!][\![t_1]\!] \cdots [\![t_n]\!][x := [\![t']\!]] \\
 & \equiv & [\![f(t_1, \ldots, t_n)]\!][x := [\![t']\!]] \\
 & \equiv & [\![t]\!][x := [\![t']\!]]
\end{array}$$

❑

Lemma 3.10. *Let $\phi \in \mathcal{F}(\Sigma)$, $x \in \mathbb{V}_s$ and $t \in \mathcal{T}_s(\Sigma)$ then*

$$[\![\phi[x := t]]\!] \equiv [\![\phi]\!][x := [\![t]\!]]$$

Proof. Induction on the structure of ϕ. We use Lemma 3.9.

- $\phi \equiv t_1 \stackrel{s}{=} t_2$. Now

$$\begin{array}{rcl}
[\![\phi[x := t]]\!] & \equiv & [\![(t_1 \stackrel{s}{=} t_2)[x := t]]\!] \\
& \equiv & [\![t_1[x := t] \stackrel{s}{=} t_2[x := t]]\!] \\
& \equiv & \Pi P : [\![s]\!] \to *^p . P[\![t_1[x := t]]\!] \to P[\![t_2[x := t]]\!] \\
& \equiv & \Pi P : [\![s]\!] \to *^p . P[\![t_1]\!][x := [\![t]\!]] \to P[\![t_2]\!][x := [\![t]\!]] \\
& \equiv & (\Pi P : [\![s]\!] \to *^p . P[\![t_1]\!] \to [\![t_2]\!])[x := [\![t]\!]] \\
& \equiv & [\![t_1 \stackrel{s}{=} t_2]\!][x := [\![t]\!]] \\
& \equiv & [\![\phi]\!][x := [\![t]\!]].
\end{array}$$

- $\phi \equiv \phi_1 \Rightarrow \phi_2$. Now 3.10 holds for ϕ_1 and ϕ_2 by induction hypothesis.

$$\begin{array}{rcl}
[\![\phi[x := t]]\!] & \equiv & [\![(\phi_1 \Rightarrow \phi_2)[x := t]]\!] \\
& \equiv & [\![\phi_1[x := t] \Rightarrow \phi_2[x := t]]\!] \\
& \equiv & [\![\phi_1[x := t]]\!] \to [\![\phi_2[x := t]]\!] \\
& \equiv & [\![\phi_1]\!][x := [\![t]\!]] \to [\![\phi_2]\!][x := [\![t]\!]] \\
& \equiv & ([\![\phi_1]\!] \to [\![\phi_2]\!])[x := [\![t]\!]] \\
& \equiv & [\![\phi_1 \Rightarrow \phi_2]\!][x := [\![t]\!]] \\
& \equiv & [\![\phi]\!][x := [\![t]\!]].
\end{array}$$

- $\phi \equiv \forall y.\psi$. By convention $y \notin \mathrm{FV}(t)$. Now 3.10 holds for ψ by induction hypothesis.

$$\begin{array}{rcl}
[\![\phi[x := t]]\!] & \equiv & [\![(\forall y.\psi)[x := t]]\!] \\
& \equiv & [\![\forall y.(\psi[x := t])]\!] \\
& \equiv & \Pi y : [\![s]\!] . [\![\psi[x := t]]\!] \\
& \equiv & \Pi y : [\![s]\!] . ([\![\psi]\!][x := [\![t]\!]]) \\
& \equiv & (\Pi y : [\![s]\!] . [\![\psi]\!])[x := [\![t]\!]] \\
& \equiv & [\![\forall y.\psi]\!][x := [\![t]\!]] \\
& \equiv & [\![\phi]\!][x := [\![t]\!]].
\end{array}$$

❑

Lemma 3.11. *Let $x \in \mathbb{V}_s$, $t \in \mathcal{T}_s(\Sigma)$ and $b \in \mathbb{T}$ then*

$$\textsc{Nf}(b)[x := [\![t]\!]] \equiv \textsc{Nf}(b[x := [\![t]\!]])$$

Proof. Obviously $\textsc{Nf}(b)[x := [\![t]\!]] =_\beta b[x := [\![t]\!]] =_\beta \textsc{Nf}(b[x := [\![t]\!]])$. Note that the substitution $x := [\![t]\!]$ does not create new redexes because $[\![t]\!]$ is not a λ-abstraction term. Consequently $\textsc{Nf}(b)[x := [\![t]\!]]$ is a normal form which proves the lemma because $N_1 =_\beta N_2 \Longrightarrow N_1 \equiv N_2$ for all normal forms N_1, N_2. ❑

Lemma 3.12. *Let $x \in \mathbb{V}_s$, $t \in \mathcal{T}_s(\Sigma)$ and $b \in \mathbb{T}$ then*

$$b[x := [\![t]\!]] \in [\![\mathcal{F}(\Sigma)]\!]_\beta \iff b \in [\![\mathcal{F}(\Sigma)]\!]_\beta$$

Proof.
($\Longrightarrow$) $b[x := [\![t]\!]] \in [\![\mathcal{F}(\Sigma)]\!]_\beta$, say $[\![\phi]\!] \equiv \textsc{Nf}(b[x := [\![t]\!]]) \equiv \textsc{Nf}(b)[x := [\![t]\!]]$. In this latter term we underline those occurrences of $[\![t]\!]$ that are substitutes of an occurrence of x in $\textsc{Nf}(b)$. In ϕ we can underline exactly those occurrences of t that correspond to underlined occurrences of $[\![t]\!]$ in $\textsc{Nf}(b)[x := [\![t]\!]]$. Because there is a 1–1 correspondence between subterms of ϕ and $[\![\phi]\!]$ it can not happen that an underlined $[\![t]\!]$ and a not-underlined $[\![t]\!]$ correspond to the same occurrence of t in ϕ. Now $\textsc{Nf}(b) \equiv [\![\psi]\!]$ where ψ is obtained from ϕ by replacing $\underline{t}$ by x so $b \in [\![\mathcal{F}(\Sigma)]\!]_\beta$.
($\Longleftarrow$) $b \in [\![\mathcal{F}(\Sigma)]\!]_\beta$, say $\textsc{Nf}(b) \equiv [\![\phi]\!]$. Now

$$\textsc{Nf}(b[x := [\![t]\!]]) \equiv \textsc{Nf}(b)[x := [\![t]\!]] \equiv [\![\phi]\!][x := [\![t]\!]] \equiv [\![\phi[x := t]]\!]$$

so $b[x := [\![t]\!]] \in [\![\mathcal{F}(\Sigma)]\!]_\beta$. ❑

Lemma 3.13. *Let $\varsigma \in \{*^s, *^f\}$ and $v \in \mathbb{V}$. If $\Gamma_\Phi(\Sigma)^+ \vdash vM_1 \ldots M_k : \sigma : \varsigma$, then*

(i) $\textsc{Nf}(\sigma) \equiv [\![s_1]\!] \to \ldots \to [\![s_n]\!]$ *for some* $s_1, \ldots, s_n \in S$

(ii) *If $k > 0$, then there exists an $s_0 \in S$ such that*

(a) $\Gamma_\Phi(\Sigma)^+ \vdash vM_1 \ldots M_{k-1} : [\![s_0]\!] \to \sigma : *^f$

(b) $\Gamma_\Phi(\Sigma)^+ \vdash M_k : [\![s_0]\!] : *^s$

Proof. From the Free Variable Lemma [Bar92] it follows that $v \in \textsc{Fv}(\Gamma_\Phi(\Sigma)^+)$. If $k = 0$, then (i) follows directly from Definition 3.1 on page 12 and (ii) holds vacuously.

Let $k > 0$ and assume that the lemma holds for $0, \ldots, k-1$. By G3 we have that there are terms A and B such that $\Gamma_\Phi(\Sigma)^+ \vdash vM_1 \ldots M_{k-1} : \Pi x{:}A.\, B$ and $\Gamma_\Phi(\Sigma)^+ \vdash M_k : A$. Moreover $B[x := M_k] =_\beta \sigma$. By Correctness of Types [Bar92], we have that $\Gamma_\Phi(\Sigma)^+ \vdash \Pi x{:}A.\, B : \varsigma'$ for some $\varsigma' \in \mathcal{S}$. From G4 it follows that $\Gamma_\Phi(\Sigma)^+, x{:}A \vdash B : \varsigma''$ for some ς'' such that $\langle \_, \varsigma'', \varsigma' \rangle \in \mathcal{R}$. By the Substitution Lemma [Bar92] we obtain that $\Gamma_\Phi(\Sigma)^+ \vdash B[x := M_k] : \varsigma''$ so $\varsigma'' \equiv \varsigma$. Now, by inspecting $\mathcal{R}$ we find that $\varsigma' \equiv *^f$ and hence we can apply the induction hypothesis on $vM_1 \ldots M_{k-1}$. Thus, we get:

(I) $\textsc{Nf}(\Pi x{:}A\,.\,B) \equiv [\![s_1]\!] \to \ldots \to [\![s_n]\!]$ for some $s_1, \ldots, s_n \in S$

(II) If $k-1>0$, then there exists an $s_0 \in S$ such that

(a) $\Gamma_\Phi(\Sigma)^+ \vdash vM_1 \ldots M_{k-2} : [\![s_0]\!] \to \Pi x{:}A\,.\,B : *^f$

(b) $\Gamma_\Phi(\Sigma)^+ \vdash M_{k-1} : [\![s_0]\!] : *^s$

Since $\Pi x{:}\textsc{Nf}(A)\,.\,\textsc{Nf}(B) \equiv \textsc{Nf}(\Pi x{:}A\,.\,B) \overset{\text{(I)}}{\equiv} [\![s_1]\!] \to \ldots \to [\![s_n]\!]$ we have that $n \geq 2$ and $\textsc{Nf}(B) \equiv [\![s_2]\!] \to \ldots \to [\![s_n]\!]$. Also, $[\![s_1]\!] =_\beta A$ so x is of supertype $*^s$. Thus, $x \not\equiv [\![s_i]\!]$ for each $1 \leq i \leq n$. Now $\textsc{Nf}(\sigma) \equiv \textsc{Nf}(B[x := M_k]) \equiv \textsc{Nf}(B) \equiv [\![s_2]\!] \to \ldots \to [\![s_n]\!]$ so (i) and (ii-a) hold. Furthermore $\Gamma_\Phi(\Sigma)^+ \vdash M_k : [\![s_1]\!]$ since the application $(vM_1 \ldots M_{k-1})\, M_k$ is well typed in $\Gamma_\Phi(\Sigma)^+$. ❑

Lemma 3.14. $\Gamma_\Phi(\Sigma)^+ \vdash M : [\![s]\!] \Longrightarrow M \in [\![\mathcal{T}(\Sigma)]\!]_\beta$.

Proof. First we prove the lemma for normal forms. Let N be a normal form and let $\Gamma_\Phi(\Sigma)^+ \vdash N : [\![s]\!]$. Let

$$N \equiv \lambda x_1{:}A_1\,.\,\ldots \lambda x_n{:}A_n\,.\,\Pi y_1{:}B_1\,.\,\ldots \Pi y_m{:}B_m\,.\,vM_1 \ldots M_k$$

We do induction on the structure of N. Assume that the lemma holds for all subterms of N.

