A Shallow Embedding of Pure Type Systems into First-Order Logic*

Łukasz Czajka[1]

1 **DIKU, University of Copenhagen, Copenhagen, Denmark**
luta@di.ku.dk

Abstract

We define a shallow embedding of logical proof-irrelevant Pure Type Systems (piPTSs) into minimal first-order logic. In logical piPTSs a distinguished sort $*^p$ of propositions is assumed. Given a context Γ and a Γ-proposition τ, i.e., a term τ such that $\Gamma \vdash \tau : *^p$, the embedding translates τ and Γ into a first-order formula $\mathcal{F}_\Gamma(\tau)$ and a set of first-order axioms Δ_Γ. The embedding is not complete in general, but it is strong enough to correctly translate most of piPTS propositions (by completeness we mean that if $\Gamma \vdash M : \tau$ is derivable in the piPTS then $\mathcal{F}_\Gamma(\tau)$ is provable in minimal first-order logic from the axioms Δ_Γ). We show the embedding to be sound, i.e., if $\mathcal{F}_\Gamma(\tau)$ is provable in minimal first-order logic from the axioms Δ_Γ, then $\Gamma \vdash M : \tau$ is derivable in the original system for some term M. The interest in the proposed embedding stems from the fact that it forms a basis of the translations used in the recently developed CoqHammer automation tool for dependent type theory.

1998 ACM Subject Classification F.4.1 Mathematical Logic

Keywords and phrases pure type systems, first-order logic, hammers, proof automation, dependent type theory

Digital Object Identifier 10.4230/LIPIcs.TYPES.2016.9

1 Introduction

In this paper we define a shallow embedding of any logical proof-irrelevant Pure Type System into untyped minimal first-order logic. Proof-irrelevant PTSs (piPTSs) extend ordinary PTSs with proof-irrelevance by incorporating it into the conversion rule. In logical piPTSs a distinguished sort $*^p$ of propositions is assumed and some restrictions are put on the rules and axioms of the system. The class of logical piPTSs is fairly broad. In particular, a proof-irrelevant version of the Calculus of Constructions with a separate set universe may be presented as a logical piPTS.

Our embedding is shallow, which means that terms of type $*^p$ are translated directly to first-order formulas. The embedding (or an optimised variant of it) is intended to be used to translate dependent type theory goals to formalisms of automated theorem provers (ATPs) for first-order logic. Hence, it is important for efficiency (i.e. the success rate of the ATPs on translated problems) that the embedding be shallow.

The interest in our embedding is justified by the fact that it is used as a basis of the translations employed in the recently developed CoqHammer tool, which is the first hammer for a proof assistant based on dependent type theory [15]. The embedding presented in this paper is only a small "core" version of the translation used in [15]. In particular, here we

* Supported by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement number 704111.

do not deal at all with inductive types. The translation in [15] handles most of the Coq logic and introduces many optimisations. Consequently, it is quite complex and not easily amenable to a direct theoretical investigation.

The aim of this present paper is to isolate a "core" of the translation from [15] and prove its *soundness*: in a logical piPTS, for any context Γ and a Γ-proposition τ, i.e., a term τ such that $\Gamma \vdash \tau : *^p$, if $\Delta_\Gamma \vdash_{\mathrm{FOL}} \mathcal{F}_\Gamma(\tau)$, i.e., the translation $\mathcal{F}_\Gamma(\tau)$ of τ is derivable in minimal first-order logic from the axioms Δ_Γ which are the translation of Γ, then there exists a term M such that $\Gamma \vdash M : \tau$ in the piPTS. The terminology comes from the hammer and automated reasoning literature [9, 8], where this implication is referred to as soundness and usually formulated in terms of satisfiability. The implication in the other direction, i.e., if $\Gamma \vdash M : \tau$ then $\Delta_\Gamma \vdash_{\mathrm{FOL}} \mathcal{F}_\Gamma(\tau)$, is called *completeness* of the embedding. In type-theoretic literature, e.g. [2, 18], the terminology is flipped. We stick with automated reasoning terminology when referring to soundness or completeness.

Our embedding is not complete, i.e., there exist a context Γ and a Γ-proposition τ such that $\Gamma \vdash M : \tau$ for some M, but $\Delta_\Gamma \nvdash_{\mathrm{FOL}} \mathcal{F}_\Gamma(\tau)$. However, the presented embedding is "complete enough" to be practically usable, i.e., sufficiently many of the derivable Γ-propositions are provable after the translation for the practical purpose of using an extended and optimised version of the embedding in a hammer tool for dependent type theory. Some empirical evidence for this claim is provided in [15, 14] where over 40% of the translations of Coq standard library theorems are reproved by first-order ATPs, using a (substantially) extended and optimised version of the present embedding. In this paper we do not attempt to rigorously justify or even formulate the "complete enough" claim, but only illustrate the (in)completeness on several examples.

The soundness proof is the main result of this paper. We present the result in a general framework of logical proof-irrelevant Pure Type Systems to avoid unnecessary reliance on any particular variant of dependent type theory. Our soundness proof employs constructive proof-theoretic methods. Assuming the decidability of type checking in the original piPTS, our soundness proof implicitly provides an algorithm to transform a natural deduction proof of the translation of a piPTS proposition into a piPTS term inhabiting the proposition.

1.1 Motivation

In order to give some motivation for our work, we now briefly describe the architecture of a hammer and the relation of the embedding in this paper to the translation used in [15]. For more background on hammers see e.g. [15, 9].

The goal of a hammer is, given a context Γ and a Γ-proposition τ, to find a term M such that $\Gamma \vdash M : \tau$. In practice, the context Γ consists of all declarations accessible at a given point from the proof assistant kernel (typically there are thousands or tens of thousands of them). Hammers work in three phases.

1. Lemma selection which heuristically chooses a subset of the accessible declarations that are likely useful for the conjecture τ. These declarations, together with the declarations they depend on, form a context $\Gamma_0 \subseteq \Gamma$. Typically, the size of Γ_0 is on the order of hundreds of declarations.
2. Translation of the conjecture τ together with the context Γ_0 to the input formats of first-order automated theorem provers (ATPs) like Vampire [22] or Eprover [25], and running the ATPs on the translations.
3. Proof reconstruction which uses the information obtained from a successful ATP run to re-prove the conjecture in the logic of the proof assistant or to directly reconstruct the proof term.

The reason for employing first-order ATPs is that they are currently the strongest and most optimised general-purpose automated theorem provers. They are capable of efficiently handling problems with hundreds of axioms, which is necessary for a hammer tool. The use of state-of-the-art first-order ATPs is the reason why shallowness of the embedding is essential, because the ATPs are heavily optimised for directly handling the primitives of first-order logic. For instance, a declaration $x : \tau$, where $\tau = \Pi y : A.py \to qy$ is a Γ-proposition but A is not (for an appropriate context Γ), should be translated directly to a formula of the form $\forall y.T_A(y) \to p(y) \to q(y)$ where p, q are first-order predicates and the first-order predicate $T_A(y)$ states that y has type A. In contrast, it would be much less efficient to use a deep embedding with Γ-propositions translated to first-order terms and using a binary "inhabitation" predicate T, where the above declaration $x : \tau$ would be translated to an axiom $T(x, \mathcal{C}_\Gamma(\tau))$ and a conjecture τ' to $\exists y.T(y, \mathcal{C}_\Gamma(\tau'))$, with $\mathcal{C}_\Gamma(\alpha)$ the translation of a type α to a first-order term. Such a translation would require the ATPs to synthesise first-order terms corresponding to proof terms which would impact the success rate, even if $T(x, \mathcal{C}_\Gamma(\tau))$ was optimised to e.g. $\forall yz.T_A(y) \to T(z, py) \to T(xyz, qy)$.

The translation in [15] is in fact not sound because of some optimisations. Also the ATPs employed in practice are classical. In the proof reconstruction phase in [15] the conjecture is actually re-proved in the logic of Coq using the lemmas which were needed in an ATP proof. This is feasible because there are typically only a few of these lemmas, so a much weaker method than a state-of-the-art ATP may be used in this final phase. Another issue is that the piPTS formalism does not exactly correspond to common variants of type theory because it assumes proof irrelevance. Since proof irrelevance is crucial to our translation, no soundness proof is possible for ordinary PTSs. However, we believe our soundness proof is still valuable for three main reasons. First, it contributes to the general understanding of the extended translation in [15], and in particular to understanding of which aspects of it are "safe" and which might be not. Second, the proof being constructive implicitly provides an algorithm to transform a natural deduction proof of the translation of a conjecture τ into a piPTS term inhabiting τ. A simplified explicit presentation of the algorithm is given in Algorithm 76. It could form a basis of a partial method for source-level proof reconstruction, i.e., a method for translating a proof found by an ATP back into a proof term in the logic of the proof assistant (possibly using the excluded middle axiom). In mature hammer systems optional source-level proof reconstruction increases success rates. Third, isolating a sound "core" of the translation from [15] might help in devising practical translations for other type theories than just the logic of Coq handled in [15].

From the point of view of proof theory, what we here call completeness of the embedding is perhaps more interesting than soundness. However, all hammer tools essentially give up on completeness. From the automated reasoning perspective it is soundness, or at least understanding the reasons for the lack of it, which is more important.

2 First-order logic

We define a proof notation system for minimal first-order intuitionistic logic. This system of notation is a restriction of the system λP_1 from [28, Chapter 8].

▶ **Definition 1.** An individual term (t, s) is a variable (x, y, z) or a function application $(f(t_1, \ldots, t_n))$. A formula (φ, ψ) is an atom $(R(t_1, \ldots, t_n))$, an implication $(\varphi \to \psi)$ or a universally quantified formula $(\forall x.\varphi)$. A proof term (M, N) is a proof variable (X, Y, Z), an individual abstraction $(\lambda x.M)$, a proof abstraction $(\lambda X : \varphi.M)$, an application of a proof term (MN) or of an individual term (Mt). An environment (Δ) is a finite set of proof

$$\Delta, X : \varphi \vdash X : \varphi$$

$$\frac{\Delta, X : \varphi \vdash M : \psi}{\Delta \vdash (\lambda X : \varphi.M) : \varphi \to \psi} \qquad \frac{\Delta \vdash M : \varphi \to \psi \quad \Delta \vdash N : \varphi}{\Delta \vdash MN : \psi}$$

$$\frac{\Delta \vdash M : \varphi}{\Delta \vdash (\lambda x.M) : \forall x \varphi} \; x \notin \mathrm{FV}(\Delta) \qquad \frac{\Delta \vdash M : \forall x \varphi}{\Delta \vdash Mt : \varphi[t/x]}$$

Figure 1 Rules of minimal first-order logic.

variable declarations of the form $X : \varphi$. We usually write $\Delta, X : \varphi$ instead of $\Delta \cup \{X : \varphi\}$. The system of first-order minimal logic is given by the rules in Figure 1. The relation of β-reduction on proof terms is defined as the contextual closure of the following rules.

$$(\lambda x.M)t \quad \to_\beta \quad M[t/x] \qquad\qquad (\lambda X : \varphi.M)N \quad \to_\beta \quad M[N/X]$$

We write $\Delta \vdash_{\mathrm{FOL}} M : \varphi$ to denote derivability in first-order minimal logic. We drop the subscript when obvious. We also omit the proof terms when irrelevant, writing e.g. $\psi, \theta \vdash \varphi$.

▶ **Lemma 2.** *If $\Delta \vdash M : \varphi$ and $\Delta \vdash M : \varphi'$ then $\varphi = \varphi'$.*

For the proofs of the following two theorems see e.g. [28, Chapter 8].

▶ **Theorem 3** (Confluence and strong normalisation)**.** *If $\Delta \vdash M : \varphi$ then M is confluent and strongly normalising (wrt. β-reduction).*

▶ **Theorem 4** (Subject reduction)**.** *If $\Delta \vdash M : \varphi$ and $M \to_\beta^* M'$ then $\Delta \vdash M' : \varphi$.*

Proof terms in *η-long normal form* or *η-lnf* are defined inductively (wrt. an implicit environment).

- If N is an η-lnf of type α then $\lambda x.N$ is an η-lnf of type $\forall x.\alpha$.
- If N is an η-lnf of type β then $\lambda X : \alpha.N$ is an η-lnf of type $\alpha \to \beta$.
- If $N_1, \ldots, N_n$ are η-lnf or individual terms and $XN_1 \ldots N_n$ is of an atom type, then $XN_1 \ldots N_n$ is an η-lnf.

▶ **Lemma 5.** *If $\Delta \vdash M : \varphi$ then there exists N in η-lnf such that $\Delta \vdash N : \varphi$.*

Proof. Take the β-normal form of M and η-expand it as much as possible, respecting the type and introducing no new β-redexes. The easy details are left to the reader. ◀

The *target* of a formula is defined inductively: $\mathrm{target}(R(t_1, \ldots, t_n)) = R$, $\mathrm{target}(\varphi \to \psi) = \mathrm{target}(\psi)$ and $\mathrm{target}(\forall x.\varphi) = \mathrm{target}(\varphi)$.

▶ **Lemma 6.** *If $\Delta \vdash M : R(t_1, \ldots, t_n)$ and M is in η-lnf then there is $(X : \varphi) \in \Delta$ such that $M = XN_1 \ldots N_k$ and $\mathrm{target}(\varphi) = R$ and each N_i is an individual term or a proof term in η-lnf.*

3 Proof-irrelevant Pure Type Systems

In this section we define proof-irrelevant Pure Type Systems. These extend Pure Type Systems with proof-irrelevance, incorporating it into the conversion rule. Our definition of

$$\text{(axiom)} \qquad \langle\rangle \vdash s_1 : s_2 \qquad \text{if } (s_1, s_2) \in \mathcal{A}$$

$$\text{(start)} \qquad \frac{\Gamma \vdash A : s}{\Gamma, x : A \vdash x : A} \qquad \text{if } x \in V^s \setminus \mathrm{dom}(\Gamma)$$

$$\text{(weakening)} \qquad \frac{\Gamma \vdash A : B \quad \Gamma \vdash C : s}{\Gamma, x : C \vdash A : B} \qquad \text{if } x \in V^s \setminus \mathrm{dom}(\Gamma)$$

$$\text{(product)} \qquad \frac{\Gamma \vdash A : s_1 \quad \Gamma, x : A \vdash B : s_2}{\Gamma \vdash (\Pi x : A.B) : s_3} \qquad \text{if } (s_1, s_2, s_3) \in \mathcal{R}$$

$$\text{(application)} \qquad \frac{\Gamma \vdash M : (\Pi x : A.B) \quad \Gamma \vdash N : A}{\Gamma \vdash MN : B[N/x]} \qquad \text{if } N \sim x$$

$$\text{(abstraction)} \qquad \frac{\Gamma, x : A \vdash M : B \quad \Gamma \vdash (\Pi x : A.B) : s}{\Gamma \vdash (\lambda x : A.M) : (\Pi x : A.B)}$$

$$\text{(conversion)} \qquad \frac{\Gamma \vdash M : A \quad \Gamma \vdash B : s}{\Gamma \vdash M : B} \qquad \text{if } B =_{\beta\varepsilon} A$$

Figure 2 Rules of proof-irrelevant PTSs

proof-irrelevant Pure Type Systems is new. It is similar to the definition of a proof-irrelevant version of ECC from [30]. A related treatment of proof-irrelevance for some extensions of the Calculus of Constructions is also present in [5]. The study of the meta-theory of ordinary Pure Type Systems was initiated in [20].

▶ **Definition 7.** The set $\mathcal{T}$ of preterms of a proof-irrelevant Pure Type System (piPTS) is defined by the grammar:

$$\mathcal{T} ::= V^s \mid \mathcal{S} \mid \mathcal{T}\mathcal{T} \mid \lambda V^s : \mathcal{T}.\mathcal{T} \mid \Pi V^s : \mathcal{T}.\mathcal{T} \mid \varepsilon$$

Here $\mathcal{S}$ is a set of sorts, and V^s is a set of variables of sort $s \in \mathcal{S}$. The constant ε represents an arbitrary proof. Its role is technical – it will never occur in well-typed terms. The set $\mathrm{FV}(M)$ of free variables of a preterm M is defined in the usual way. To save on notation we sometimes treat $\mathrm{FV}(M)$ as a list. We use $x, y, z, \ldots$ for variables, $N, M, A, B, \ldots$ for preterms, and $s, s', s_1, s_2, \ldots$ for sorts. We sometimes write x^s to indicate that $x^s \in V^s$. We assume there exists a sort $*^p \in \mathcal{S}$ of propositions.

Note that we tag variables with the sorts of their types, like in [30, 24]. This already appears in [20, 18, 2]. We treat preterms up to α-equivalence, but we do not consider bound variables of different sorts to be α-convertible. For example, if $s_1 \neq s_2$ then $\lambda x^{s_1} : *^p.x^{s_1} \neq_\alpha \lambda x^{s_2} : *^p.x^{s_2}$. Also, whenever we write $\lambda x : A.M$ we assume $x \notin \mathrm{FV}(A)$.

▶ **Definition 8.** The *ε-reduction* is defined as the contextual closure of the rewrite rules:

$$x^{*^p} \to_\varepsilon \varepsilon \qquad \varepsilon M \to_\varepsilon \varepsilon \qquad \lambda x : A.\varepsilon \to_\varepsilon \varepsilon$$

▶ **Definition 9.** A term N is *on the same level as* a variable x, notation $N \sim x$, if one of the following cases holds:

- $x \in V^{*^p}$ and $N \to_\varepsilon^* \varepsilon$, or
- $x \notin V^{*^p}$ and $N \not\to_\varepsilon^* \varepsilon$.

▶ **Definition 10.** We define restricted β-reduction as follows:

$$(\lambda x : A.M)N \quad \to_\beta \quad M[N/x] \qquad \text{if } N \sim x$$

The restriction $N \sim x$ is necessary to ensure confluence of $\beta\varepsilon$-reduction on preterms. Without the restriction, for e.g. $M = (\lambda x^{*^p} : A.x^{*^p})*^p$ we would have $M \to_\varepsilon^* \varepsilon$ and $M \to_\beta *^p$.

▶ **Definition 11.** The *specification* of a proof-irrelevant PTS is a triple $(\mathcal{S}, \mathcal{A}, \mathcal{R})$, where $\mathcal{S}$ is a set of sorts, $\mathcal{A}$ is a set of axioms of the form (s_1, s_2) with $s_1, s_2 \in \mathcal{S}$, and $\mathcal{R}$ is a set of rules of the form (s_1, s_2, s_3) with $s_1, s_2, s_3 \in \mathcal{S}$. We often write (s_1, s_2) for $(s_1, s_2, s_2) \in \mathcal{R}$. A *context* is a finite list of declarations of the form $x : A$, or more formally a function from a finite subset of the set of variables to the set of terms. We denote the empty context by $\langle\rangle$. If $\Gamma = x_1 : A_1, \ldots, x_n : A_n$ then $\mathrm{dom}(\Gamma) = \{x_1, \ldots, x_n\}$ and $\Gamma(x_i) = A_i$. We denote contexts by Γ, Γ', etc. We write $\Gamma' \supseteq \Gamma$ if $\mathrm{dom}(\Gamma) \subseteq \mathrm{dom}(\Gamma')$ and $\Gamma(x) = \Gamma'(x)$ for $x \in \mathrm{dom}(\Gamma)$. A judgement has the form $\Gamma \vdash A : B$. We write $\Gamma \vdash A : B : C$ if $\Gamma \vdash A : B$ and $\Gamma \vdash B : C$. The *proof-irrelevant PTS* (piPTS) determined by the specification $(\mathcal{S}, \mathcal{A}, \mathcal{R})$ is defined by the rules and axioms in Figure 2. We often identify a piPTS with its specification.

▶ **Definition 12.** Let Γ be a context and A a preterm.

1. Γ is *legal* if $\Gamma \vdash M : N$ for some $M, N \in \mathcal{T}$.
2. A is a Γ-*term* if $\Gamma \vdash A : B$ or $\Gamma \vdash B : A$ for some $B \in \mathcal{T}$.
3. A is a Γ-*subject* if $\Gamma \vdash A : B$ for some $B \in \mathcal{T}$.
4. A is a Γ-*type* if $\Gamma \vdash A : s$ for some $s \in \mathcal{S}$.
5. A is a Γ-*proposition* if $\Gamma \vdash A : *^p$.
6. A is a Γ-*proof* if $\Gamma \vdash A : B : *^p$,
7. A is *legal* if there exists Γ' such that A is a Γ'-term.

In comparison to ordinary PTSs, as presented in [2, Section 5.2], we only change the application and conversion rules. The side condition in the application rule is necessary because we modify the notion of β-reduction. We need the side condition to prove standard lemmas about piPTSs, in particular the substitution lemma. However, for a class of logical piPTSs, defined below, this side condition may be omitted: $\Gamma \vdash M : A$ iff $\Gamma \vdash^- M : A$ where $\vdash^-$ is the derivation system with the side condition in the application rule omitted (see Lemma 34). The conversion rule is changed to incorporate proof-irrelevance into the system – this is the major difference with ordinary PTSs. In contrast to [30] we do not a priori require $x \in V^{s_1}$ in the product rule.

▶ **Definition 13.** Let $\lambda S = (\mathcal{S}, \mathcal{A}, \mathcal{R})$ be a piPTS.
λS is *functional* if

1. $(s, s_1), (s, s_2) \in \mathcal{A}$ implies $s_1 = s_2$,
2. $(s, s', s_1), (s, s', s_2) \in \mathcal{R}$ implies $s_1 = s_2$.

λS is *logical* if

1. it is functional,
2. $(*^p, *^p, *^p) \in \mathcal{R}$,
3. all rules in $\mathcal{R}$ involving $*^p$ have the form $(s, *^p, *^p)$ or $(*^p, s, s)$,
4. there is no $s \in \mathcal{S}$ with $(s, *^p) \in \mathcal{A}$,
5. there exists $s \in \mathcal{S}$ with $(*^p, s) \in \mathcal{A}$.

Functional PTSs are called *singly-sorted* in [2]. The notion of functional PTSs comes from [20], and also appears in [18]. A notion of logical PTSs similar to ours occurs in [12, 6], but it differs in some technical details. The restrictions in the definition of a logical piPTS ensure that the sort of propositions $*^p$ has the expected properties, which turn out to be needed in the soundness proof.

▶ **Example 14.** A paradigmatic example of a logical piPTS is the calculus of constructions CC^s with a separate impredicative set universe $*^s$.

- $\mathcal{S} = \{*^p, *^s, \Box\}$.
- $\mathcal{A} = \{(*^p, \Box), (*^s, \Box)\}$.
- $\mathcal{R} = \{(*^p, *^p), (*^s, *^p), (*^p, *^s), (*^s, *^s), (\Box, *^p), (*^p, \Box), (\Box, *^s), (*^s, \Box), (\Box, \Box)\}$.

All (piPTS analogons of) systems of the lambda-cube [2, Definition 5.1.10] are also logical piPTSs if we take $*^p = *$. But since they do not have a distinct sort $*^s$ for a set universe, translating them using our embedding does not make much sense – terms intuitively denoting set elements would be erased instead of translated to first-order terms.