$n=0$ since there do not exist terms A and B such that $\Pi x{:}A\,.\,B =_\beta [\![s]\!] \in \mathbb{V}$ (use G3). Furthermore $m=0$ since there is no sort ς such that $[\![s]\!] =_\beta \varsigma$.

Let $0 \leq i \leq k$. From repeated application of G3 it follows that

$$\Gamma_\Phi(\Sigma)^+ \vdash vM_1 \ldots M_{i-1} : \sigma$$

for some term σ. By Lemma 3.13(i) $\textsc{Nf}(\sigma) \equiv [\![s_{i,1}]\!] \to \ldots \to [\![s_{i,m(i)}]\!]$ so $\Gamma_\Phi(\Sigma)^+ \vdash M_i : [\![s_{i,1}]\!]$ and $\Gamma_\Phi(\Sigma)^+ \vdash v : [\![s_{0,1}]\!] \to \ldots \to [\![s_{0,m(0)}]\!]$. Now it follows from the induction hypothesis that $M_i \in [\![\mathcal{T}(\Sigma)]\!]_\beta$, say $M_i = [\![t_i]\!]$. Furthermore $v = [\![f]\!]$ for some $f \in \Sigma$. Now $[\![f(t_1, \ldots, t_n)]\!] = N$ so $N \in [\![\mathcal{T}(\Sigma)]\!]_\beta$.

The general case follows from SR:

$$\begin{aligned} \Gamma_\Phi(\Sigma)^+ \vdash M : [\![s]\!] &\overset{\text{SR}}{\Longrightarrow} \Gamma_\Phi(\Sigma)^+ \vdash \textsc{Nf}(M) : [\![s]\!] \\ &\overset{3.14}{\Longrightarrow} \textsc{Nf}(M) \in [\![\mathcal{F}(\Sigma)]\!]_\beta \\ &\Longrightarrow M \in [\![\mathcal{F}(\Sigma)]\!]_\beta. \end{aligned}$$

❑

The next lemma states that proof terms do not occur in types. It suffices to prove this fact for types that are in normal form.

Lemma 3.15. *Let* $\Gamma \vdash M : \varsigma$ *and let* $\Gamma \vdash v :: *^p$ *then* $v \notin \textsc{Fv}(\textsc{Nf}(M))$.

Proof. Let $v \in \mathbb{V}$. We do induction to the derivation of $\Gamma \vdash \textsc{Nf}(M) : \varsigma$.

- Start. Now $\varsigma \in \{*^s, *^p\}$ and $M \equiv \textsc{Nf}(M) \in \mathbb{V}$. $M \not\equiv v$ because $\Gamma \vdash v :: *^p$ so $v \notin \textsc{Fv}(\textsc{Nf}(M))$.

- Π-Elimination. Let $\textsc{Nf}(M) \equiv F\,a$.

  $$\frac{\Gamma \vdash F : A \to \varsigma \qquad \Gamma \vdash a : A}{\Gamma \vdash F\,a : \varsigma}\ \Pi\text{-Elimination}$$

  From Lemma 2.10 it follows that $F \in \mathbb{V} \cup \lambda\mathbb{V}{:}\mathbb{T}.\,\mathbb{T}$. However, if $F \in \lambda\mathbb{V}{:}\mathbb{T}.\,\mathbb{T}$ then $F\,a$ is a redex, contradicting the assumption that $\textsc{Nf}(M) \equiv F\,a$. So $F \in \mathbb{V}$. Now $v \in \textsc{Fv}(F)$ iff $v \equiv F$ which is ruled out by the fact that v and F have different types.

  Furthermore it follows from Lemma 2.8 that $\Gamma \vdash A : *^s$. By Lemma 3.14 we know that $a \equiv [\![t]\!]$ for some $t \in \mathcal{T}(\Sigma)$, so $v \notin \textsc{Fv}(a)$. Now we are done since $\textsc{Fv}(F\,a) = \textsc{Fv}(F) \cup \textsc{Fv}(a)$.

- Π-Introduction. Impossible, product types are not β-convertible with sorts.

- Π-Formation. Let $\textsc{Nf}(M) \equiv \Pi x{:}A.\,B$.

  $$\frac{\Gamma \vdash A : \varsigma_1 \qquad \Gamma, x{:}A \vdash B : \varsigma_2}{\Gamma \vdash \Pi x{:}A.\,B : \varsigma}\ \langle \varsigma_1, \varsigma_2, \varsigma \rangle \in \mathcal{R}$$

  By the induction hypothesis we know that $v \notin \textsc{Fv}(A)$ and $v \notin \textsc{Fv}(B)$. Since $\textsc{Fv}(\Pi x{:}A.\,B) \subseteq \textsc{Fv}(A) \cup \textsc{Fv}(B)$ we have that $v \notin \textsc{Fv}(\Pi x{:}A.\,B)$ and we are done.

Axiom holds vacuously, Weakening and β-Conversion are trivial applications of the induction hypothesis. ❑

Lemma 3.16. *Let $\Gamma \supseteq \Gamma_\Phi(\Sigma)^+$ and $A, B \in \mathbb{T}_\Gamma$. Suppose $\Gamma \vdash A : *^p$, then*

$$\Pi x{:}A.\,B \in [\![\mathcal{F}(\Sigma)]\!]_\beta \iff A, B \in [\![\mathcal{F}(\Sigma)]\!]_\beta \text{ and } x \notin \textsc{Fv}(\textsc{Nf}(B)).$$

Proof. By induction over Definition 3.4 on page 12.
($\Longrightarrow$) $\textsc{Nf}(\Pi x{:}A.\,B) \equiv \Pi x{:}\textsc{Nf}(A).\,\textsc{Nf}(B) \not\equiv \Pi P{:}[\![s]\!] \to *^p.\,(P[\![t_1]\!]) \to (P[\![t_2]\!])$ so $\textsc{Nf}(\Pi x{:}A.\,B) \not\equiv [\![t_1 \stackrel{s}{=} t_2]\!]$. Similarly we have that $\textsc{Nf}(\Pi x{:}A.\,B) \not\equiv [\![\forall x.\,\phi]\!]$. Consequently $\textsc{Nf}(\Pi x{:}A.\,B) \equiv [\![\phi]\!] \to [\![\psi]\!]$ for some $\phi, \psi \in \mathcal{F}(\Sigma)$. Now $\textsc{Nf}(A) \equiv [\![\phi]\!]$, $\textsc{Nf}(B) \equiv [\![\psi]\!]$ and $x \notin \textsc{Fv}(\textsc{Nf}(B))$. By definition we have $A, B \in [\![\mathcal{F}(\Sigma)]\!]_\beta$.
($\Longleftarrow$) $\textsc{Nf}(\Pi x{:}A.\,B) \equiv \textsc{Nf}(A \to B) \equiv \textsc{Nf}(A) \to \textsc{Nf}(B) \equiv [\![\phi]\!] \to [\![\psi]\!] \equiv [\![\phi {\to} \psi]\!]$ for some $\phi, \psi \in \mathcal{F}(\Sigma)$. ❑

The next lemma treats formulas of the form $\forall x \,.\, \phi$.

Lemma 3.17. *Let* $\Gamma \supseteq \Gamma_\Phi(\Sigma)^+$ *and* $A, B \in \mathbb{T}_\Gamma$. *Suppose* $\Gamma \vdash A : *^s$, *then*

$$\Pi x{:}A \,.\, B \in [\![\mathcal{F}(\Sigma)]\!]_\beta \iff B \in [\![\mathcal{F}(\Sigma)]\!]_\beta \text{ and } \mathrm{NF}(A) \equiv [\![s]\!]$$

for some $s \in S$.

Proof.
($\Longrightarrow$) $\mathrm{NF}(\Pi x{:}A \,.\, B) \not\equiv [\![t_1 \overset{s}{=} t_2]\!]$ and $\mathrm{NF}(\Pi x{:}A \,.\, B) \not\equiv [\![\phi_1 \to \phi_2]\!]$ so $\mathrm{NF}(\Pi x{:}A \,.\, B) \equiv \Pi x{:}\mathrm{NF}(A) \,.\, \mathrm{NF}(B) \equiv \Pi x{:}[\![s]\!] \,.\, [\![\phi]\!]$ for some $s \in \mathcal{S}$ and $\phi \in \mathcal{F}(\Sigma)$. This implies $\mathrm{NF}(A) \equiv [\![s]\!]$ and $\mathrm{NF}(B) \equiv [\![\phi]\!]$.
($\Longleftarrow$) Let $\mathrm{NF}(B) \equiv [\![\phi]\!]$, then $\mathrm{NF}(\Pi x{:}A \,.\, B) \equiv \Pi x{:}[\![s]\!] \,.\, B \equiv [\![\forall x \,.\, \phi]\!]$ so $\Pi x{:}A \,.\, B \in [\![\mathcal{F}(\Sigma)]\!]_\beta$. ❑

Lemma 3.18. *Let* $\Gamma \supseteq \Gamma_\Phi(\Sigma)^+$ *and* $A, B \in \mathbb{T}_\Gamma$. *Suppose* $\Gamma \vdash A : \triangle$, *then*

$$\Pi x{:}A \,.\, B \in [\![\mathcal{F}(\Sigma)]\!]_\beta \iff \mathrm{NF}(\Pi x{:}A \,.\, B) \equiv [\![t_1 \overset{s}{=} t_2]\!]$$

for some $t_1, t_2 \in \mathcal{T}_s(\Sigma)$ *and for some* $s \in S$.

Proof. Analogous to 3.16 and 3.17. ❑

4 Soundness and completeness

In this section we prove that the representation, given in Section 3 is sound and complete. That is, $\Phi \Vdash \phi$ iff $\Gamma_\Phi(\Sigma)^+ \vdash M : [\![\phi]\!]$ for some term M. The proof from left to right (soundness) is straightforward.

Completeness of the representation, i.e. the statement from right to left, is complicated. In case M is the representation of some derivation $\mathcal{D}$, completeness is immediate since then $\mathcal{D}$ is a proof for ϕ. However, in general M is not the representation of a derivation. We prove that we can, in that case, replace M by a term M' that is the interpretation of a derivation. In some cases M' is obtained by normalising M. But in general it is not enough to normalise M in order obtain the interpretation of a derivation. The simplest example of a normal form that is not the interpretation of a derivation is based on the absence of η-reduction in our system:

$$\Gamma_\Phi(\Sigma) \vdash \lambda P{:}[\![s]\!] \to *^p \,.\, v_{t_1 \overset{s}{=} t_2} P : [\![t_1 \overset{s}{=} t_2]\!]$$

if $t_1 \overset{s}{=} t_2 \in \Phi$. Here the proofterm $\lambda P \,.\, v\, P$ is a simple η-expansion of $v \equiv [\![\mathbf{P}(t_1 \overset{s}{=} t_2)]\!]$. More complex examples of normal forms that are not the interpretation of a derivation contain what we will call β_L-redexes or η_L-redexes. These are redexes that are *interweaved* with an application of Leibniz equality. We come back to this in the next section.