▶ **Example 15.** Let $(\mathcal{S}, \mathcal{A}, \mathcal{R})$ be any logical piPTS. Let $\Box$ be such that $(*^p, \Box) \in \mathcal{A}$ and let $\alpha \in V^{\Box}$ and $x \in V^{*^p}$. We have $\alpha : *^p \vdash (\alpha \to \alpha) : *^p$ and $\alpha : *^p \vdash ((\alpha \to \alpha) \to \alpha \to \alpha) : *^p$, because $(*^p, *^p, *^p) \in \mathcal{R}$. Hence by the abstraction rule $\alpha : *^p \vdash (\lambda x : \alpha.x) : \alpha \to \alpha$ and $\alpha : *^p \vdash (\lambda x : \alpha \to \alpha.x) : (\alpha \to \alpha) \to \alpha \to \alpha$. Also $(\lambda x : \alpha.x) \sim x$, because $x \in V^{*^p}$ and $\lambda x : \alpha.x \to_\varepsilon \lambda x : \alpha.\varepsilon \to_\varepsilon \varepsilon$. So $\alpha : *^p \vdash ((\lambda x : \alpha \to \alpha.x)(\lambda x : \alpha.x)) : \alpha \to \alpha$ by the application rule.

The meta-theory of piPTSs is similar to that of ordinary PTSs (see [2, Section 5.2]). The proofs follow the same pattern, except that there is one difficulty caused by the mismatch between $\beta\varepsilon$-reduction in the conversion rule and β-reduction for which the subject reduction theorem holds. Below we only state a few results concerning piPTSs. We delegate the proofs and other details to Appendix A.

The relation $\to_\varepsilon$ (Definition 8) is confluent and strongly normalising. By $\mathrm{nf}_\varepsilon(M)$ we denote the normal form of M w.r.t. $\to_\varepsilon$. Note that $\mathrm{FV}(\mathrm{nf}_\varepsilon(M)) \subseteq \mathrm{FV}(M)$.

▶ **Lemma 16.** *If $N \sim x$ then $\mathrm{nf}_\varepsilon(M[N/x]) = \mathrm{nf}_\varepsilon(M)[\mathrm{nf}_\varepsilon(N)/x]$.*

▶ **Lemma 17** (Confluence of $\beta\varepsilon$-reduction)**.** *If $M \to^*_{\beta\varepsilon} M_1$ and $M \to^*_{\beta\varepsilon} M_2$ then there exists M' such that $M_1 \to^*_{\beta\varepsilon} M'$ and $M_2 \to^*_{\beta\varepsilon} M'$.*

▶ **Lemma 18.** *If $M =_{\beta\varepsilon} N$ then $M \to^*_\varepsilon \varepsilon$ is equivalent to $N \to^*_\varepsilon \varepsilon$.*

▶ **Lemma 19.** *If N does not contain ε and $M \to^*_{\beta\varepsilon} N$ then $M \to^*_\beta N$.*

▶ **Lemma 20** (Free variable lemma)**.** *If $\Gamma = x_1 : A_1, \ldots, x_n : A_n$ and $\Gamma \vdash B : C$ then:*

1. *the $x_1, \ldots, x_n$ are all distinct,*
2. *$\mathrm{FV}(B), \mathrm{FV}(C) \subseteq \{x_1, \ldots, x_n\}$,*
3. *$\mathrm{FV}(A_i) \subseteq \{x_1, \ldots, x_{i-1}\}$ for $i = 1, \ldots, n$.*

▶ **Lemma 21** (Start lemma)**.** *Let Γ be a legal context.*

1. *If $(s_1, s_2) \in \mathcal{A}$ then $\Gamma \vdash s_1 : s_2$.*
2. *If $(x : A) \in \Gamma$ then $\Gamma \vdash x : A$ and there is $s \in \mathcal{S}$ with $\Gamma_1 \vdash A : s$ and $x \in V^s$, where $\Gamma = \Gamma_1, x : A, \Gamma_2$.*

▶ **Lemma 22** (Substitution lemma)**.** *If $\Gamma, x : A, \Gamma' \vdash B : C$ and $\Gamma \vdash D : A$ and $D \sim x$ then $\Gamma, \Gamma'[D/x] \vdash B[D/x] : C[D/x]$.*

▶ **Lemma 23** (Thinning lemma). *If $\Gamma \vdash A : B$ and $\Gamma' \supseteq \Gamma$ is a legal context then $\Gamma' \vdash A : B$.*

▶ **Lemma 24** (Generation lemma).

1. *If $\Gamma \vdash s : A$ then there is $s' \in \mathcal{S}$ with $A =_{\beta\varepsilon} s'$ and $(s, s') \in \mathcal{A}$.*
2. *If $\Gamma \vdash x : A$ then there are $s \in \mathcal{S}$ and B such that $A =_{\beta\varepsilon} B$ and $\Gamma \vdash B : s$ and $(x : B) \in \Gamma$ and $x \in V^s$.*
3. *If $\Gamma \vdash (\Pi x : A.B) : C$ then there is $(s_1, s_2, s_3) \in \mathcal{R}$ with $\Gamma \vdash A : s_1$ and $\Gamma, x : A \vdash B : s_2$ and $C =_{\beta\varepsilon} s_3$.*
4. *If $\Gamma \vdash (\lambda x : A.M) : C$ then there are $s \in \mathcal{S}$ and B such that $\Gamma \vdash (\Pi x : A.B) : s$ and $\Gamma, x : A \vdash M : B$ and $C =_{\beta\varepsilon} \Pi x : A.B$.*
5. *If $\Gamma \vdash MN : C$ then there are A, B such that $\Gamma \vdash M : (\Pi x : A.B)$ and $\Gamma \vdash N : A$ and $C =_{\beta\varepsilon} B[N/x]$ and $N \sim x$.*

▶ **Corollary 25.** *In a logical piPTS, if $\Gamma \vdash (\Pi x : A.B) : *^p$ then $\Gamma, x : A \vdash B : *^p$.*

Proof. By the generation lemma there are $s, s' \in \mathcal{S}$ such that $(s, s', *^p) \in \mathcal{R}$ and $\Gamma, x : A \vdash B : s'$. Because the piPTS is logical, we have $s' = *^p$. ◀

▶ **Lemma 26** (Correctness of types lemma). *If $\Gamma \vdash M : A$ then there is $s \in \mathcal{S}$ such that $A = s$ or $\Gamma \vdash A : s$.*

▶ **Lemma 27** (Uniqueness of types lemma).

1. *In a functional piPTS, if $\Gamma \vdash A : B$ and $\Gamma \vdash A : B'$ then $B =_{\beta\varepsilon} B'$.*
2. *In a logical piPTS, if $\Gamma \vdash M_1 : A_1$ and $\Gamma \vdash M_2 : A_2$ and $M_1 =_{\beta\varepsilon} M_2$ and $M_1 \not\to^*_\varepsilon \varepsilon$ and $M_2 \not\to^*_\varepsilon \varepsilon$ then $A_1 =_{\beta\varepsilon} A_2$.*

▶ **Corollary 28.** *In a functional piPTS, if $\Pi x : A.B$ is a Γ-term and $\Gamma \vdash A : s$ then $x \in V^s$.*

Proof. By the correctness of types and the generation lemmas $\Gamma, x : A$ is a legal context. By the start lemma $\Gamma \vdash A : s'$ and $x \in V^{s'}$ for some $s' \in \mathcal{S}$. But $s' = s$ by the uniqueness of types lemma. ◀

▶ **Theorem 29** (Subject reduction theorem). *If $\Gamma \vdash A : B$ and $A \to^*_\beta A'$ then $\Gamma \vdash A' : B$.*

Subject reduction obviously does not hold for $\beta\varepsilon$-reduction, because ε is not meant to be typable. This generates a small difficulty in proving the following theorem. See Appendix A.

▶ **Theorem 30.** *Assume the piPTS is logical and M is a Γ-term. Then M is a Γ-proof if and only if $M \to^*_\varepsilon \varepsilon$.*

▶ **Lemma 31.** *In a logical piPTS, if M is a Γ-term and $M =_{\beta\varepsilon} N$ and $\Gamma \vdash N : s$ then $\Gamma \vdash M : s$.*

▶ **Lemma 32.** *In a logical piPTS, if $\Gamma \vdash M : A$ and $\Gamma, x : A$ is a legal context then $M \sim x$.*

▶ **Definition 33.** Given a piPTS specification $(\mathcal{S}, \mathcal{A}, \mathcal{R})$, we write $\Gamma \vdash^- M : N$ if the judgement $\Gamma \vdash M : N$ is derivable in the piPTS determined by the specification (i.e. using the rules in Figure 2), but with the side condition $N \sim x$ omitted in the application rule.

▶ **Lemma 34.** *In a logical piPTS, $\Gamma \vdash^- M : N$ is equivalent to $\Gamma \vdash M : N$.*

▶ Remark. We have not investigated the normalisation or decidability properties of piPTSs. We expect that the (strong) normalisation of an ordinary PTS carries over to its proof-irrelevant version. The same is expected about the decidability of type checking and type inference. The normalisation of our proof-irrelevant version of the Calculus of Constructions (CC^s from Example 14) may probably be shown by adapting a proof-irrelevant model of the ordinary Calculus of Constructions. We do not attempt to answer these questions in the present paper.

4 The embedding

In this and the following section we assume a fixed logical piPTS $\lambda S = (\mathcal{S}, \mathcal{A}, \mathcal{R})$.

By $\mathbb{T}_{\mathrm{FOL}}$ we denote the set of first-order terms, by $\mathcal{F}_{\mathrm{FOL}}$ the set of first-order formulas, by V_{FOL} the set of first-order variables, and by Σ_{FOL} the first-order signature. We assume that each piPTS variable is also a first-order variable and ε and all piPTS sorts are also first-order constants. Further, we assume five functions $\Lambda_0 : V_{\mathrm{FOL}} \times \mathcal{F}_{\mathrm{FOL}} \times \mathbb{T}_{\mathrm{FOL}} \to \Sigma_{\mathrm{FOL}}$ and $\Lambda_1 : V_{\mathrm{FOL}} \times \mathbb{T}_{\mathrm{FOL}} \times \mathbb{T}_{\mathrm{FOL}} \to \Sigma_{\mathrm{FOL}}$ and $\Phi : \mathcal{F}_{\mathrm{FOL}} \to \Sigma_{\mathrm{FOL}}$ and $\mathcal{G}_0 : V_{\mathrm{FOL}} \times \mathcal{F}_{\mathrm{FOL}} \times \mathbb{T}_{\mathrm{FOL}} \times \mathcal{S} \to \Sigma_{\mathrm{FOL}}$ and $\mathcal{G}_1 : V_{\mathrm{FOL}} \times \mathbb{T}_{\mathrm{FOL}} \times \mathbb{T}_{\mathrm{FOL}} \times \mathcal{S} \to \Sigma_{\mathrm{FOL}}$ returning unique fresh first-order constants. The functions are assumed to yield equal results for terms which differ only in the names of variables, e.g., if σ is a renaming then $\Lambda_1(x, r, t) = \Lambda_1(\sigma(x), \sigma(r), \sigma(t))$. The functions are assumed to give different results for terms which differ not only in the names of variables.

The intention here is that $\Lambda_0, \Lambda_1, \Phi, \mathcal{G}_0, \mathcal{G}_1$ return "fresh" first-order symbol names to be used for translations of "lifted-out" lambda-expressions, propositions or dependent products. The functions should return equal results for translations of lambda-expressions differing only in the names of bound variables (the translations then differ only in the names of first-order variables). Note that such functions always exist – e.g. for $\Lambda_1(x, r, t)$ one may simply choose a new symbol name for each new triple (x, r, t) with variable names standardised, e.g, by renaming to x_i the i-th distinct variable in the triple, counting from the left.