In informal explanations of the different concepts we sometimes use PRED2 denotations instead of λL denotations — using that $\lambda\mathrm{L} \subset \lambda\mathrm{PRED2} \cong \mathrm{PRED2}$ — in order to improve the readability. If convenient, we abbreviate the PRED2 formula $\forall P\,.\,P(t_1) \to P(t_2)$ by $t_1 \stackrel{s}{=} t_2$. Moreover, we write

$$\frac{}{t \stackrel{s}{=} t} \qquad \text{and} \qquad \frac{P(t_1) \qquad t_1 \stackrel{s}{=} t_2}{P(t_2)}$$

where we actually mean

$$\dfrac{\dfrac{[\ P(t)\ ]^1}{P(t) \to P(t)}\,1}{\forall P\,.\,P(t) \to P(t)} \qquad \text{and} \qquad \dfrac{P(t_1) \qquad \dfrac{\forall Q\,.\,Q(t_1) \to Q(t_2)}{P(t_1) \to P(t_2)}}{P(t_2)}$$

4.1 Leibniz reductions

We introduce η_L-reductions to deal with η-redexes — possibly interweaved with Leibniz equality — and β_L-reductions to deal with β-redexes that are interweaved with Leibniz equality. For β-redexes that are not interweaved with Leibniz equality, ordinary β-reduction suffices.

First we introduce η_L-reductions. Since an η-expansion of proofterm $\lambda x{:}A\,.\,f\,x$ with A of type $*^s$ or $*^p$ represents a derivation iff f itself represents a derivation, these η-expansions are not bothering us. Hence we may assume that A is of type $\triangle$. Thus, the λ-term corresponding to the following PRED2 derivation is a typical example of an η-redex:

$$\dfrac{\dfrac{\dfrac{\forall Q\,.\,Q(t) \to Q(t_1)}{P(t) \to P(t_1)} \qquad t_1 \stackrel{s}{=} t_2}{P(t) \to P(t_2)}}{\forall P\,.\,P(t) \to P(t_2)}$$

Actually, this is not just an η-redex but an η_L-redex, because there is an application of the substitution rule between the elimination and the introduction of the second order variable P. We added this substitution just to show that such interweaving with Leibniz equality is not complicating matters in the η-case. The idea behind η_L-reduction is that we instantiate the second order variable Q with $\boldsymbol{\lambda} x\,.\,t \stackrel{s}{=} x$. The last step of the derivation then becomes a $\to$-elimination instead of a $\forall$-introduction. Thus we get:

$$\dfrac{\dfrac{}{t \stackrel{s}{=} t} \qquad \dfrac{\dfrac{\forall Q\,.\,Q(t) \to Q(t_1)}{t \stackrel{s}{=} t \to t \stackrel{s}{=} t_1} \qquad t_1 \stackrel{s}{=} t_2}{t \stackrel{s}{=} t \to t \stackrel{s}{=} t_2}}{\forall P\,.\,P(t) \to P(t_2)}$$

and the η_L-redex disappeared. Id est, the $\forall$-introduction of a second order variable preceded by an $\forall$-elimination where was instantiated with that particular second order variable is not present anymore. In Lemma 4.1 we present the formalisation of this idea.

Lemma 4.1. *There exists a type preserving function* $\eta_L : \mathbb{T} \longrightarrow \mathbb{T}$ *such that* $\eta_L(\lambda x{:}A\,.\,b) \notin \lambda\mathbb{V}{:}\mathbb{T}\,.\,\mathbb{T}$ *for all* A, b *such that*

(i) $\Gamma \vdash A : \triangle$

(ii) $\Gamma \vdash \lambda x{:}A\,.\,b : [\![\phi]\!]$

for some context Γ *and for some formula* ϕ.

Proof. Let Γ and ϕ be such that (i) and (ii) are satisfied. By Lemma 2.9(iii), $A \equiv [\![s]\!] \to *^p$. G5 gives that $\Gamma, x{:}A \vdash b : B$ and $\mathrm{NF}(\Pi x{:}A\,.\,B) \equiv [\![\phi]\!]$. Hence $\mathrm{NF}(B) \equiv x[\![t_1]\!] \to x[\![t_2]\!]$, for some $t_1, t_2 \in \mathcal{T}_s(\Sigma)$. Abbreviate $\lambda z{:}[\![s]\!]\,.\,\Pi P{:}A\,.\,P[\![t_1]\!] \to P\,z$ by Q and define

$$\eta_L : X \longmapsto \mathrm{NF}(X\,Q\,[\![\,\mathbf{R}\,t_1]\!])$$

We show that $\eta_L(\lambda x{:}A\,.\,b) \notin \lambda\mathbb{V}{:}\mathbb{T}\,.\,\mathbb{T}$. Assume that

$$b \equiv \lambda x_1{:}A_1\,.\,\ldots\lambda x_n{:}A_n\,.\,\Pi y_1{:}B_1\,.\,\ldots\Pi y_m{:}B_m\,.\,vM_1\ldots M_k.$$

Suppose $v \equiv x$. Now $k \in \{0,1\}$ because x is of type $[\![s]\!]{\to}*^p$. If $k = 0$ then the type of $vM_1\ldots M_k$ is not a sort so $m = 0$. However, $\mathcal{R}$ does not contain a rule of the form $\langle \_, \triangle, \_\rangle$ necessary to build $\lambda x{:}A\,.\,\lambda x_1{:}A_1\,.\,\ldots\lambda x_n{:}A_n\,.\,v$. Contradiction, so $k = 1$. Now $m > 0$ is possible but does (according to $\mathcal{R}$) not infer the type $*^p$ of vM_1. Also, $n > 0$ (namely $n = 1$) is possible, but in both cases ($n = 0$ or $n = 1$) we can not form the type of $\lambda x{:}A\,.\,b$. That is, $\mathcal{R}$ does not contain rules $(\triangle, \triangle, \_)$ or $(\triangle, \square^p, \_)$, needed to construct the type of $\lambda x{:}A\,.\,b$ in case $n = 0$ and $n = 1$ respectively. We conclude that $v \not\equiv x$.

Suppose that $n = 0$. If $m > 0$ then, by G4, $B =_\beta \varsigma$. This contradicts $\mathrm{NF}(B) \equiv x[\![t_1]\!] \to x[\![t_2]\!]$, so $m = 0$. Now

$$\begin{aligned}
\eta_L(\lambda x{:}A\,.\,b) &= \mathrm{NF}((\lambda x{:}A\,.\,b)\,Q\,[\![\,\mathbf{R}\,t_1]\!]) \\
&= \mathrm{NF}(b[x := Q]\,[\![\,\mathbf{R}\,t_1]\!]) \\
&= \mathrm{NF}(v\,M_1[x := Q]\ldots M_k[x := Q]\,[\![\,\mathbf{R}\,t_1]\!]) \\
&= v\,\mathrm{NF}(M_1[x := Q])\ldots\mathrm{NF}(M_k[x := Q])[\![\,\mathbf{R}\,t_1]\!] \in \mathbb{T}\,\mathbb{T}
\end{aligned}$$

Next, assume that $n = 1$. Similar to the previous case we have that $m = 0$ (If $m > 0$ then $\mathrm{NF}(B) \equiv A \to \varsigma$ for some sort ς, etc.) We distinguish between two cases:

- $v \equiv x_1$. Now $k = 0$ because v does not have a product type. Thus

$$\begin{aligned}
&\mathrm{NF}((\lambda x{:}A\,.\,b)\,Q\,([\![\,\mathbf{R}\,t_1]\!])) \\
\equiv\;& \mathrm{NF}((\lambda x{:}A\,.\,\lambda x_1{:}A_1\,.\,x_1)\,Q\,([\![\,\mathbf{R}\,t_1]\!])) \\
\equiv\;& \mathrm{NF}((x_1[x := Q][x_1 := [\![\,\mathbf{R}\,t_1]\!]]) \\
\equiv\;& \mathrm{NF}([\![\,\mathbf{R}\,t_1]\!]) \\
\equiv\;& \mathrm{NF}(\mathit{refl}_s\,[\![t_1]\!]) \in \mathbb{T}\;\mathbb{T}
\end{aligned}$$

- $v \not\equiv x_1$. Write $[\theta]$ for $[x := Q][x_1 := [\![\, \mathbf{R}\, t_1]\!]]$, then

$$
\begin{aligned}
\mathrm{N{\scriptsize F}}((\lambda x\!:\!A\,.\,b)\,Q\,([\![\,\mathbf{R}\,t_1]\!])) &\equiv \mathrm{N{\scriptsize F}}((\lambda x\!:\!A\,.\,\lambda x_1\!:\!A_1\,.\,vM_1\ldots M_k)\,Q\,([\![\,\mathbf{R}\,t_1]\!]))\\
&\equiv \mathrm{N{\scriptsize F}}((vM_1\ldots M_k)[x := Q][x_1 := [\![\,\mathbf{R}\,t_1]\!]])\\
&\equiv \mathrm{N{\scriptsize F}}((vM_1\ldots M_k)[\theta])\\
&\equiv \mathrm{N{\scriptsize F}}(vM_1[\theta]\ldots M_k[\theta])\\
&\equiv v\mathrm{N{\scriptsize F}}(M_1[\theta])\ldots \mathrm{N{\scriptsize F}}(M_k[\theta]) \notin \lambda\mathbb{V}\!:\!\mathbb{T}\,.\,\mathbb{T}
\end{aligned}
$$

❑

We say that A *η_L-reduces to B* iff (A,B) is in the compatible closure of the relation $\{(x,\eta_L(x)) \mid x \in \mathbb{T}\}^{\circledast}$. Note that β-normal forms η_L-reduce to β-normal forms.

Example 4.2. Suppose $t \in \Sigma_s$ and let $H \equiv \lambda P\!:\![\![s]\!]\!\to\!*^p\,.\,\lambda h\!:\!P\,[\![t]\!]\,.\,h$ then $\Gamma(\Sigma) \vdash H : [\![t \stackrel{s}{=} t]\!]$ so according to Lemma 4.1 $\eta_L(H)$ should not be a λ-abstraction. Indeed

$$
\begin{aligned}
\eta_L(H) &= \eta_L(\lambda P\!:\![\![s]\!]\!\to\!*^p\,.\,\lambda h\!:\!P\,[\![t]\!]\,.\,h)\\
&= (\lambda P\!:\![\![s]\!]\!\to\!*^p\,.\,\lambda h\!:\!P\,[\![t]\!]\,.\,h)\,[\![\boldsymbol{\lambda} x.t \stackrel{s}{=} x]\!]\,[\![\,\mathbf{R}\ t]\!]\\
&= [\![\,\mathbf{R}\ t]\!] = \mathit{refl}_s\,[\![t]\!]
\end{aligned}
$$

which is an application. ❑

The idea of η_L-reduction is that it removes abstractions of the form $\lambda x\!:\!A\,.\,b$ with A of type $\triangle$. In most cases, those abstraction are unnecessary (as we explained in the beginning of this section). Actually, the reason for adding the variables refl_s is to get rid of the only case (namely, reflexivity proofs) where abstractions of the form $\lambda x\!:\!A\,.\,b$ with A of type $\triangle$ were needed. If we had defined $[\![\,\mathbf{R}\ t]\!]\ \ \lambda P\!:\![\![s]\!]\!\to\!*^p\,.\,\lambda h\!:\!P\,[\![t]\!]\,.\,h$ then $\eta_L([\![\,\mathbf{R}\ t]\!])\ {}_\beta\ [\![\,\mathbf{R}\ t]\!]$, so η_L would not be normalising. In the set-up with variables refl_s we do have strong normalisation of η_L since the number of "wrong" abstractions decreases after each step.