We assume that the first-order signature contains a unary relation symbol P, two binary relation symbols T and E, and a binary function symbol @. An atom $P(t)$ is to be intuitively interpreted as "t is provable", and $T(u, t)$ is to be interpreted as "u has type t". The symbol E represents equality. We prefer to work in minimal first-order logic without equality and add necessary equality axioms in the translation. Using first-order logic with equality would complicate the proof notation system λP_1 and the definition of η-long normal forms. The symbol @ represents application. We usually write tu instead of $@(t, u)$, and we assume application to be left-associative.

We often abbreviate e.g. $MN_1 \dots N_n$ by $M\vec{N}$, and $M[N_1/x_1] \dots [N_n/x_n]$ by $M[\vec{N}/\vec{x}]$, and $\Pi x_1 : A_1 \dots \Pi x_n : A_n.B$ by $\Pi \vec{x} : \vec{A}.B$, and $\lambda x_1 : A_1 \dots \lambda x_n : A_n.M$ by $\lambda \vec{x} : \vec{A}.M$. We sometimes treat a list of variables $\vec{x}$ as a set. When we write e.g. $\vec{x} = \mathrm{FV}(\varphi, t)$ then $\vec{x}$ is the list of free variables occurring in φ, t in the fixed order from left to right.

The embedding translates a context Γ and a Γ-proposition τ into a set of axioms Δ_Γ and a first-order formula $\mathcal{F}_\Gamma(\tau)$. The embedding uses two functions:

1. $\mathcal{F}_\Gamma$ which translates Γ-propositions to first-order formulas,
2. $\mathcal{C}_\Gamma$ which translates Γ-terms to first-order individual terms.

▶ **Definition 35.** The functions $\mathcal{F}_\Gamma$ and $\mathcal{C}_\Gamma$ are defined by mutual induction on the structure of piPTS terms.

The definition of $\mathcal{F}_\Gamma$ is as follows.

- if $\Gamma \vdash A : *^p$ then $\mathcal{F}_\Gamma(\Pi x : A.B) = \mathcal{F}_\Gamma(A) \to \mathcal{F}_{\Gamma, x:A}(B)$,
- if $\Gamma \nvdash A : *^p$ then $\mathcal{F}_\Gamma(\Pi x : A.B) = \forall x.T(x, \mathcal{C}_\Gamma(A)) \to \mathcal{F}_{\Gamma, x:A}(B)$,
- if M is not a product then $\mathcal{F}_\Gamma(M) = P(\mathcal{C}_\Gamma(M))$.

The definition of $\mathcal{C}_\Gamma$ is as follows. If M is a Γ-proof then $\mathcal{C}_\Gamma(M) = \varepsilon$. Otherwise, we are in one of the following cases.

- $M = s \in \mathcal{S}$. Then $\mathcal{C}_\Gamma(s) = s$.
- $M = x$ is a variable. Then $\mathcal{C}_\Gamma(x) = x$.
- $M = NQ$. Then $\mathcal{C}_\Gamma(NQ) = \mathcal{C}_\Gamma(N)\mathcal{C}_\Gamma(Q)$.

- $M = \lambda x : A.N$ with $\Gamma \vdash A : *^p$. Let $\varphi = \mathcal{F}_\Gamma(A)$ and $t = \mathcal{C}_{\Gamma,x:A}(N)$ and $\vec{y} = \mathrm{FV}(\varphi, t) \setminus \{x\}$ and $f = \Lambda_0(x, \varphi, t)$. Then $\mathcal{C}_\Gamma(\lambda x : A.N) = f\vec{y}$. The idea here is to "lift-out" the translation of a complex lambda-expression M by introducing a name f for it. In Δ_Γ there will be an axiom describing the functional behaviour of f.
- $M = \lambda x : A.N$ with $\Gamma \nvdash A : *^p$. Let $r = \mathcal{C}_\Gamma(A)$ and $t = \mathcal{C}_{\Gamma,x:A}(N)$ and $\vec{y} = \mathrm{FV}(r, t) \setminus \{x\}$ and $f = \Lambda_1(x, r, t)$. Then $\mathcal{C}_\Gamma(\lambda x : A.N) = f\vec{y}$.
- $M = \Pi x : A.B$ and $\Gamma \vdash M : *^p$. Let $\varphi = \mathcal{F}_\Gamma(\Pi x : A.B)$ and $\vec{y} = \mathrm{FV}(\varphi)$ and $f = \Phi(\varphi)$. Then $\mathcal{C}_\Gamma(\Pi x : A.B) = f\vec{y}$.
- $M = \Pi x : A.B$ and $\Gamma \vdash M : s$ with $s \neq *^p$, and $\Gamma \vdash A : *^p$. Let $\varphi = \mathcal{F}_\Gamma(A)$ and $t = \mathcal{C}_{\Gamma,x:A}(B)$ and $\vec{y} = \mathrm{FV}(\varphi, t) \setminus \{x\}$ and $f = \mathcal{G}_0(x, \varphi, t, s)$. Then $\mathcal{C}_\Gamma(\Pi x : A.B) = f\vec{y}$.
- $M = \Pi x : A.B$ and $\Gamma \vdash M : s$ with $s \neq *^p$, and $\Gamma \nvdash A : *^p$. Let $t_1 = \mathcal{C}_\Gamma(A)$ and $t_2 = \mathcal{C}_{\Gamma,x:A}(B)$ and $\vec{y} = \mathrm{FV}(t_1, t_2) \setminus \{x\}$ and $f = \mathcal{G}_1(x, t_1, t_2, s)$. Then $\mathcal{C}_\Gamma(\Pi x : A.B) = f\vec{y}$.

Note that it follows from the uniqueness of types lemma that all cases in the definition of $\mathcal{F}_\Gamma$ (resp. $\mathcal{C}_\Gamma$) are exclusive.

▶ **Example 36.** Suppose the piPTS is CC^s from Example 14. Let $\Gamma = \alpha : *^s, p : \alpha \to *^p$ and $\tau = \Pi x : \alpha.px \to px$. Then $\Gamma \vdash \tau : *^p$ and $\mathcal{F}_\Gamma(\tau) = \forall x.T(x, \alpha) \to P(px) \to P(px)$. In practice, the atom $P(px)$ may often be further optimised to $P_p(x)$ with P_p a first-order predicate corresponding to p. This optimisation is performed in [15]. For $Q = \lambda x.\Lambda X : T(x, \alpha).\Lambda Y : P(px).Y$ in η-lnf we have $\vdash_{\mathrm{FOL}} Q : \mathcal{F}_\Gamma(\tau)$. The first-order proof Q may be translated back into a piPTS proof term $M = \lambda x : \alpha.\lambda y : px.y$. In CC^s we have $\Gamma \vdash M : \tau$.

Now let $\Gamma' = \alpha : *^s, p : \alpha \to *^p, a : \alpha, q : pa \to *^p$ and $\tau' = \Pi x : pa.qx \to qx$. Then $\Gamma' \vdash \tau' : *^p$ and $\mathcal{F}_{\Gamma'}(\tau') = P(pa) \to P(q\varepsilon) \to P(q\varepsilon)$. For $Q = \Lambda X : P(pa).\Lambda Y : P(q\varepsilon).Y$ in η-lnf we have $\vdash_{\mathrm{FOL}} Q : \mathcal{F}_{\Gamma'}(\tau')$. The proof Q may be translated back to a piPTS proof term $M = \lambda x : pa.\lambda y : qx.y$. In CC^s we have $\Gamma' \vdash M : \tau'$.

▶ **Definition 37.** The translation $\lceil \Gamma \rceil$ of a context Γ is defined inductively:
- $\lceil \langle\rangle \rceil = \emptyset$,
- $\lceil \Gamma, x : A \rceil = \lceil \Gamma \rceil, \mathcal{F}_\Gamma(A)$ if $\Gamma \vdash A : *^p$,
- $\lceil \Gamma, x : A \rceil = \lceil \Gamma \rceil, T(x, \mathcal{C}_\Gamma(A))$ if $\Gamma \nvdash A : *^p$.

The set Δ_Γ will consist of $\lceil \Gamma \rceil$ and a set of axioms Δ_{Ax}. To precisely formulate the axioms we need a technical definition of a function $\mathbb{A}_\Gamma$ such that for a FOL formula φ the formula $\mathbb{A}_\Gamma(\varphi)$ is φ prepended with the declarations in Γ translated into guards.

▶ **Definition 38.** The function $\mathbb{A}$ takes a legal context and a FOL formula and returns a FOL formula. It is defined by induction on the length of the context Γ:
- $\mathbb{A}_{\langle\rangle}(\varphi) = \varphi$,
- $\mathbb{A}_{\Gamma,x:A}(\varphi) = \mathbb{A}_\Gamma(\forall x.T(x, \mathcal{C}_\Gamma(A)) \to \varphi)$ if $\Gamma \vdash A : s$ and $s \neq *^p$,
- $\mathbb{A}_{\Gamma,x:A}(\varphi) = \mathbb{A}_\Gamma(\mathcal{F}_\Gamma(A) \to \varphi)$ if $\Gamma \vdash A : *^p$.

The $\mathbb{A}$-*length* of a legal context Γ, denoted $\mathrm{len}_{\mathbb{A}}(\Gamma)$, is defined inductively:
- $\mathrm{len}_{\mathbb{A}}(\langle\rangle) = 0$,
- $\mathrm{len}_{\mathbb{A}}(\Gamma, x : A) = \mathrm{len}_{\mathbb{A}}(\Gamma) + 2$ if $\Gamma \vdash A : s$ and $s \neq *^p$,
- $\mathrm{len}_{\mathbb{A}}(\Gamma, x : A) = \mathrm{len}_{\mathbb{A}}(\Gamma) + 1$ if $\Gamma \vdash A : *^p$.

The $\mathbb{A}$-length of Γ indicates how many arguments need to be applied to a first-order proof of $\mathbb{A}_\Gamma(\varphi)$ in order to obtain a proof of φ. It follows from the uniqueness of types lemma that $\mathbb{A}_\Gamma$ and $\mathrm{len}_{\mathbb{A}}(\Gamma)$ are well-defined for a legal context Γ.

▶ **Example 39.** In CC^s let $\Gamma = \alpha : *^s, a : \alpha, p : \alpha \to *^p, q : pa$ and $\varphi = P(pa)$. Then

$$\mathbb{A}_\Gamma(\varphi) = \forall\alpha.T(\alpha, *^s) \to \forall a.T(a, \alpha) \to \forall p.T(p, f\alpha) \to P(pa) \to P(pa)$$

where $f = \mathcal{G}_1(x, \alpha, *^p, \Box)$. The $\mathbb{A}$-length of Γ is 7, i.e., $\mathrm{len}_{\mathbb{A}}(\Gamma) = 7$.

We need to define a set of axioms Δ_{Ax} for our embedding. These will be axioms concerning the constants introduced in the translation via the functions Λ_0, Λ_1, Φ, $\mathcal{G}_0$ and $\mathcal{G}_1$, and axioms for equality.

▶ **Definition 40.** The set Δ_{Λ_0} contains the following FOL formulas which describe the behaviour of the constants representing "lifted-out" lambda-expressions with propositional arguments.

Given a variable x, a FOL formula φ and a FOL term t, let $f = \Lambda_0(x, \varphi, t)$. Let Γ and A, B be such that:

- $\Gamma \vdash A : *^p$, and
- $\lambda x : A.B$ is a Γ-term but not a Γ-proof, and
- $\varphi = \mathcal{F}_\Gamma(A)$ and,
- $t = \mathcal{C}_{\Gamma, x:A}(B)$.

Let $\vec{y} = \mathrm{FV}(\varphi, t) \setminus \{x\}$. Then Δ_{Λ_0} contains the FOL formula:

- $\mathbb{A}_\Gamma(\varphi \to E(f\vec{y}\varepsilon, t))$.

▶ **Definition 41.** The set Δ_{Λ_1} contains the following FOL formulas which describe the behaviour of the constants representing "lifted-out" lambda-expressions with non-propositional arguments.

Given a variable x and FOL terms r, t, let $f = \Lambda_1(x, r, t)$. Let Γ and A, B be such that:

- $\Gamma \nvdash A : *^p$, and
- $\lambda x : A.B$ is a Γ-term but not a Γ-proof, and
- $r = \mathcal{C}_\Gamma(A)$, and
- $t = \mathcal{C}_{\Gamma, x:A}(B)$.

Let $\vec{y} = \mathrm{FV}(r, t) \setminus \{x\}$. Then Δ_{Λ_1} contains the FOL formula:

- $\mathbb{A}_\Gamma(\forall x.T(x, r) \to E(f\vec{y}x, t))$.

▶ **Example 42.** In CC^s let $\Gamma = \alpha : *^s$ and $M = \lambda x : \alpha.x$. Then $\Gamma \vdash M : \alpha \to \alpha : *^s$. We have $\mathcal{C}_\Gamma(\alpha) = \alpha$ and $\mathcal{C}_{\Gamma, x:\alpha}(x) = x$. Let $f = \Lambda_1(x, \alpha, x)$. Then Δ_{Λ_1} contains the FOL formula

$$\forall \alpha.T(\alpha, *^s) \to \forall x.T(x, \alpha) \to E(f\alpha x, x).$$

Recall that E represents equality.

▶ **Definition 43.** The set Δ_Φ contains the following FOL formulas which are axioms for the constants representing "lifted-out" propositions. Given a FOL formula φ, let $f = \Phi(\varphi)$ and $\vec{y} = \mathrm{FV}(\varphi)$. Then Δ_Φ contains $\forall \vec{y}.\varphi \to P(f\vec{y})$.

▶ **Definition 44.** The set $\Delta_{\mathcal{G}_0}$ contains the following FOL formulas which describe the behaviour of the constants representing "lifted-out" dependent product types with propositional source types.

Given a variable x, a FOL formula φ, a FOL term t and a sort $s \neq *^p$, let $f = \mathcal{G}_0(x, \varphi, t, s)$. Let Γ and A, B be such that:

- $\Gamma \vdash A : *^p$, and
- $\Gamma \vdash \Pi x : A.B : s$, and
- $\varphi = \mathcal{F}_\Gamma(A)$, and
- $t = \mathcal{C}_{\Gamma, x:A}(B)$.

Let $\vec{y} = \mathrm{FV}(\varphi, t) \setminus \{x\}$ and $z \notin \mathrm{FV}(\varphi, t)$. Then $\Delta_{\mathcal{G}_0}$ contains:

- $\mathbb{A}_\Gamma(\forall z.T(z, f\vec{y}) \to \varphi \to T(z\varepsilon, t))$.

▶ **Definition 45.** The set $\Delta_{\mathcal{G}_1}$ contains the following FOL formulas which describe the behaviour of the constants representing "lifted-out" dependent product types with non-propositional source types.

Given a variable x, FOL terms t_1, t_2, and a sort $s \neq *^p$, let $f = \mathcal{G}_1(x, t_1, t_2, s)$. Let Γ and A, B be such that:

- $\Gamma \nvdash A : *^p$, and
- $\Gamma \vdash \Pi x : A.B : s$, and
- $t_1 = \mathcal{C}_\Gamma(A)$, and
- $t_2 = \mathcal{C}_{\Gamma, x:A}(B)$.

Let $\vec{y} = \mathrm{FV}(t_1, t_2) \setminus \{x\}$ and $z \notin \mathrm{FV}(t_1, t_2)$. Then $\Delta_{\mathcal{G}_1}$ contains:

- $\mathbb{A}_\Gamma(\forall z.T(z, f\vec{y}) \to \forall x.T(x, t_1) \to T(zx, t_2))$.

▶ **Example 46.** In CC^s let $\Gamma = \alpha : *^s, p : \alpha \to *^s$. We have $\mathcal{C}_\Gamma(\alpha) = \alpha$ and $\mathcal{C}_{\Gamma, x:\alpha}(px) = px$ and $\Gamma \vdash \Pi x : \alpha.px : *^s$. Let $f = \mathcal{G}_1(x, \alpha, px, *^s)$. Then $\Delta_{\mathcal{G}_1}$ contains

$$\forall \alpha.T(\alpha, *^s) \to \forall p.T(p, g\alpha) \to \forall z.T(z, f\alpha) \to \forall x.T(x, \alpha) \to T(zx, px)$$

where $g = \mathcal{G}_1(x, \alpha, *^s, \Box)$ and $\Delta_{\mathcal{G}_1}$ also contains

$$\forall \alpha.T(\alpha, *^s) \to \forall p.T(p, g\alpha) \to \forall z.T(z, g\alpha) \to \forall x.T(x, \alpha) \to T(zx, *^s).$$

Also $\lceil \Gamma \rceil = T(\alpha, *^p), T(p, g\alpha)$. In [15] there is a distinction between a local context which contains variables bound locally by a λ or a Π, and a global environment which contains the preselected declarations accessible in the proof assistant kernel (Γ_0 from Section 1.1). The guards are not generated for the declarations in the global environment. Assuming Γ here corresponds to the global environment (it is the context translated together with the conjecture), in [15] the last axiom above would be optimised to

$$\forall z.T(z, g\alpha) \to \forall x.T(x, \alpha) \to T(zx, *^s).$$

▶ **Definition 47.** The set Δ_{τ_0} contains the following FOL formulas which describe the types of the constants representing "lifted-out" lambda-expressions with propositional arguments.

Given a variable x, a FOL formula φ, FOL terms t, u and a sort $s \neq *^p$, let $f = \Lambda_0(x, \varphi, t)$ and $g = \mathcal{G}_0(x, \varphi, u, s)$. Let Γ and A, B, M be such that:

- $\Gamma \vdash A : *^p$, and
- $\Gamma \vdash (\lambda x : A.M) : \Pi x : A.B : s$, and
- $\varphi = \mathcal{F}_\Gamma(A)$, and
- $t = \mathcal{C}_{\Gamma, x:A}(M)$, and
- $u = \mathcal{C}_{\Gamma, x:A}(B)$.

Let $\vec{y} = \mathrm{FV}(\varphi, t) \setminus \{x\}$ and $\vec{z} = \mathrm{FV}(\varphi, u) \setminus \{x\}$. Then Δ_{τ_0} contains the FOL formula:

- $\mathbb{A}_\Gamma(T(f\vec{y}, g\vec{z}))$.

▶ **Definition 48.** The set Δ_{τ_1} contains the following FOL formulas which describe the types of the constants representing "lifted-out" lambda-expressions with non-propositional arguments.

Given a variable x, FOL terms r, t, u and a sort $s \neq *^p$, let $f = \Lambda_1(x, r, t)$ and $g = \mathcal{G}_1(x, r, u, s)$. Let Γ and A, B, M be such that:

- $\Gamma \nvdash A : *^p$, and
- $\Gamma \vdash (\lambda x : A.M) : \Pi x : A.B : s$, and
- $r = \mathcal{C}_\Gamma(A)$, and
- $t = \mathcal{C}_{\Gamma, x:A}(M)$, and

- $u = \mathcal{C}_{\Gamma,x:A}(B)$.

Let $\vec{y} = \mathrm{FV}(r,t) \setminus \{x\}$ and $\vec{z} = \mathrm{FV}(r,u) \setminus \{x\}$. Then Δ_{τ_1} contains the FOL formula:

- $\mathbb{A}_\Gamma(T(f\vec{y}, g\vec{z}))$.

▶ **Definition 49.** The set Δ_E, which axiomatises the equality predicate E, contains the following FOL formulas:

- (reflexivity) $\forall x.E(x,x)$,
- (symmetry) $\forall xy.E(x,y) \to E(y,x)$,
- (transitivity) $\forall xyz.E(x,y) \to E(y,z) \to E(x,z)$,
- (congruence) $\forall xyx'y'.E(x,x') \to E(y,y') \to E(xy,x'y')$,
- (substitutivity for P) $\forall xx'.E(x,x') \to P(x) \to P(x')$,
- (substitutivity for T) $\forall xyx'y'.E(x,x') \to E(y,y') \to T(x,y) \to T(x',y')$.

▶ **Definition 50.** We set $\Delta_{\mathrm{Ax}} = \Delta_{\Lambda_0} \cup \Delta_{\Lambda_1} \cup \Delta_\Phi \cup \Delta_{\mathcal{G}_0} \cup \Delta_{\mathcal{G}_1} \cup \Delta_{\tau_0} \cup \Delta_{\tau_1} \cup \Delta_E$ and $\Delta_\Gamma = \Delta_{\mathrm{Ax}} \cup \lceil\Gamma\rceil$.

▶ Remark. Strictly speaking, the set Δ_{Ax} is infinite, because $\Delta_{\Lambda_0}, \Delta_{\Lambda_1}, \Delta_\Phi, \Delta_{\mathcal{G}_0}, \Delta_{\mathcal{G}_1}, \Delta_{\tau_0}, \Delta_{\tau_1}$ are. However, in practice one needs to add the axioms only for the constants f, contexts Γ and terms A, B, M that actually occur during the translation of a given conjecture and its context. There are only finitely many of them. Also, to make proof reconstruction computable we assume that for any constant $f \in \Lambda_1(x,r,t)$ (and analogously for $\Lambda_0, \mathcal{G}_0, \mathcal{G}_1$) it is possible to compute Γ, A, B satisfying the conditions in Definition 41. In practice, Γ, A, B may be associated to f during the translation when an axiom for f is first added.

▶ **Example 51.** In CC^s let $\Gamma = \alpha : *^s, p : (\alpha \to \alpha) \to *^p$ and

$$\tau = p(\lambda x : \alpha.x) \to p((\lambda x : \alpha \to \alpha.x)(\lambda x : \alpha.x)).$$

We have $\lceil\Gamma\rceil = T(\alpha, *^s), T(p, \tau_1\alpha)$ and $\mathcal{F}_\Gamma(\tau) = P(p(f\alpha)) \to P(p(g\alpha(f\alpha)))$ where $f = \Lambda_1(x,\alpha,x)$ and $g = \Lambda_1(x,\tau_2\alpha,x)$ and $\tau_2 = \mathcal{G}_1(x,\alpha,\alpha,*^s)$ and $\tau_1 = \mathcal{G}_1(x,\tau_2\alpha,*^s,\Box)$. The set Δ_{Ax} contains, among others, the following axioms:

- $\forall\alpha.T(\alpha,*^s) \to \forall p.T(p,\tau_1\alpha) \to \forall x.T(x,\tau_2\alpha) \to E(g\alpha x, x)$,
- $\forall\alpha.T(\alpha,*^s) \to T(f\alpha,\tau_2\alpha)$.

In practice, these may be optimised to:

- $\forall x.T(x,\tau_2\alpha) \to E(g\alpha x, x)$,
- $T(f\alpha,\tau_2\alpha)$.