Next we introduce β_L-reduction. Normalisation of β_L-reduction is more involved than normalisation of η_L-reduction. In order to prove normalisation of β_L-reduction we assign to each term $M \in \mathbb{T}_\Gamma$ a complexity $\gamma^\Gamma(M) \in \mathbb{N} \longrightarrow \mathbb{N}$ and prove that this complexity is decreasing by each β_L-reduction step.

Definition 4.3. *The relation $\sqsubset$ on $\mathbb{N} \longrightarrow \mathbb{N}$ is defined by $f \sqsubset g$ iff there exists an $x \in \mathbb{N}$ such that*

(i) *$f(y) = g(y)$ for all $y > x$,*

(ii) *$f(x) < g(x)$.*

This relation[2] is not well-founded. For instance $\{f_i\}_{i=0}^{\infty}$ with

$$f_i(x) = \begin{cases} 0 & \text{if } x < i \\ 1 & \text{if } x \geq i \end{cases}$$

is an infinite decreasing sequence. However, we are only interested in the restriction of this relation to the set $\mathbb{F} \subset \mathbb{N} \longrightarrow \mathbb{N}$ of functions with finite support[3], since all functions of the form $\gamma^{\Gamma}(M)$ have a finite support. This follows immediately from the finiteness of λ-term M and inspection of Definition 4.6. The restriction of $\sqsubset$ to $\mathbb{F}$ is well-founded.

We identify the function $f \in \mathbb{F}$ with the sequence $[\, f(0), \ldots , f(n-1)\,]$, where $n \in \mathbb{N}$ is the least element such that $f(x) = 0$ for all $x \geq n$. The function $\mathit{Shift} : \mathbb{F} \longrightarrow \mathbb{F}$ is defined by

$$\mathit{Shift}(f) : x \longmapsto \begin{cases} 0 & \text{if } x = 0 \\ f(x-1) & \text{if } x > 0 \end{cases}$$

We write $f \bullet g$ for the function $\boldsymbol{\lambda} x \,.\, f(x) \bullet g(x)$, where $\bullet$ is an arbitrary operation on $\mathbb{N}$. Furthermore $n \bullet f$ stands for $\boldsymbol{\lambda} x \,.\, n \bullet f(x)$ for all $n \in \mathbb{N}$.

Lemma 4.4. *Let $f, g, h : \mathbb{F}$. Then*

(i) $f \sqsubset g \iff f + h \ \sqsubset\ g + h$

(ii) $\mathit{Shift}(f) = \varepsilon \iff f = \varepsilon$

(iii) $\mathit{Shift}(f + g) = \mathit{Shift}(f) + \mathit{Shift}(g)$

Proof. Easy. ❑

Next, we define the *complexity* $\gamma^{\Gamma}(M)$ of $M \in \mathbb{T}_{\Gamma}$ under valuation ϑ. The complexity function $\gamma^{\Gamma}(..)$ makes use of an auxiliary function $\chi^{\Gamma}(..)$ that detects predicates:

$$\chi^{\Gamma}(A) = \begin{cases} [\,1\,] & \text{if } \Gamma \vdash A :: \triangle \\ \varepsilon & \text{otherwise} \end{cases}$$

Lemma 4.5. *Suppose $\Gamma \vdash a : A$, then*

$$\chi^{\Gamma, x:A}(M) = \chi^{\Gamma}(M[x := a]) \tag{5}$$

for all $M \in \mathbb{T}_{\Gamma, x:A}$.

Proof. $\chi^{\Gamma,x:A}(M)$ and $\chi^{\Gamma}(M[x := a])$ are both either ϵ or $[\,1\,]$ depending on the supertypes of M and $M[x := a]$ respectively. Thus, the lemma follows immediately from Lemma 2.6 which states that supertypes are not affected by substitution. ❑

[2] $\sqsubset$ is a partial ordering on $\mathbb{N} \longrightarrow \mathbb{N}$. However, we do not need this fact

[3] the support of a function f is the set $\{x \in \mathbb{N} \mid f(x) \neq 0\}$

Since the left-hand-side of (5) does not depend on term a we also have that

$$\chi^{\Gamma}(M[x := a_1]) = \chi^{\Gamma}(M[x := a_2]) \tag{6}$$

for all terms a_1 and a_2 of type A in context Γ.

Let $\vartheta : \mathbb{V} \longrightarrow \mathbb{F}$ be a valuation, assigning functions with finite support to variables.

Definition 4.6. *Let $M \in \mathbb{T}_{\Gamma}$. The complexity $\gamma^{\Gamma}(M) \in \mathbb{F}$ is defined by induction over the structure of M.*

- $\gamma^{\Gamma}_{\vartheta}(x) = \begin{cases} \vartheta(x) & \text{if } \Gamma \vdash x :: *^p \\ \varepsilon & \text{otherwise} \end{cases}$ *for all* $x \in \mathbb{V} \cup \mathbb{C}$
- $\gamma^{\Gamma}_{\vartheta}(\lambda x{:}A\,.\,b) = \gamma^{\Gamma,x:A}_{\vartheta}(b)$
- $\gamma^{\Gamma}_{\vartheta}(\Pi x{:}A\,.\,B) = \varepsilon$
- $\gamma^{\Gamma}_{\vartheta}(A\,B) = \gamma^{\Gamma}_{\vartheta}(B) + \mathit{Shift}^{\vartheta}(\gamma^{\Gamma}_{\vartheta}(A)) + \chi^{\Gamma}(B)$

We omit the subscritpt ϑ in $\gamma^{\Gamma}_{\vartheta}(..)$ if it is irrelevant.

The complexity remains ε until an application $F\,P$ with P a predicate occurs. Then, a $[\,1\,]$ is produced and shifted to the right one step each time the expression occurs as the left component of an application. Thus counting the number of terms to which F is applied.

Lemma 4.7. *Let $\Gamma \vdash M : \sigma$. Then*

$$\gamma^{\Gamma}(M) \neq \varepsilon \Longrightarrow \Gamma \vdash M :: *^p$$

Proof. By induction to the derivation of $\Gamma \vdash M : \sigma$.

- *Weakening*. Immediate, $\gamma^{\Gamma}(M)$ does not depend on Γ_2, if $\Gamma = \Gamma_1 \uplus \Gamma_2$ and $M \in \mathbb{T}_{\Gamma_1}$.
- Π-*Introduction*. Suppose

  $$\frac{\Gamma, x{:}A \vdash b : B \qquad \Gamma \vdash \Pi x{:}A\,.\,B : \varsigma}{\Gamma \vdash \lambda x{:}A\,.\,b : \Pi x{:}A\,.\,B}$$

  and assume that $\gamma^{\Gamma}(\lambda x{:}A\,.\,b) \neq \varepsilon$, then $\gamma^{\Gamma,x:A}(b) \neq \varepsilon$. Now $\Gamma, x{:}A \vdash B : *^p$ and hence $\varsigma = *^p$ is forced by $\mathcal{R}$.
- Π-*Elimination*. Suppose

  $$\frac{\Gamma \vdash F : \Pi x{:}A\,.\,B \qquad \Gamma \vdash a : A}{\Gamma \vdash F\,a : B[x := a]}$$

  and assume that $\gamma^{\Gamma}(F\,a) \neq \varepsilon$. By Correctness of Types [Bar92] $\Gamma \vdash \Pi x{:}A\,.\,B : \varsigma$ for some $\varsigma \in \mathcal{S}$. We distinguish between three cases:

- $\mathit{Shift}(\gamma^\Gamma(F)) \neq \varepsilon$. Now $\Gamma \vdash \Pi x{:}A.B :: *^p$ follows immediately from the induction hypothesis. Quod non.
- $\gamma^\Gamma(a) \neq \varepsilon$. From the induction hypothesis we derive that $\Gamma \vdash A :: *^p$. By Lemma 2.7(i), $\Gamma, x{:}A \vdash B : *^p$. From the Substitution Lemma [Bar92] it follows that $\Gamma \vdash B[x := a] : *^p$, which we had to prove.
- $\chi^\Gamma(a) \neq \varepsilon$. From G4 and $\Gamma \vdash A : \triangle$ we derive that $\Gamma x{:}A \vdash B : \varsigma'$ for some ς' such that $(\triangle, \varsigma', \varsigma) \in \mathcal{R}$. Now $\varsigma' = *^p$ so we are done.

β-Conversion is trivial. Finally *Axiom*, *Start* and *Π-Formation* hold vacuously, since $\gamma^\Gamma(x) = \varepsilon$ for all $x \in \mathbb{V} \cup \mathbb{C}$. ❑

Lemma 4.8. *Suppose* $\Gamma \vdash a_i : A$ *for* $i = 1, 2$, *then*

$$\gamma^\Gamma(a_1) = \gamma^\Gamma(a_2) \Longrightarrow \gamma^\Gamma(M[x := a_1]) = \gamma^\Gamma(M[x := a_2]).$$

for all pseudo-terms M *with* $\mathrm{FV}(M) \subseteq \Gamma, x{:}A$.

Proof. By induction to the structure of M. We treat the only non-trivial case, namely the case that M is an application. Assume that $\gamma^\Gamma(a_1) = \gamma^\Gamma(a_2)$ and suppose $M \equiv M_1 M_2$, then

$$\begin{aligned}
&\gamma^\Gamma(M[x := a_1]) \\
&= \gamma^\Gamma((M_1 M_2)[x := a_1]) \\
&= \gamma^\Gamma(M_1[x := a_1] M_2[x := a_1]) \\
&= \mathit{Shift}(\gamma^\Gamma(M_1[x := a_1])) + \gamma^\Gamma(M_2[x := a_1]) + \chi^\Gamma(M_2[x := a_1]) \\
&\overset{\text{IH}}{=} \mathit{Shift}(\gamma^\Gamma(M_1[x := a_2])) + \gamma^\Gamma(M_2[x := a_2]) + \chi^\Gamma(M_2[x := a_1]) \\
&\overset{(6)}{=} \mathit{Shift}(\gamma^\Gamma(M_1[x := a_2])) + \gamma^\Gamma(M_2[x := a_2]) + \chi^\Gamma(M_2[x := a_2]) \\
&= \gamma^\Gamma(M_1[x := a_2] M_2[x := a_2]) \\
&= \gamma^\Gamma((M_1 M_2)[x := a_2]) \\
&= \gamma^\Gamma(M[x := a_2])
\end{aligned}$$

❑

Corollary 4.9. $\gamma^\Gamma(x) = \gamma^\Gamma(a) \Longrightarrow \gamma^\Gamma(M) = \gamma^\Gamma(M[x := a])$

Corollary 4.9 is also an immediate consequence of the following equation, which describes the natural connection between valuation and substitution:

$$\gamma^\Gamma_\vartheta(M[x := a]) = \gamma^\Gamma_{\vartheta[x \mapsto \gamma^\Gamma_\vartheta(a)]}(M) \tag{7}$$

Namely, if $\gamma^\Gamma_\vartheta(x) = \gamma^\Gamma_\vartheta(a)$, then $\vartheta[x \mapsto \gamma^\Gamma_\vartheta(a)] = \vartheta[x \mapsto \gamma^\Gamma_\vartheta(x)] = \vartheta[x \mapsto \vartheta(x)] = \vartheta$, so $\gamma^\Gamma_\vartheta(M[x := a]) = \gamma^\Gamma_{\vartheta[x \mapsto \gamma^\Gamma_\vartheta(a)]}(M) = \gamma^\Gamma_\vartheta(M)$. However, 4.9 is sufficient for our purposes so we will not prove (7).