One may derive $\Delta_{\mathrm{Ax}}, \lceil\Gamma\rceil \vdash_{\mathrm{FOL}} E(g\alpha(f\alpha), f\alpha)$. Using the axioms for equality from Δ_E one may thus show $\Delta_{\mathrm{Ax}}, \lceil\Gamma\rceil, P(p(f\alpha)) \vdash_{\mathrm{FOL}} P(p(g\alpha(f\alpha)))$. Hence $\Delta_{\mathrm{Ax}}, \lceil\Gamma\rceil \vdash_{\mathrm{FOL}} \mathcal{F}_\Gamma(\tau)$. Also there is M with $\Gamma \vdash M : \tau$ in CC^s. The use of equality axioms from Δ_E in the derivation of $\Delta_{\mathrm{Ax}}, \lceil\Gamma\rceil \vdash_{\mathrm{FOL}} \mathcal{F}_\Gamma(\tau)$ corresponds to the use of the conversion rule in the derivation of $\Gamma \vdash M : \tau$.

Now consider $\tau' = p(\lambda x : \alpha.x) \to p(\lambda x : \alpha.(\lambda x : \alpha.x)x)$. Then

$$\mathcal{F}_\Gamma(\tau') = P(p(f\alpha)) \to P(p(h\alpha))$$

where $h = \Lambda_1(x,\alpha,f\alpha x)$. In Δ_{Ax} we have the axioms

- $\forall\alpha.T(\alpha,*^s) \to \forall p.T(p,\tau_1\alpha) \to \forall x.T(x,\tau_2\alpha) \to E(h\alpha x, f\alpha x)$,
- $\forall\alpha.T(\alpha,*^s) \to \forall p.T(p,\tau_1\alpha) \to \forall x.T(x,\alpha) \to E(f\alpha x, x)$.

We have $\Delta_{\mathrm{Ax}}, \lceil\Gamma\rceil \nvdash \mathcal{F}_\Gamma(\tau')$, because $\Delta_\Gamma \nvdash_{\mathrm{FOL}} E(h\alpha, f\alpha)$ – only $\Delta_\Gamma, x : \alpha \vdash E(h\alpha x, f\alpha x)$. On the other hand, $\Gamma \vdash (\lambda D : p(\lambda x : \alpha.x).D) : \tau'$ because $\lambda x : \alpha.x =_\beta \lambda x : \alpha.(\lambda x : \alpha.x)x$.

▶ **Example 52.** In CC^s let $\Gamma = p : *^p, q : p \to *^p$ and $\tau = \Pi x : p.\Pi y : p.qx \to qy$. Then $\lceil\Gamma\rceil = T(p, *^p), T(q, \tau_1 p)$ and $\mathcal{F}_\Gamma(\tau) = P(p) \to P(p) \to P(q\varepsilon) \to P(q\varepsilon)$ where $\tau_1 = \mathcal{G}_0(x, p, *^p, \Box)$. The formula $\mathcal{F}_\Gamma(\tau)$ is an intuitionistic tautology. Also $\Gamma \vdash (\lambda x : p.\lambda y : p.\lambda D : qx.D) : \tau$, because $qx =_\varepsilon qy$. This example shows that proof irrelevance is necessary for soundness.

▶ Remark. The incompleteness of the embedding is due to the fact that not enough axioms are present in Δ_{Ax}. After adding axioms expressing the ξ-rule of β-equality, axioms allowing to form new types, axioms corresponding to piPTS axioms, etc., one would probably obtain a complete embedding.

▶ Remark. Assuming the decidability of type checking, the embedding is computable.

Any renaming σ, i.e. a bijection on the set of variables which respects variable sorts, extends in a natural way to a function on first-order terms, formulas (renaming both free and bound variables), piPTS terms, and piPTS contexts.

▶ **Lemma 53.** *Let σ be a renaming.*

1. *If $\Gamma \vdash M : A$ then $\sigma(\Gamma) \vdash \sigma(M) : \sigma(A)$.*
2. $\mathcal{C}_{\sigma(\Gamma)}(\sigma(M)) = \sigma(\mathcal{C}_\Gamma(M))$.
3. $\mathcal{F}_{\sigma(\Gamma)}(\sigma(M)) = \sigma(\mathcal{F}_\Gamma(M))$.
4. $\mathbb{A}_{\sigma(\Gamma)}(\sigma(\varphi)) = \sigma(\mathbb{A}_\Gamma(\varphi))$.

5 Soundness

For the soundness theorem one would want to prove: if $\Delta_{\mathrm{Ax}}, \lceil\Gamma\rceil \vdash \mathcal{F}_\Gamma(A)$ and $\Gamma \vdash A : *^p$ then there is M with $\Gamma \vdash M : A$. However, in the soundness proof we need a weaker notion than the function $\mathcal{F}_\Gamma$. The problem is that $\mathcal{F}_\Gamma$ does not have the necessary substitution properties. For instance if $N = \Pi z : A.B$ with $\Gamma \vdash A : *^p$ and $\Gamma \vdash N : *^p$, then $\mathcal{C}_\Gamma(N) = f$ with $f = \Phi(\mathcal{F}_\Gamma(N))$, and we have

$$\begin{array}{rcl} \mathcal{F}_{\Gamma, x:*^p}(\Pi y : x.x)[\mathcal{C}_\Gamma(N)/x] & = & (P(x) \to P(x))[\mathcal{C}_\Gamma(N)/x] \\ & = & P(f) \to P(f) \end{array}$$

while

$$\begin{array}{rcl} \mathcal{F}_\Gamma((\Pi y : x.x)[N/x]) & = & \mathcal{F}_\Gamma(\Pi y : N.N) \\ & = & \mathcal{F}_\Gamma(N) \to \mathcal{F}_{\Gamma, y:N}(N) \\ & = & (\mathcal{F}_\Gamma(A) \to \mathcal{F}_{\Gamma, z:A}(B)) \to (\mathcal{F}_{\Gamma, y:N}(A) \to \mathcal{F}_{\Gamma, y:N, z:A}(B)). \end{array}$$

In the proof we would need these two expressions to be equal. An analogous problem occurs with $\mathcal{C}_\Gamma$. For example if $\Gamma \vdash A : s$ and $\Gamma \nvdash A \to A : *^p$, then

$$\mathcal{C}_{\Gamma, x:A\to A}(\lambda y : A.xy)[\mathcal{C}_\Gamma(\lambda z : A.z)/x] = (fx)[\mathcal{C}_\Gamma(\lambda z : A.z)/x] = fg$$

where $f = \Lambda_1(y, \mathcal{C}_{\Gamma, x:A\to A}(A), xy)$ and $g = \Lambda_1(z, \mathcal{C}_\Gamma(A), z)$, but

$$\mathcal{C}_{\Gamma, x:A\to A}((\lambda y : A.xy)[(\lambda z : A.z)/x]) = \mathcal{C}_{\Gamma, x:A\to A}((\lambda y : A.(\lambda z : A.z)y)) = h$$

where $h = \Lambda_1(y, \mathcal{C}_{\Gamma, x:A\to A}(A), g'y)$ and $g' = \Lambda_1(z, \mathcal{C}_{\Gamma, x:A\to A, y:A}(A), z)$.

The problem is essentially that lambda-abstractions and dependent products may contain free variables. In our setting it does not seem possible to easily solve this problem by e.g. first translating all lambda-abstractions to supercombinators, i.e., terms of the form $\lambda x_1 : A_1 \ldots \lambda x_n : A_n.t$ with $\mathrm{FV}(t) \subseteq \{x_1, \ldots, x_n\}$, and changing the definition of $\mathcal{C}_\Gamma$ to

translate multiple consecutive lambda-abstractions at once, thus eliminating the need for the free variables $\vec{y}$ in the axioms in Δ_{Λ_i}. First of all, this is because for a translation to supercombinators to preserve typing additional assumptions on the piPTS would be necessary. Secondly, this would not help with the problem with dependent products exemplified above, which essentially stems from the fact that with our embedding a piPTS proposition may be translated using either $\mathcal{F}$ or $\mathcal{C}$ depending on where it occurs in a term.

We therefore use weaker relations $\succ^{\mathcal{F}}_{\Gamma}$ and $\succ^{\mathcal{C}}_{\Gamma}$ instead of the functions $\mathcal{F}_\Gamma$ and $\mathcal{C}_\Gamma$.

▶ **Definition 54.** First, we define a relation $\Gamma' \rightsquigarrow \Gamma$ which expresses the fact that Γ may be obtained from Γ' by repeated substitutions (in the sense of the substitution lemma) and context extensions. More precisely, we define $\rightsquigarrow$ as the transitive-reflexive closure of the relation given by the rule:

$$\Gamma_1, x : A, \Gamma_2 \rightsquigarrow \Gamma \text{ if } \Gamma \supseteq \Gamma_1, \Gamma_2[N/x] \text{ is a legal context and } \Gamma_1 \vdash N : A \text{ and } N \sim x.$$

We write $\Gamma' \rightsquigarrow_{\vec{x},\vec{N}} \Gamma$ to make the terms and the variables substituted for explicit, e.g.,

$$\Gamma_1, x : A, \Gamma_2, y : B, \Gamma_3 \rightsquigarrow_{y,x,N_1,N_2} \Gamma_1, \Gamma_2[N_2/x], \Gamma_3[N_1/y][N_2/x]$$

if $\Gamma_1, x : A, \Gamma_2 \vdash N_1 : B$ and $\Gamma_1 \vdash N_2 : A$ and $N_1 \sim y$ and $N_2 \sim x$. Note that the order of the terms and the variables in the subscript is significant. If additionally $N_i \succ^{\mathcal{C}}_{\Gamma_i} t_i$ (the relation $\succ^{\mathcal{C}}_{\Gamma}$ is defined below) for appropriate Γ_i, then we write $\Gamma' \rightsquigarrow_{\vec{x},\vec{N},\vec{t}} \Gamma$. For instance,

$$\Gamma_1, x : A, \Gamma_2, y : B, \Gamma_3 \rightsquigarrow_{x,y,N_1,N_2,t_1,t_2} \Gamma_1, \Gamma_2[N_1/x], \Gamma_3[N_1/x][N_2/y]$$

if $\Gamma_1 \vdash N_1 : A$ and $\Gamma_1, \Gamma_2[N_1/x] \vdash N_2 : B[N_1/x]$ and $N_1 \sim x$ and $N_2 \sim y$ and $N_1 \succ^{\mathcal{C}}_{\Gamma_1} t_1$ and $N_2 \succ^{\mathcal{C}}_{\Gamma_1,\Gamma_2[N_1/x]} t_2$.

▶ **Definition 55.** The relation $\succ^{\mathcal{F}}_{\Gamma}$ between Γ-propositions and first-order formulas, and the relation $\succ^{\mathcal{C}}_{\Gamma}$ between Γ-subjects and first-order terms, are defined by mutual induction on the structure of piPTS terms.

The definition of $\succ^{\mathcal{F}}_{\Gamma}$ is as follows.

- if $\Gamma \vdash A : *^p$ and $A \succ^{\mathcal{F}}_{\Gamma} \varphi$ and $B \succ^{\mathcal{F}}_{\Gamma,x:A} \psi$ then $\Pi x : A.B \succ^{\mathcal{F}}_{\Gamma} \varphi \to \psi$,
- if $\Gamma \nvdash A : *^p$ and $A \succ^{\mathcal{C}}_{\Gamma} t$ and $B \succ^{\mathcal{F}}_{\Gamma,x:A} \varphi$ then $\Pi x : A.B \succ^{\mathcal{F}}_{\Gamma} \forall x.T(x,t) \to \varphi$,
- if $A \succ^{\mathcal{C}}_{\Gamma} t$ then $A \succ^{\mathcal{F}}_{\Gamma} P(t)$.

The last case is not exclusive with the first two.

The definition of $\succ^{\mathcal{C}}_{\Gamma}$ is as follows. If M is a Γ-proof then $M \succ^{\mathcal{C}}_{\Gamma} \varepsilon$. Otherwise, we are in one of the following cases.

- $M = s \in \mathcal{S}$. Then $s \succ^{\mathcal{C}}_{\Gamma} s$.
- $M = x$ is a variable. Then $x \succ^{\mathcal{C}}_{\Gamma} x$.
- $M = NQ$. If $N \succ^{\mathcal{C}}_{\Gamma} t_1$ and $Q \succ^{\mathcal{C}}_{\Gamma} t_2$ then $NQ \succ^{\mathcal{C}}_{\Gamma} t_1 t_2$.
- $M = (\lambda x : A.Q)[\vec{N}/\vec{x}]$ and there is Γ' such that $\Gamma' \vdash A : *^p$ and $\Gamma' \rightsquigarrow_{\vec{x},\vec{N},\vec{t}} \Gamma$. Assume $A[\vec{N}/\vec{x}] \succ^{\mathcal{F}}_{\Gamma} \varphi[\vec{t}/\vec{x}]$ and $Q[\vec{N}/\vec{x}] \succ^{\mathcal{C}}_{\Gamma,x:A[\vec{N}/\vec{x}]} t[\vec{t}/\vec{x}]$. Let $f = \Lambda_0(x, \varphi, t)$ and $\vec{y} = \mathrm{FV}(\varphi, t) \setminus \{x\}$. Then $M \succ^{\mathcal{C}}_{\Gamma} (f\vec{y})[\vec{t}/\vec{x}]$.
- $M = (\lambda x : A.Q)[\vec{N}/\vec{x}]$ and there is Γ' such that $\Gamma' \nvdash A : *^p$ and $\Gamma' \rightsquigarrow_{\vec{x},\vec{N},\vec{t}} \Gamma$. Assume $A[\vec{N}/\vec{x}] \succ^{\mathcal{C}}_{\Gamma} r[\vec{t}/\vec{x}]$ and $Q[\vec{N}/\vec{x}] \succ^{\mathcal{C}}_{\Gamma,x:A[\vec{N}/\vec{x}]} t[\vec{t}/\vec{x}]$. Let $f = \Lambda_1(x, r, t)$ and $\vec{y} = \mathrm{FV}(r, t) \setminus \{x\}$. Then $M \succ^{\mathcal{C}}_{\Gamma} (f\vec{y})[\vec{t}/\vec{x}]$.
- $M = (\Pi x : A.B)[\vec{N}/\vec{x}]$ and there is Γ' such that $\Gamma' \vdash (\Pi x : A.B) : *^p$ and $\Gamma' \rightsquigarrow_{\vec{x},\vec{N},\vec{t}} \Gamma$. Assume $M \succ^{\mathcal{F}}_{\Gamma} \varphi[\vec{t}/\vec{x}]$. Let $f = \Phi(\varphi)$ and $\vec{y} = \mathrm{FV}(\varphi)$. Then $M \succ^{\mathcal{C}}_{\Gamma} (f\vec{y})[\vec{t}/\vec{x}]$.

- $M = (\Pi x : A.B)[\vec{N}/\vec{x}]$. and there is Γ' such that $\Gamma' \vdash (\Pi x : A.B) : s$ with $s \neq *^p$ and $\Gamma' \vdash A : *^p$ and $\Gamma' \rightsquigarrow_{\vec{x},\vec{N},\vec{t}} \Gamma$. Assume $A[\vec{N}/\vec{x}] \succ^{\mathcal{F}}_{\Gamma} \varphi[\vec{t}/\vec{x}]$ and $B[\vec{N}/\vec{x}] \succ^{\mathcal{C}}_{\Gamma,x:A[\vec{N}/\vec{x}]} t[\vec{t}/\vec{x}]$. Let $f = \mathcal{G}_0(x, \varphi, t, s)$ and $\vec{y} = \mathrm{FV}(\varphi, t) \setminus \{x\}$. Then $M \succ^{\mathcal{C}}_{\Gamma} (f\vec{y})[\vec{t}/\vec{x}]$,
- $M = (\Pi x : A.B)[\vec{N}/\vec{x}]$ and there is Γ' such that $\Gamma' \vdash (\Pi x : A.B) : s$ with $s \neq *^p$ and $\Gamma' \nvdash A : *^p$ and $\Gamma' \rightsquigarrow_{\vec{x},\vec{N},\vec{t}} \Gamma$. Assume $A[\vec{N}/\vec{x}] \succ^{\mathcal{C}}_{\Gamma} u_1[\vec{t}/\vec{x}]$ and $B[\vec{N}/\vec{x}] \succ^{\mathcal{C}}_{\Gamma,x:A[\vec{N}/\vec{x}]} u_2[\vec{t}/\vec{x}]$. Let $f = \mathcal{G}_1(x, u_1, u_2, s)$ and $\vec{y} = \mathrm{FV}(u_1, u_2) \setminus \{x\}$. Then $M \succ^{\mathcal{C}}_{\Gamma} (f\vec{y})[\vec{t}/\vec{x}]$.

Note that not all cases are mutually exclusive.

▶ **Lemma 56.**
1. *If A is a Γ-proposition then $A \succ^{\mathcal{F}}_{\Gamma} \mathcal{F}_\Gamma(A)$.*
2. *If A is a Γ-subject then $A \succ^{\mathcal{C}}_{\Gamma} \mathcal{C}_\Gamma(A)$.*

Proof. Induction on the definition of $\mathcal{F}_\Gamma(A)$ and $\mathcal{C}_\Gamma(A)$, using the generation lemma and Corollary 25. ◀

▶ **Definition 57.** The relation $\succ$ between contexts and first-order environments is defined inductively:
- $\langle\rangle \succ \emptyset$,
- if $\Gamma \vdash A : *^p$ and $A \succ^{\mathcal{F}}_{\Gamma} \varphi$ and $\Gamma \succ \Delta$ then $\Gamma, x : A \succ \Delta, \varphi$,
- if $\Gamma \nvdash A : *^p$ and $A \succ^{\mathcal{C}}_{\Gamma} t$ and $\Gamma \succ \Delta$ then $\Gamma, x : A \succ \Delta, T(x, t)$.

The relation $\succ$ is a "relaxed" analogon of the function $\lceil - \rceil$ from Definition 37.

▶ **Definition 58.** We define the relation $\psi \succ^{\mathbb{A}}_{\Gamma;\Gamma'} \varphi$ by induction on Γ':
- $\varphi \succ^{\mathbb{A}}_{\Gamma;\langle\rangle} \varphi$,
- $\psi \succ^{\mathbb{A}}_{\Gamma;\Gamma',x:A} \varphi$ if $\Gamma, \Gamma' \vdash A : s$ and $s \neq *^p$ and $A \succ^{\mathcal{C}}_{\Gamma,\Gamma'} t$ and $\forall x.T(x, t) \to \psi \succ^{\mathbb{A}}_{\Gamma;\Gamma'} \varphi$.
- $\psi \succ^{\mathbb{A}}_{\Gamma;\Gamma',x:A} \varphi$ if $\Gamma, \Gamma' \vdash A : *^p$ and $A \succ^{\mathcal{F}}_{\Gamma,\Gamma'} \psi_A$ and $\psi_A \to \psi \succ^{\mathbb{A}}_{\Gamma;\Gamma'} \varphi$.

Intuitively, $\psi \succ^{\mathbb{A}}_{\Gamma;\Gamma'} \varphi$ means that φ is ψ with prepended relaxed translations of the declarations in Γ' into guards, like in $\mathbb{A}_{\Gamma'}(\psi)$ from Definition 38. The context Γ provides additional declarations for the purpose of typing – they are not translated into guards. By a "relaxed" translation of M we mean a first-order term t (resp. formula θ) satisfying $M \succ^{\mathcal{C}}_{\Gamma''} t$ (resp. $M \succ^{\mathcal{F}}_{\Gamma''} \theta$) for appropriate Γ''.

▶ **Lemma 59.** *If Γ is a legal context then $\Gamma \succ \lceil \Gamma \rceil$ and $\varphi \succ^{\mathbb{A}}_{\langle\rangle;\Gamma} \mathbb{A}_\Gamma(\varphi)$.*

Proof. Induction on Γ, using Lemma 56. ◀

▶ **Lemma 60.** *If $\Gamma' \rightsquigarrow_{\vec{x},\vec{N},\vec{t}} \Gamma$ and $\Gamma', y : A$ is a legal context and y is fresh, i.e., it does not occur in Γ', Γ or any intermediate context, then $\Gamma', y : A \rightsquigarrow_{\vec{x},\vec{N},\vec{t}} \Gamma, y : A[\vec{N}/\vec{x}]$.*

Proof. Induction on the definition of $\Gamma' \rightsquigarrow_{\vec{x},\vec{N},\vec{t}} \Gamma$. ◀

▶ **Lemma 61.** *If $\Gamma' \vdash A : B$ and $\Gamma' \rightsquigarrow_{\vec{x},\vec{N}} \Gamma$ then $\Gamma \vdash A[\vec{N}/\vec{x}] : B[\vec{N}/\vec{x}]$.*

Proof. Follows by repeatedly applying the substitution and thinning lemmas. ◀

From now on, whenever we write $M \succ^{\mathcal{F}}_{\Gamma} t$ we implicitly assume that M is a Γ-proposition. Similarly, whenever we write $M \succ^{\mathcal{C}}_{\Gamma} t$ we assume M is a Γ-subject. Note that it follows from the generation lemma and Lemma 61 that if e.g. $\Pi x : A.B \succ^{\mathcal{F}}_{\Gamma} \varphi \to \psi$ and $\Pi x : A.B$ is a Γ-proposition, then A is a Γ-proposition and B is a $(\Gamma, x : A)$-proposition (and analogously for all other cases of Definition 55). So the assumption that the left-hand sides of $\succ^{\mathcal{F}}$ are propositions is preserved for $A \succ^{\mathcal{F}}_{\Gamma} \varphi$ and $B \succ^{\mathcal{F}}_{\Gamma,x:A} \psi$. We will often use this observation implicitly. Because of page limits, proofs of many of the following helper lemmas have been moved to Appendix B.

▶ **Lemma 62.**
1. *If* $M \succ^{\mathcal{F}}_{\Gamma} \varphi$ *and* $\Gamma' \supseteq \Gamma$ *is a legal context then* $M \succ^{\mathcal{F}}_{\Gamma'} \varphi$.
2. *If* $M \succ^{\mathcal{C}}_{\Gamma} t$ *and* $\Gamma' \supseteq \Gamma$ *is a legal context then* $M \succ^{\mathcal{C}}_{\Gamma'} t$.

▶ **Corollary 63.** *If* $\psi \succ^{\mathbb{A}}_{\Gamma;\Gamma_0} \varphi$ *and* $\Gamma' \supseteq \Gamma$ *and* Γ', Γ_0 *is a legal context then* $\psi \succ^{\mathbb{A}}_{\Gamma';\Gamma_0} \varphi$.

▶ **Lemma 64.** *Assume* $N \sim x$. *Then* $M \to^*_\varepsilon \varepsilon$ *iff* $M[N/x] \to^*_\varepsilon \varepsilon$.

▶ **Lemma 65.** *Assume* $\Gamma_1 \vdash N : A$ *and* $N \succ^{\mathcal{C}}_{\Gamma_1} t$ *and* $N \sim y$.
1. *If* $M \succ^{\mathcal{F}}_{\Gamma_1, y:A, \Gamma_2} \varphi$ *then* $M[N/y] \succ^{\mathcal{F}}_{\Gamma_1, \Gamma_2[N/y]} \varphi[t/y]$.
2. *If* $M \succ^{\mathcal{C}}_{\Gamma_1, y:A, \Gamma_2} u$ *then* $M[N/y] \succ^{\mathcal{C}}_{\Gamma_1, \Gamma_2[N/y]} u[t/y]$.