Lemma 4.10. *Let $M, M' \in \mathbb{T}_\Gamma$ and $M \to_\beta M'$. There exists a valuation ϑ such that $\gamma^\Gamma_\vartheta(M') \sqsubseteq \gamma^\Gamma_\vartheta(M)$.*

Proof. Suppose that M' is obtained from M by contracting the redex

$$(\lambda x{:}A.\, b)\, a \to_\beta b[x := a].$$

Let ϑ' be an arbitrary valuation and define $\vartheta = \vartheta'[x \mapsto \gamma^\Gamma_{\vartheta'}(a)]$, then we have $\gamma^\Gamma_\vartheta(a) = \gamma^\Gamma_{\vartheta'}(a)$ because $x \notin \mathrm{FV}(a)$, and hence

$$\gamma^\Gamma_\vartheta(x) = \vartheta(x) = \vartheta'[x \mapsto \gamma^\Gamma_{\vartheta'}(a)](x) = \gamma^\Gamma_{\vartheta'}(a) = \gamma^\Gamma_\vartheta(a) \tag{8}$$

if $\Gamma \vdash A : *^p$. Obviously, (8) also holds if not $\Gamma \vdash A : *^p$ since then both $\gamma^\Gamma_\vartheta(x)$ and $\gamma^\Gamma_\vartheta(a)$ are equal to ϵ (using Definition 4.6 and Lemma 4.7). We apply Lemma 4.9 and conclude that

$$\gamma^\Gamma_\vartheta(b[x := a]) = \gamma^\Gamma_\vartheta(b) \sqsubseteq \mathit{Shift}(\gamma^\Gamma_\vartheta(b)) + \gamma^\Gamma_\vartheta(a) = \gamma^\Gamma_\vartheta((\lambda x{:}A.\, b)\, a).$$

Suppose M'' is the (pseudo-)term that is obtained from M by replacing the redex $(\lambda x{:}A.\, b)\, a$ by a fresh variable y, then $M''[y := (\lambda x{:}A.\, b)\, a] \equiv M$ and $M''[y := b[x := a]] \equiv M'$ so we can apply Lemma 4.8 and we are done. ❑

If $\Gamma \vdash \wp : [\![t_1 \stackrel{s}{=} t_2]\!]$ then $\Gamma \vdash \mathit{sym}_s\, [\![t_1]\!]\, [\![t_2]\!]\, \wp : [\![t_2 \stackrel{s}{=} t_1]\!]$. In the sequel we abbreviate $\mathit{sym}_s\, [\![t_1]\!]\, [\![t_2]\!]\, \wp$ by $\wp^{-1}$.

Lemma 4.11. $\gamma^\Gamma_\vartheta(\wp^{-1}) = \mathit{Shift}^3(\vartheta(\mathit{sym}_s)) + \gamma^\Gamma_\vartheta(\wp)$.

Proof. By a straightforward calculation. ❑

In practice, we will always choose a valuation ϑ such that $\vartheta(\mathit{sym}_s) = \epsilon$ for all $s \in S$. In that case we have that

$$\gamma^\Gamma_\vartheta(\wp^{-1}) = \gamma^\Gamma_\vartheta(\wp). \tag{9}$$

(9) would not be valid for any valuation if we could not dispose over the variables sym_s: if $\wp^{-1} \equiv \wp\, [\![\boldsymbol{\lambda} x.x \stackrel{s}{=} t_1]\!]\, [\![\mathbf{R}\, t_1]\!]$, then $\gamma^\Gamma(\wp) \sqsubset \gamma^\Gamma(\wp^{-1})$. In fact, this is precisely the reason for introducing the variables sym_s. Valuations ϑ such that $\vartheta(\mathit{sym}_s) = \epsilon$ for all $s \in S$ are called *symmetry resistant.*

Lemma 4.12. *Let $\Gamma \supseteq \Gamma_\Phi(\Sigma)$ and let $\Gamma \vdash \wp : [\![t_1 \stackrel{s}{=} t_2]\!]$. Suppose*

$$\Gamma \vdash \wp\, P\, h\, \alpha : \sigma \tag{10}$$

then $\mathrm{NF}(P) \equiv \lambda n{:}A.\, \Pi y{:}B_1.\, B_2$ for some $A, B_1, B_2 \in \mathbb{T}_\Gamma$.

Proof. $\Gamma \vdash P : [\![s]\!] \to *^p : \triangle$ so by Lemma 2.10 we have that $P \in \mathbb{V} \cup \lambda\mathbb{V}{:}\mathbb{T}.\, \mathbb{T}$. If $P \in \mathbb{V}$ then $\Gamma \vdash \wp\, P\, h : P[\![t_2]\!]$ with $P[\![t_2]\!]$ not β-convertible with a product type (by Lemma 2.3). This contradicts the assumption that $\wp\, P\, h$ can be applied to α, so $P \in \lambda\mathbb{V}{:}\mathbb{T}.\, \mathbb{T}$, say $P \equiv \lambda n{:}A.\, B$. Then, $\Gamma \vdash \wp\, P\, h : B[n := [\![t_2]\!]]$ and for the same reason as in the previous case $\mathrm{NF}(B[n := [\![t_2]\!]])$ is a product type. Hence $\mathrm{NF}(B)$ is, according to 2.11, itself is a product type, so we are done. ❑

Next we introduce β_L-reduction. Consider the following derivation:

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}[A]^1\\ \vdots\\ \phi(t_1)\end{array}}{A \Rightarrow \phi(t_1)}\,1 \qquad t_1 \stackrel{s}{=} t_2}{A \Rightarrow \phi(t_2)} \qquad A}{\phi(t_2)}$$

This derivation contains an $\Rightarrow$-introduction, followed by an $\Rightarrow$-elimination. Furthermore, there is an application of Leibniz equality interweaved with this introduction and elimination rule. Thus, the corresponding λ-term is a typical example of a β_L-redex. In the following lemma we deal with such β_L-redexes.

Lemma 4.13. *There exists a type preserving function* $\beta_L : \mathbb{T} \longrightarrow \mathbb{T}$ *and a valuation* ϑ *such that if* $\Gamma \vdash \wp : [\![t_1 \stackrel{s}{=} t_2]\!]$ *and*

$$\Gamma \vdash \wp\, P\, h\, \alpha : \sigma \tag{11}$$

then the following conditions are satisfied:

(i) $\Gamma \vdash \beta_L(\wp\, P\, h\, \alpha) : \sigma$

(ii) $\gamma^{\Gamma}(\beta_L(\wp\, P\, h\, \alpha)) \sqsubset \gamma^{\Gamma}(\wp\, P\, h\, \alpha)$

(iii) η_L*-normality is preserved.*

Proof. Assume that $\Gamma \vdash \wp : [\![t_1 \stackrel{s}{=} t_2]\!]$ and $\Gamma \vdash \wp\, P\, h\, \alpha : \sigma$. By Lemma 4.12 $P \equiv \lambda n{:}A\,.\,\Pi x{:}B_1\,.\,B_2$ for some $A, B_1, B_2 \in \mathrm{NF}(\mathbb{T}_\Gamma)$. Note that $\Gamma \vdash A : *^s$ and $\Gamma, n{:}A \vdash \Pi x{:}B_1\,.\,B_2 : *^p$. By G4, there must be sorts ς and ς' such that $\langle \varsigma, \varsigma', *^p \rangle \in \mathcal{R}$ and $\Gamma \vdash B_1 : \varsigma$. We distinguish between the three cases, corresponding to the three rules in $\mathcal{R}$ of the form $\langle \_, *^p \rangle$. In all these three cases we build an alternative proof term of lower complexity and prove that this alternative proof term is still a normal form. First we calculate the complexity of the "old" proof term, i.e. the proof term in (11) with respect to a symmetry resistant valuation ϑ:

$$\begin{aligned}
&\gamma^{\Gamma}_{\vartheta}(\wp\, P\, h\, \alpha) = \\
&\mathit{Shift}(\gamma^{\Gamma}_{\vartheta}(\wp\, P\, h)) + \gamma^{\Gamma}_{\vartheta}(\alpha) + \chi^{\Gamma}(\alpha) = \\
&\mathit{Shift}(\mathit{Shift}(\gamma^{\Gamma}_{\vartheta}(\wp\, P)) + \gamma^{\Gamma}_{\vartheta}(h) + \underbrace{\chi^{\Gamma}(h)}_{\epsilon}) + \gamma^{\Gamma}_{\vartheta}(\alpha) + \chi^{\Gamma}(\alpha) = \\
&\mathit{Shift}(\mathit{Shift}(\mathit{Shift}(\gamma^{\Gamma}_{\vartheta}(\wp)) + \underbrace{\gamma^{\Gamma}_{\vartheta}(P)}_{\epsilon} + \underbrace{\chi^{\Gamma}(P)}_{[\,1\,]}) + \gamma^{\Gamma}_{\vartheta}(h)) + \gamma^{\Gamma}_{\vartheta}(\alpha) + \chi^{\Gamma}(\alpha) = \\
&\mathit{Shift}(\mathit{Shift}(\mathit{Shift}(\gamma^{\Gamma}_{\vartheta}(\wp)) + [\,1\,]) + \gamma^{\Gamma}_{\vartheta}(h)) + \gamma^{\Gamma}_{\vartheta}(\alpha) + \chi^{\Gamma}(\alpha) = \\
&\mathit{Shift}^3(\gamma^{\Gamma}_{\vartheta}(\wp) + [\,1\,]) + \mathit{Shift}^2(\gamma^{\Gamma}_{\vartheta}(h)) + \gamma^{\Gamma}_{\vartheta}(\alpha) + \chi^{\Gamma}(\alpha) = \\
&[\,0\,0\,1\,] + \mathit{Shift}^3(\gamma^{\Gamma}_{\vartheta}(\wp)) + \mathit{Shift}^2(\gamma^{\Gamma}_{\vartheta}(h)) + \gamma^{\Gamma}_{\vartheta}(\alpha) + \chi^{\Gamma}(\alpha)
\end{aligned}$$

Next, we calculate the complexities of the alternative proof term for σ. We treat the three possible types for B_1 separately:

($*^p$) By Lemma 3.15 we have that $x \notin \mathrm{FV}(B_2)$ so $B \equiv B_1 \to B_2$ and hence we have $\Gamma \vdash \lambda n{:}A.\, B_2 : A \to *^p$. Now

$$\Gamma \vdash \wp\,(\lambda n{:}A.\, B_2)\,(h\,(\wp^{-1}\,(\lambda n{:}A.\, B_1)\,\alpha)) : \sigma \tag{12}$$

If h is a λ-abstraction term, say $h \equiv \lambda y{:}B[n := t_2].\, h'$, then (12) contains a redex. According to Lemma 4.10 we may assume that contracting this redex does not increase the complexity. Namely, by choosing the valuation for y such that $\gamma_\vartheta^\Gamma(\wp^{-1}\,(\lambda n{:}A.\, B_1)\,\alpha) \sqsubset \vartheta(y)$. Note that $\gamma_\vartheta^\Gamma(\wp^{-1}\,(\lambda n{:}A.\, B_1)\,\alpha)$ does not depend on $\vartheta(y)$ so this can indeed be done.

If h is not a λ-abstraction term then (12) is normal. We define $\beta_L(\wp\, P\, h\, \alpha) = \mathrm{NF}(\wp\,(\lambda n{:}A.\, B_2)\,(h\,(\wp^{-1}\,(\lambda n{:}A.\, B_1)\,\alpha)))$ so (i) is satisfied by SR. This proof term in (12) is of complexity

$$\begin{aligned}&\gamma_\vartheta^\Gamma(\wp\,(\lambda n{:}A.\, B_2)\,(h\,(\wp^{-1}\,(\lambda n{:}A.\, B_1)\,\alpha))) = \\ &\mathit{Shift}(\gamma_\vartheta^\Gamma(\wp\,(\lambda n{:}A.\, B_2))) + \gamma_\vartheta^\Gamma(h\,(\wp^{-1}\,(\lambda n{:}A.\, B_1)\,\alpha))\end{aligned}$$

since $\chi^\Gamma(h\,(\wp^{-1}\,(\lambda n{:}A.\, B_1)\,\alpha)) = \epsilon$.

However,

$$\begin{aligned}&\gamma_\vartheta^\Gamma(h\,(\wp^{-1}\,(\lambda n{:}A.\, B_1)\,\alpha)) = \\ &\mathit{Shift}(\gamma_\vartheta^\Gamma(h)) + \gamma_\vartheta^\Gamma(\wp^{-1}\,(\lambda n{:}A.\, B_1)\,\alpha) + \underbrace{\chi^\Gamma(\wp^{-1}\,(\lambda n{:}A.\, B_1)\,\alpha)}_{\epsilon} = \\ &\mathit{Shift}(\gamma_\vartheta^\Gamma(h)) + \mathit{Shift}(\gamma_\vartheta^\Gamma(\wp^{-1}\,(\lambda n{:}A.\, B_1))) + \gamma_\vartheta^\Gamma(\alpha) + \underbrace{\chi^\Gamma(\alpha)}_{\epsilon}\end{aligned}$$

and $(\wp^{-1}\,(\lambda n{:}A.\, B_1))$ and $(\wp\,(\lambda n{:}A.\, B_2))$ both have complexity $\mathit{Shift}(\gamma_\vartheta^\Gamma(\wp)) + [\,1\,]$ since $\gamma_\vartheta^\Gamma(\wp) = \gamma_\vartheta^\Gamma(\wp^{-1})$, so

$$\begin{aligned}&\gamma_\vartheta^\Gamma(\wp\,(\lambda n{:}A.\, B_2)\,(h\,(\wp^{-1}\,(\lambda n{:}A.\, B_1)\,\alpha))) = \\ &2 \cdot \mathit{Shift}(\mathit{Shift}(\gamma_\vartheta^\Gamma(\wp)) + [\,1\,]) + \mathit{Shift}(\gamma_\vartheta^\Gamma(h)) + \gamma_\vartheta^\Gamma(\alpha) = \\ &[\,0\,2\,] + \mathit{Shift}^2(\gamma_\vartheta^\Gamma(\wp)) + \mathit{Shift}(\gamma_\vartheta^\Gamma(h)) + \gamma_\vartheta^\Gamma(\alpha)\end{aligned}$$

which is obviously smaller than $\gamma_\vartheta^\Gamma(\wp\, P\, h\, \alpha)$.

If h is not a λ-abstraction, then the proof term in (12) is a normal form so (ii) holds. If h is a λ-abstraction, say $h \equiv \lambda y{:}B_1[n := t_2].\, h'$, then $h'[y := \wp^{-1}(\lambda n{:}A.\, B_1)\,\alpha]$ is a normal form since h' is a normal form and $\wp^{-1}(\lambda n{:}A.\, B_1)\,\alpha$ is an application in normal form. So normalisation consists in this case of contracting exactly *one* β-redex, of which we know that it does not increase the complexity.

For the validity of (iii) we have to verify that no equality proofs of the form $\lambda x{:}A'.\,M$ with A' of type $\triangle$ are formed. The only λ-abstraction terms that we introduce — besides copies of already existing ones — are the predicates $\lambda n.B_1$ and $\lambda n.B_2$ that are obviously not equality proofs.
A λ-abstraction term that already existed might also turn into a η_L-redex. This would be the case if its type is instantiated in such a way that it becomes the interpretation of an equality formula. However, the only instantiation occurring in the normalisation of (12) is on the level of proof terms and such terms do not occur in a type, as stated by Lemma 3.15.

($*^s$) From Lemma 2.9 (i) it follows that $B_1 \equiv [\![s']\!]$ for some $s' \in S$. Thus, $B_1 \in \mathbb{V}$ and $n \notin \mathrm{FV}(B_1)$. Now $\lambda n{:}A.\,B_2[x := \alpha] \in \mathbb{T}_\Gamma$ and

$$\Gamma \vdash \wp\,(\lambda n{:}A.\,B_2[x := \alpha])\,(h\,\alpha) : [\![\phi]\!]. \tag{13}$$

Define $\beta_L(\wp\,P\,h\,\alpha) = \wp\,(\lambda n{:}A.\,B_2[x := \alpha])\,(h\,\alpha)$ in the ($*^s$)-case, then (i) is valid. Again, we calculate the complexities:

$$\begin{aligned}
&\gamma^\Gamma_\vartheta(\wp\,(\lambda n{:}A.\,B_2[x := \alpha])\,(h\,\alpha)) = \\
&\mathit{Shift}\,(\gamma^\Gamma_\vartheta(\wp\,(\lambda n{:}A.\,B_2[x := \alpha]))) + \underbrace{\gamma^\Gamma_\vartheta(h\,\alpha)}_{\epsilon} + \underbrace{\chi^\Gamma(h\,\alpha)}_{\epsilon} = \\
&\mathit{Shift}\,(\mathit{Shift}\,(\gamma^\Gamma_\vartheta(\wp)) + \underbrace{\gamma^\Gamma_\vartheta(\lambda n{:}A.\,B_2[x := \alpha])}_{\epsilon} + \underbrace{\chi^\Gamma(\lambda n{:}A.\,B_2[x := \alpha])}_{[\,1\,]}) = \\
&[\,0\,1\,] + \mathit{Shift}^2(\gamma^\Gamma_\vartheta(\wp))
\end{aligned}$$

which is smaller than $\gamma^\Gamma_\vartheta(\wp\,P\,h\,\alpha)$. Since the proof term in (13) is a normal form we can immediately conclude that we do not create η_L-redexes.

($\triangle$) Similar to the ($*^s$)-case.

❑

We say that A *β_L-reduces to B* iff (A, B) is in the compatible closure of the relation $\{(x, \beta_L(x)) \mid x \in \mathbb{T}\}$. Since $(\mathbb{F}, \sqsubset)$ is well-founded, and β_L-reduction decreases the complexity, there does not exist an infinite sequence $\{{\beta_L}^i(M)\}_{i=0}^{\infty}$. In other words, eventually we obtain a λ-term ${\beta_L}^k(M)$ such that the conditions for Lemma 4.13 are not satisfied. We let ${\beta_L}^*(M)$ be the result of applying β_L sufficiently many times, i.e. β_L is the β_L-normalform of M. Analogously, we let ${\eta_L}^*(M)$ denote the η_L-normalform of M.

Since both η_L and β_L preserve β-normality, and moreover, β_L preserves η_L-normality, we can find a β, β_L, η_L-normal form of M by calculating ${\beta_L}^*({\eta_L}^*(\mathrm{NF}(M)))$. We call this expression the *super normal form*.

Note that $\gamma^\Gamma_\vartheta(M)$ of super normal form M is not necessarily $\sqsubset$-minimal.[4]

[4]In fact: if $M \in \mathbb{T}_\Gamma$ is a super normal form, then $\gamma^\Gamma_\vartheta(M) < [\,0\,0\,1\,]$

Theorem 4.14. (Conservativity). *Let $\phi \in \mathcal{F}(\Sigma)$ and suppose that $\Gamma_\Phi(\Sigma)^+ \vdash M : [\![\phi]\!]$ for some super-normal form M, then there exists a normal proof tree $\mathcal{D}$ such that $M \equiv [\![\mathcal{D}]\!]$ and $\mathcal{D}$ proves $\Phi \Vdash \phi$.*

Proof. We prove the slightly more general statement, where $[\![\phi]\!]$ is replaced by an arbitrary term σ such that $\textsc{Nf}(\sigma) \equiv [\![\phi]\!]$. Thus distributing the induction step for the *β-conversion* rule over the other rules. We apply induction on the derivation of $\Gamma_\Phi(\Sigma)^+ \vdash M : \sigma$. Note that $\sigma \in [\![\mathcal{F}(\Sigma)]\!]_\beta$.

- M is built from *Axiom*. Impossible because $(M, \sigma) \notin \mathcal{A}$ since $M, \sigma \notin \mathcal{S}$.
- M is built from *Start*.

  $$\frac{\Gamma \vdash \sigma : *^p}{\Gamma_\Phi(\Sigma)^+ \vdash M : \sigma}\ \textit{Start}$$

  where $\Gamma_\Phi(\Sigma)^+ \equiv \Gamma, M{:}\sigma$ and $M \in \mathbb{V}$. From Definition 3.6 it follows that $\sigma \equiv \textsc{Nf}(\sigma) \equiv [\![\phi]\!]$. Now $M \equiv [\![\mathbf{P}\phi]\!]$ and $\mathbf{P}\phi$ proves $\Phi \Vdash \phi$.
- M is built from *Weakening*.

  $$\frac{\Gamma \vdash M : \sigma \qquad \Gamma \vdash A : \varsigma}{\Gamma_\Phi(\Sigma)^+ \vdash M : \sigma}$$

  where $\Gamma_\Phi(\Sigma)^+ \equiv \Gamma, x{:}A$. From $x{:}A \in \Gamma_\Phi(\Sigma)^+$ it follows that $\varsigma \in \{*^s, *^f, *^p, \Box^s\}$. If $\varsigma \equiv \Box^s$, say $A = [\![s]\!]$, then $s \in S$ is not used in Σ. Consider Σ as $S \setminus \{s\}$-sorted signature and apply the induction hypothesis. Similarly, if $\varsigma \equiv *^s$ or $\varsigma \equiv *^f$ then Σ contains a symbol that is not used, etc. If $\varsigma \equiv *^p$ then $A = [\![\psi]\!]$ for some $\psi \in \Phi$. Now by induction there is a prooftree $\mathcal{D}$ such that $M \equiv [\![\mathcal{D}]\!]$ and $\mathcal{D}$ proves $\Phi \backslash \{\psi\} \Vdash \phi$. Use *Thinning*.
- M is built from Π-*Introduction*. Say $\sigma \equiv \Pi x{:}A\,.\,B$.

  $$\frac{\Gamma_\Phi(\Sigma)^+, x{:}A \vdash b : B \qquad \Gamma_\Phi(\Sigma)^+ \vdash \Pi x{:}A\,.\,B : \varsigma}{\Gamma_\Phi(\Sigma)^+ \vdash \lambda x{:}A\,.\,b : \Pi x{:}A\,.\,B}$$

  By G4 we have $\Gamma_\Phi(\Sigma)^+ \vdash A : \varsigma'$ and $\Gamma_\Phi(\Sigma)^+, x{:}A \vdash B : \varsigma''$ for some $\varsigma', \varsigma'' \in \mathcal{S}$ such that $\langle \varsigma', \varsigma'', \varsigma \rangle \in \mathcal{R}$. $\textsc{Nf}(\Pi x{:}A\,.\,B) \equiv [\![\phi]\!]$ so $\varsigma = *^p$ and hence also $\varsigma'' = *^p$. Furthermore $\varsigma' \in \{*^p, *^s, \triangle\}$. We treat those three cases for ς' seperately:

  - $\varsigma' \equiv *^p$. By Lemma 3.16 and $\sigma \in [\![\mathcal{F}(\Sigma)]\!]_\beta$ we know that $A, B \in [\![\mathcal{F}(\Sigma)]\!]_\beta$, say $\textsc{Nf}(A) \equiv [\![\phi_A]\!]$ and $\textsc{Nf}(B) \equiv [\![\phi_B]\!]$ with $\phi_A, \phi_B \in \mathcal{F}(\Sigma)$. Furthermore $x \notin \textsc{Fv}(\textsc{Nf}(B))$ and $\phi \equiv \phi_A \Rightarrow \phi_B$. By the induction hypothesis we have a proof tree $\mathcal{D}$ such that $b \equiv [\![\mathcal{D}]\!]$ and such that $\mathcal{D}$ proves $\Phi \cup \{\phi_A\} \Vdash \phi_B$. Now $\lambda x{:}A\,.\,b \equiv [\![\mathbf{I}\,\phi_A.\mathcal{D}]\!]$ and $\mathbf{I}\,\phi_A.\mathcal{D}$ proves $\Phi \Vdash \phi_A \Rightarrow \phi_B$ so we are done.
  - $\varsigma' \equiv *^s$. Same kind of argument.

– $\varsigma' \equiv \triangle$. Say $\phi \equiv t_1 \stackrel{s}{=} t_2$. A is a normal form, so $A \equiv [\![s]\!] \to *^p$. Furthermore $\mathrm{NF}(B) \equiv x[\![t_1]\!] \to x[\![t_2]\!]$. By G5,

$$\Gamma_\Phi(\Sigma)^+, x{:}[\![s]\!] \to *^p \vdash b : B'$$

for some term B' such that $B' =_\beta B$. $b \in \lambda\mathbb{V}{:}\mathbb{T}.\mathbb{T}$ since M is η_L-normal. Let $b \equiv \lambda y{:}C.b'$. One easily verifies that $C \equiv x[\![t_1]\!]$. We do a case distinction on b'.

  - $b' \in \mathbb{V}$. From $\Gamma_\Phi(\Sigma)^+, x{:}[\![s]\!] \to *^p, y{:}x[\![t_1]\!] \vdash b' : x[\![t_2]\!]$ and Definition 3.1 it follows that $b' \equiv y$, so $t_1 \equiv t_2$ and hence $\phi \equiv t_1 \stackrel{s}{=} t_1$ which is proved by $\mathbf{R}\, t_1$.
  - $b' \in \mathbb{T}\mathbb{T}$. Excluded by η_L-normality of M.
  - $b' \in \lambda\mathbb{V}{:}\mathbb{T}.\mathbb{T}$. Impossible, b' does not have a Π-type.
  - $b' \in \Pi\mathbb{V}{:}\mathbb{T}.\mathbb{T}$. Impossible, the type of b' is not a sort.

• M is built from Π-*Elimination*.

$$\frac{\Gamma_\Phi(\Sigma)^+ \vdash F : \Pi y{:}A.B \qquad \Gamma_\Phi(\Sigma)^+ \vdash a : A}{\Gamma_\Phi(\Sigma)^+ \vdash Fa : B[y := a] \equiv \sigma}$$

F is not an abstraction because Fa is in normal form. Assume that F is an application, say $G_1 a_1$ then G_1 is not an abstraction either. Assume that G_1 is an application, say $G_2 a_2$ and so on. We end up with a finite sequence of applications $G_n a_n \cdots a_1$ such that G_n is a variable ($n \geq 1$). From Definition 3.1 it follows that $G_n \equiv H_\varphi$ for some $\varphi \in \Phi$. We distinguish between four cases:

– $n < \propto(\varphi)$. Write C_j for the type of $G_n a_n \cdots a_j$ in $\Gamma_\Phi(\Sigma)^+$, for all $0 \leq j \leq n+1$, then ($a_0 \equiv a$)

$$\Gamma_\Phi(\Sigma)^+ \vdash G_n : C_{n+1} \qquad \Gamma_\Phi(\Sigma)^+ \vdash F : C_1 \qquad \Gamma_\Phi(\Sigma)^+ \vdash F\ a : C_0$$

We show by downwards induction that $C_j \in [\![\mathcal{F}(\Sigma)]\!]_\beta$, i.e. there exists a formula ϕ_j, such that $\mathrm{NF}(C_j) \equiv [\![\phi_j]\!]$, and such that there is a proof tree $\mathcal{D}_j$ for $\Phi \Vdash \phi_j$ for all $n+1 \geq j \geq 0$. For $j = n+1$ this holds: $\mathrm{NF}(C_{n+1}) \equiv [\![\varphi]\!]$ and $\mathbf{P}\varphi$ proves $\Phi \Vdash \varphi$.

Assume that $C_{j+1} \in [\![\mathcal{F}(\Sigma)]\!]_\beta$, say $\mathrm{NF}(C_{j+1}) \equiv [\![\phi_{j+1}]\!]$ and $\mathcal{D}_j$ proves $\Phi \Vdash \phi_{j+1}$ for some proof tree $\mathcal{D}_{j+1}$. We make a case distinction on ϕ_{j+1}.

  - $\phi_{j+1} \equiv \psi_j \Rightarrow \phi_j$. Now $\mathrm{NF}(C_j) \equiv \phi_j$ and $\Gamma_\Phi(\Sigma)^+ \vdash a_j : [\![\psi_j]\!]$. By induction there is a prooftree $\mathcal{D}'_j$ for $\Phi \Vdash \phi_j$. Define $\mathcal{D}_j = \mathcal{D}_{j+1}\mathcal{D}'_j$ then $\mathcal{D}_j$ proves $\Phi \Vdash \phi_j$.
  - $\phi_{j+1} \equiv \forall x_s.\psi_j$. Now $\mathrm{NF}(C_j) \equiv \psi_j$ and $\Gamma_\Phi(\Sigma)^+ \vdash a_j : [\![s]\!]$. From Lemma 3.14 it follows that $a_j \equiv [\![t]\!]$ for some $t \in \mathcal{T}_s(\Sigma)$. Define $\mathcal{D}_j = \mathcal{D}_{j+1}t$ then $\mathcal{D}_j$ proves $\Phi \Vdash \phi_j$.
  - $\psi_{j+1} \equiv [\![t_1 \stackrel{s}{=} t_2]\!]$. Now $\Gamma_\Phi(\Sigma)^+ \vdash G_n \underbrace{a_n \ldots a_{j+1}}_{n-j} : [\![t_1 \stackrel{s}{=} t_2]\!]$. This means that $\propto(\varphi) = n - j$ and hence $n < n - j$ contradicting $j \geq 0$.

This finishes the induction step. We conclude that $\mathcal{D}_j$ proves $\Phi \Vdash \phi_j$ for all $0 \leq j \leq n+1$. In particular $\mathcal{D}_0$ proves $\Phi \Vdash \phi_0$ which is what we had to prove because $\phi_0 \equiv \phi$.

– $n = \propto(\varphi)$. Now $\Gamma_\Phi(\Sigma)^+ \vdash G_n a_n \cdots a_1 : [\![t_1 \stackrel{s}{=} t_2]\!]$ for some $t_1, t_2 \in \mathcal{T}_s(\Sigma)$. $[\![\phi]\!] =_\beta B[y := a] =_\beta a[\![t_1]\!] \to a[\![t_2]\!]$ so $a[\![t_1]\!] \to a[\![t_2]\!] \in [\![\mathcal{F}(\Sigma)]\!]_\beta$. From Lemma 3.16 it follows that $a[\![t_1]\!], a[\![t_2]\!] \in [\![\mathcal{F}(\Sigma)]\!]_\beta$. By Lemma 3.12 we have that $a[\![x]\!] \in [\![\mathcal{F}(\Sigma)]\!]_\beta$ for any arbitrary fresh variable $x \in \mathbb{V}_s$. Suppose $\textsc{Nf}(a[\![x_s]\!]) \equiv [\![\psi]\!]$, with $\psi \in \mathcal{F}(\Sigma)$. Applying Lemma 3.11 with $b \equiv a[\![x]\!]$ gives

$$\begin{aligned} \textsc{Nf}(a[\![t_i]\!]) &\equiv \textsc{Nf}(a[\![x]\!][[\![x]\!] := [\![t_i]\!]]) \\ &\equiv \textsc{Nf}(a[\![x]\!])[[\![x]\!] := [\![t_i]\!]] \\ &\equiv [\![\psi]\!][[\![x]\!] := [\![t_i]\!]] \\ &\equiv [\![\psi[x := t_i]]\!] \end{aligned}$$

for $i = 1, 2$. So

$$\begin{aligned} [\![\phi]\!] &\equiv \textsc{Nf}(a[\![t_1]\!] \to a[\![t_2]\!]) \\ &\equiv \textsc{Nf}(a[\![t_1]\!]) \to \textsc{Nf}(a[\![t_2]\!]) \\ &\equiv [\![\psi[x := t_1]]\!] \to [\![\psi[x := t_2]]\!] \\ &\equiv [\![\psi[x := t_1] \Rightarrow \psi[x := t_2]]\!] \end{aligned}$$

and hence $\phi \equiv \psi[x := t_1] \Rightarrow \psi[x := t_2]$.
By the induction hypothesis, applied to $\Gamma_\Phi(\Sigma)^+ \vdash F : [\![t_1 \stackrel{s}{=} t_2]\!]$, we have a proof tree $\mathcal{D} : \Phi \Vdash t_1 \stackrel{s}{=} t_2$. *Thinning* gives us a proof tree
$\mathcal{D} : \Phi \cup \{\psi[x := t_1]\} \Vdash t_1 \stackrel{s}{=} t_2$. Furthermore
$\mathbf{P}\psi[x := t_1] : \Phi \cup \{\psi[x := t_1]\} \Vdash \psi[x := t_1]$. Now
$\mathbf{I}\,\psi[x := t_1].\,\mathbf{S}[\boldsymbol{\lambda} x.\psi]\,.(\mathbf{P}\psi[x := t_1])/\mathcal{D} : \Phi \Vdash \psi[x := t_1] \Rightarrow \psi[x := t_2]$.

– $n = \propto(\varphi) + 1$. Same kind of arguments.

– $n > \propto(\varphi) + 1$. Contradicting β_L-normality of M.

– M is build from *β-Conversion*.

$$\frac{\Gamma_\Phi(\Sigma)^+ \vdash N : \sigma'}{\Gamma_\Phi(\Sigma)^+ \vdash N : \sigma} \; \sigma =_\beta \sigma'$$

Now $\textsc{Nf}(\sigma') \equiv \textsc{Nf}(\sigma) \equiv [\![\phi]\!]$. By the induction hypothesis we have that $\mathcal{D}$ proves $\Phi \Vdash \phi$ and we are done.

❑

Next we prove the soundness of our implementation. That is, we can construct a term $t : [\![\phi]\!]$ in context $\Gamma_\Phi(\Sigma)^+$ whenever there is a proof tree $\mathcal{D} : \Phi \Vdash \phi$. (Actually, $t \equiv [\![\mathcal{D}]\!]$ does the job.) This phraseology stems from the object in view. We construct derivations in the type theory[5] where we are actually interested in derivability in the logic. Theorem 4.15 garantees us the existence of a λ-term of type $[\![\phi]\!]$ whenever ϕ is provable in the logic.

Theorem 4.15. *Let $\Phi \subseteq \mathcal{F}(\Sigma, V)$ be a set of equations and $\phi \in \mathcal{F}(\Sigma, V)$, then*

$$\mathcal{D} : \Phi \Vdash \phi \Longrightarrow \Gamma_\Phi(\Sigma)^+ \vdash [\![\mathcal{D}]\!] : [\![\phi]\!].$$

Proof. Induction on the stucture of $\mathcal{D}$.

- $\mathcal{D} \equiv \mathbf{P}\phi$. Now $[\![\mathcal{D}]\!] \equiv v_\phi$ and $v_\phi : [\![\phi]\!] \in \Gamma_\Phi(\Sigma)^+$. Use Start.

- $\mathcal{D} \equiv \mathbf{R}\, t$. Suppose $t \in \mathcal{T}_s(\Sigma)$. Now $\phi \equiv t \stackrel{s}{=} t$ and by Definition 3.8 $[\![\mathcal{D}]\!] \equiv \lambda P{:}[\![s]\!] \to *^p .\, \lambda h{:}P[\![t]\!] .\, h$ which is of type $\Pi P{:}[\![s]\!] \to *^p .\, P[\![t]\!] \to P[\![t]\!] \equiv [\![t \stackrel{s}{=} t]\!] \equiv [\![\phi]\!]$.

- $\mathcal{D} \equiv \mathbf{S}[\boldsymbol{\lambda} x.\psi]\,.\mathcal{D}_1/\mathcal{D}_2$. Say $\mathcal{D}_2 : t_1 \stackrel{s}{=} t_2$. Now $\phi \equiv \psi[x := t_2]$ and $\mathcal{D}_1 : \Phi \Vdash \psi[x := t_1]$. Furthermore $[\![\mathcal{D}]\!] \equiv [\![\mathcal{D}_2]\!][\![\boldsymbol{\lambda} x.\psi]\!][\![\mathcal{D}_1]\!]$ by Definition 3.8. By the induction hypothesis we know that $\Gamma_\Phi(\Sigma)^+ \vdash [\![\mathcal{D}_1]\!] : [\![\psi[x := t_1]]\!]$ and $\Gamma_\Phi(\Sigma)^+ \vdash [\![\mathcal{D}_2]\!] : [\![t_1 \stackrel{s}{=} t_2]\!](\equiv \Pi P{:}[\![s]\!] \to *^p .\, P[\![t_1]\!] \to P[\![t_2]\!])$. Now $\Gamma_\Phi(\Sigma)^+ \vdash [\![\mathcal{D}_2]\!][\![\boldsymbol{\lambda} x.\psi]\!] : [\![\boldsymbol{\lambda} x.\psi]\!][\![t_1]\!] \to [\![\boldsymbol{\lambda} x.\psi]\!][\![t_2]\!]$. By β-conversion we find $\Gamma_\Phi(\Sigma)^+ \vdash [\![\mathcal{D}_2]\!][\![\boldsymbol{\lambda} x.\psi]\!] : [\![\psi]\!][x := [\![t_1]\!]] \to [\![\psi]\!][x := [\![t_2]\!]]$. But $[\![\psi]\!][x := [\![t_1]\!]] \to [\![\psi]\!][x := [\![t_2]\!]] \equiv [\![\psi[x := t_1]]\!] \to [\![\psi[x := t_2]]\!]$ by Lemma 3.10 so $\Gamma_\Phi(\Sigma)^+ \vdash [\![\mathcal{D}_2]\!][\![\boldsymbol{\lambda} x.\psi]\!][\![\mathcal{D}_1]\!] : [\![\psi[x := t_2]]\!]$ follows by Π-elimination.

- $\mathcal{D} \equiv \mathbf{I}\, \phi_1.\mathcal{D}_1$. Say $\phi \equiv \phi_1 \Rightarrow \phi_2$ and $\mathcal{D}_1 : \Phi \cup \{\phi_1\} \Vdash \phi_2$. By the induction hypothesis we know that $\Gamma_{\Phi \cup \{\phi_1\}}(\Sigma) \vdash [\![\mathcal{D}_1]\!] : [\![\phi_2]\!]$. By definition $\Gamma_{\Phi \cup \{\phi_1\}}(\Sigma) \equiv \Gamma_\Phi(\Sigma)^+, v_{\phi_1} : [\![\phi_1]\!]$. From Π-introduction we conclude $\Gamma_\Phi(\Sigma)^+ \vdash [\![\, \mathbf{I}\, \phi_1.\mathcal{D}_1]\!] : \Pi v_{\phi_1}{:}[\![\phi_1]\!] .\, [\![\phi_2]\!](\equiv [\![\phi_1]\!] \to [\![\phi_2]\!] \equiv [\![\phi_1 \Rightarrow \phi_2]\!] \equiv [\![\phi]\!])$.

- $\mathcal{D} \equiv \mathcal{D}_1\mathcal{D}_2$. Say $\mathcal{D}_1 : \Phi \Vdash \psi \Rightarrow \phi$ and $\mathcal{D}_2 : \Phi \Vdash \psi$. By the induction hypothesis we know that $\Gamma_\Phi(\Sigma)^+ \vdash [\![\mathcal{D}_1]\!] : [\![\psi \Rightarrow \phi]\!](\equiv [\![\psi]\!] \to [\![\phi]\!])$ and $\Gamma_\Phi(\Sigma)^+ \vdash [\![\mathcal{D}_2]\!] : [\![\psi]\!]$. Now $\Gamma_\Phi(\Sigma)^+ \vdash [\![\mathcal{D}]\!] : [\![\phi]\!]$ by Π-elimination.

- $\mathcal{D} \equiv \mathbf{G}\, x.\mathcal{D}_1$. Say $\mathcal{D}_1 : \Phi \Vdash \psi$ and $x \notin \mathrm{FV}(\Phi)$. By the induction hypothesis we have that $\Gamma_\Phi(\Sigma)^+, x : [\![s]\!] \vdash [\![\mathcal{D}_1]\!] : [\![\psi]\!]$. Note that $x \notin \mathrm{FV}([\![\rho]\!])$ for all $v_\rho : [\![\rho]\!] \in \Gamma_\Phi(\Sigma)^+$ and hence x does not occur in any type of the context. From Π-introduction it follows that
  $\Gamma_\Phi(\Sigma)^+ \vdash [\![\, \mathbf{G}\, x.\mathcal{D}_1]\!] : \Pi x{:}[\![s]\!] .\, [\![\psi]\!](\equiv [\![\forall x.\psi]\!] \equiv [\![\phi]\!])$.

- $\mathcal{D} \equiv \mathcal{D}_1 t$. Say $\mathcal{D}_1 : \Phi \Vdash \forall x.\psi$ and $t \in \mathcal{T}_s(\Sigma)$. Now $\phi \equiv \psi[x := t]$. By the induction hypothesis we have that $\Gamma_\Phi(\Sigma)^+ \vdash [\![\mathcal{D}_1]\!] : [\![\forall x.\psi]\!](\equiv \Pi x{:}[\![s]\!] .\, [\![\psi]\!])$. Furthermore $\Gamma_\Phi(\Sigma)^+ \vdash [\![t]\!] : [\![s]\!]$ because $[\![t]\!] : [\![s]\!] \in \Gamma_\Phi(\Sigma)^+$. Now $\Gamma_\Phi(\Sigma)^+ \vdash [\![\mathcal{D}]\!] : [\![\psi]\!][x := [\![t]\!]](\equiv [\![\psi[x := t]]\!] \equiv [\![\phi]\!])$.

❑

[5] Actually, those type systems that are implemented, are much stronger than λL, so the here presented results are only a first step to a justification of this approach

References

[Bar92] H.P. Barendregt. Lambda calculi with types. In S. Abramsky, D.M. Gabbay, and T.S.E. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 2, pages 117–309. Oxford Science Publications, 1992.

[Ber89] S. Berardi. Some unforeseen features of proof-checkers based on Coquand's calculus of constructions and the other systems in Barendregt's cube. Technical report, Department of Computer Science, CMU and Dipartimento di Matematica, Torino, 1989. extended abstract.

[Geu93] H. Geuvers. *Logics and Type Systems.* PhD thesis, University of Nijmegen, Toernooiveld 1, 6525 ED Nijmegen, The Netherlands, September 1993.

[How80] W.A. Howard. The formulae-as-types notion of construction. In J.R. Hindley and J.P. Seldin, editors, *Essays on Combinatory Logic, Lambda Calculus and Formalism*, pages 479–490. Academic Press, London, 1980. Dedicated to Haskell B. Curry on the occasion of his 80th birthday.

[Sel96] M.P.A. Sellink. *Computer-Aided Verification of Protocols, The Type Theoretic Approach.* PhD thesis, Utrecht University, 1996.

[TF92] H. Tonino and K. Fujita. On the adequacy of representing higher order intuitionistic logic as a pure type system. *Annals of Pure and Applied Logic*, 57:251–276, 1992.