▶ **Corollary 66.**
1. *If* $M \succ^{\mathcal{F}}_{\Gamma'} \varphi$ *and* $\Gamma' \rightsquigarrow_{\vec{x},\vec{N},\vec{t}} \Gamma$ *then* $M[\vec{N}/\vec{x}] \succ^{\mathcal{F}}_{\Gamma} \varphi[\vec{t}/\vec{x}]$.
2. *If* $M \succ^{\mathcal{C}}_{\Gamma'} \varphi$ *and* $\Gamma' \rightsquigarrow_{\vec{x},\vec{N},\vec{t}} \Gamma$ *then* $M[\vec{N}/\vec{x}] \succ^{\mathcal{C}}_{\Gamma} \varphi[\vec{t}/\vec{x}]$.

▶ **Lemma 67.** *Assume* $y \in V^{*^p}$.
1. *If* $M \succ^{\mathcal{F}}_{\Gamma} \varphi$ *then* $y \notin \mathrm{FV}(\varphi)$.
2. *If* $M \succ^{\mathcal{C}}_{\Gamma} t$ *then* $y \notin \mathrm{FV}(t)$.

▶ **Lemma 68.**
1. *If* $M \succ^{\mathcal{F}}_{\Gamma} \varphi$ *then* $\mathrm{FV}(\varphi) = \mathrm{FV}(\mathrm{nf}_\varepsilon(M))$.
2. *If* $M \succ^{\mathcal{C}}_{\Gamma} t$ *then* $\mathrm{FV}(t) = \mathrm{FV}(\mathrm{nf}_\varepsilon(M))$.

▶ **Lemma 69.** *Assume* $\Gamma =_\varepsilon \Gamma'$.
1. *If* $M \succ^{\mathcal{F}}_{\Gamma} \varphi$ *and* $M' \succ^{\mathcal{F}}_{\Gamma'} \varphi$ *then* $M =_\varepsilon M'$.
2. *If* $M \succ^{\mathcal{C}}_{\Gamma} t$ *and* $M' \succ^{\mathcal{C}}_{\Gamma'} t$ *then* $M =_\varepsilon M'$.

▶ **Lemma 70.** *If* $\psi \in \Delta$ *and* $\Gamma \succ \Delta$ *and* ψ *has target* P, *then there are* $\Gamma_1, \Gamma_2, \Delta_1, \Delta_2$ *and* C *such that* $\Gamma = \Gamma_1, x : C, \Gamma_2$ *and* $\Delta = \Delta_1, \psi, \Delta_2$ *and* $\Gamma_1 \succ \Delta_1$ *and* $\Gamma_1 \vdash C : *^p$ *and* $C \succ^{\mathcal{F}}_{\Gamma_1} \psi$.

▶ **Lemma 71.** *If* $\psi \in \Delta$ *and* $\Gamma \succ \Delta$ *and* ψ *has target* T, *then* $\psi = T(x,t)$ *and there are* Γ_1, Γ_2 *and* C *such that* $\Gamma = \Gamma_1, x : C, \Gamma_2$ *and* $C \succ^{\mathcal{C}}_{\Gamma_1} t$.

▶ **Lemma 72.** *If* $C \succ^{\mathcal{F}}_{\Gamma} \varphi$ *then* $C = \Pi x_1 : A_1 \ldots \Pi x_n : A_n . B$ *with* $B \succ^{\mathcal{C}}_{\Gamma,\Gamma_0} t$ *and* $P(t) \succ^{\mathbb{A}}_{\Gamma;\Gamma_0} \varphi$ *and* $\Gamma_0 = x_1 : A_1, \ldots, x_n : A_n$.

▶ **Definition 73.** Assume $\Delta_{\mathrm{Ax}}, \Delta \vdash Q : \varphi$ and $\Gamma \succ \Delta$. A $\Gamma, \Delta, A, \varphi$*-reconstruction* of Q, or just a *reconstruction* of Q, is defined as follows, depending on the form of φ.
1. If $A \succ^{\mathcal{F}}_{\Gamma} \varphi$ and $\Gamma \vdash A : *^p$ then any M such that $\Gamma \vdash M : A$ is a $\Gamma, \Delta, A, \varphi$*-reconstruction* of Q.
2. If $\varphi = T(t,t')$ and $A \succ^{\mathcal{C}}_{\Gamma} t'$ and $\Gamma \vdash A : s$ with $s \neq *^p$ then any M such that $M \succ^{\mathcal{C}}_{\Gamma} t$ and $\Gamma \vdash M : A$ is a $\Gamma, \Delta, A, \varphi$*-reconstruction* of Q.
3. If $\varphi = E(t',t)$ and $A \succ^{\mathcal{C}}_{\Gamma} t'$ (resp. $A \succ^{\mathcal{C}}_{\Gamma} t$) and A is a Γ-subject then any Γ-subject M such that $M \succ^{\mathcal{C}}_{\Gamma} t$ (resp. $M \succ^{\mathcal{C}}_{\Gamma} t'$) and $M =_{\beta\varepsilon} A$ is a $\Gamma, \Delta, A, \varphi$*-reconstruction* of Q.

Note that if $A \succ^{\mathcal{F}}_{\Gamma} \varphi$ then φ does not have the form $T(t,t')$ or $E(t,t')$, so the three above cases are actually exclusive. We stress that the notion of a reconstruction depends on the $\Gamma, \Delta, A, \varphi$, but we often omit them when clear.

A first-order proof term Q is *reconstructible* if for any $\Gamma, \Delta, A, \varphi$, satisfying the appropriate conditions as above, a $\Gamma, \Delta, A, \varphi$-reconstruction of Q exists.

▶ **Lemma 74.** *Suppose* $\Gamma \succ \Delta$ *and* $\Delta_{\mathrm{Ax}}, \Delta \vdash XQ_1 \ldots Q_m : \psi$, *where each* Q_i *is either an individual term or a reconstructible proof term. Let* $\Gamma_0 = x_1 : A_1, \ldots, x_n : A_n$ *be such that* $m = \mathrm{len}_{\mathbb{A}}(\Gamma_0)$ *and* Γ, Γ_0 *is a legal context. If* $(X : \gamma) \in \Delta_{\mathrm{Ax}}, \Delta$ *with* $\varphi \succ^{\mathbb{A}}_{\Gamma;\Gamma_0} \gamma$, *then there exist* $N_1, \ldots, N_n$ *and* $u_1, \ldots, u_n$ *such that* $\psi = \varphi[\vec{u}/\vec{x}]$ *and* $\Gamma, \Gamma_0 \rightsquigarrow_{\vec{x},\vec{N},\vec{u}} \Gamma$.

▶ **Theorem 75** (Soundness of the embedding). *Every first-order proof term Q in η-long normal form is reconstructible.*

The proof of the soundness of the embedding is a bit long and tedious because of the many cases that need to be considered. As mentioned before, the soundness proof implicitly defines an algorithm to transform a first-order proof term Q in η-lnf into its reconstruction. More precisely, given a proof term Q in η-lnf and $\Gamma, \Delta, A, \varphi$ satisfying the conditions in Definition 73, the algorithm constructs a $\Gamma, \Delta, A, \varphi$-reconstruction M of Q. We first informally sketch this algorithm. The proof of Theorem 75 is essentially a proof of its correctness. Because any proof term may be β-reduced and η-expanded to a proof term in η-lnf (Lemma 5), this provides a general proof reconstruction method.

▶ **Algorithm 76.** Assume $\Delta_{\mathrm{Ax}}, \Delta \vdash Q : \varphi$ and $\Gamma \succ \Delta$. We assume that $\Gamma \succ \Delta$ is given constructively, i.e., given $(X : \varphi) \in \Delta$ it is possible to retrieve $(x : C) \in \Gamma$ such that $C \succ^{\mathcal{F}}_{\Gamma} \varphi$, or $\varphi = T(x, t)$ and $C \succ^{\mathcal{C}}_{\Gamma} t$ (c.f. Definition 57 and Lemma 62). We have the following cases. For the sake of readability we do not treat all cases in full generality.

1. $A \succ^{\mathcal{F}}_{\Gamma} \varphi$ and $\Gamma \vdash A : *^p$. We seek M with $\Gamma \vdash M : A$. Consider possible forms of φ.
   - $\varphi = \varphi_1 \to \varphi_2$. Then $Q = \lambda X : \varphi_1.Q'$ (because Q is in η-lnf) and $A = \Pi x : B.C$ (by Definition 55) with $B \succ^{\mathcal{F}}_{\Gamma} \varphi_1$ and $C \succ^{\mathcal{F}}_{\Gamma, x:B} \varphi_2$. Recursively construct a $\Gamma', \Delta', C, \varphi_2$-reconstruction M' of Q', where $\Gamma' = \Gamma, x : B$ and $\Delta' = \Delta, \varphi_1$. Take $M = \lambda x : B.M'$.
   - $\varphi = \forall x.T(x, t) \to \psi$. Then $Q = \lambda x \lambda X : T(x, t).Q'$ and $A = \Pi x : B.C$. Recursively construct a $\Gamma', \Delta', C, \psi$-reconstruction M' of Q', where $\Gamma' = \Gamma, x : B$ and $\Delta' = \Delta, T(x, t)$. Take $M = \lambda x : B.M'$.
   - $\varphi = P(t_A)$ with $A \succ^{\mathcal{C}}_{\Gamma} t_A$. Then $Q = XD_1 \dots D_k$ where $(X : \psi) \in \Delta_{\mathrm{Ax}}, \Delta$ and $\mathrm{target}(\psi) = P$, and each D_i is a first-order proof term in η-long normal form or an individual term. We consider possible forms of ψ.
     - $(X : \psi) \in \Delta$. For example, $Q = XtD_1D_2$ where D_1, D_2 are proof terms in η-lnf and $\psi = \forall x.T(x, t_B) \to \psi' \to P(fx)$ and $\Delta_{\mathrm{Ax}}, \Delta \vdash D_1 : T(t, t_B)$ and $\Delta_{\mathrm{Ax}}, \Delta \vdash D_2 : \psi'[t/x]$ and f is a *variable*. There is $(z : C) \in \Gamma$ such that $C \succ^{\mathcal{F}}_{\Gamma} \psi$ and $C = \Pi x : B.\Pi y : B'.fx$. Recursively construct a $\Gamma, \Delta, B, T(x, t_B)$-reconstruction M_1 of D_1 and a $\Gamma, \Delta, B', \psi'[t/x]$-reconstruction M_2 of D_2. Take $M = zM_1M_2$.
     - $(X : \psi) \in \Delta_E$ and $\psi = \forall xx'.E(x, x') \to P(x) \to P(x')$. Then $Q = Xtt_AD_1D_2$ and $\Delta_{\mathrm{Ax}}, \Delta \vdash D_1 : E(t, t_A)$ and $\Delta_{\mathrm{Ax}}, \Delta \vdash D_2 : P(t)$. Recursively construct a $\Gamma, \Delta, A, E(t, t_A)$-reconstruction B of D_1 and then a $\Gamma, \Delta, B, P(t)$-reconstruction M' of D_2. Take $M = M'$.
     - $(X : \psi) \in \Delta_\Phi$ and e.g. $\psi = \forall y.\psi' \to P(fy)$ and $\mathrm{FV}(\psi') = \{y\}$ and $f = \Phi(\psi')$. Then $t_A = ft$ and $Q = XtD$ with $\Delta_{\mathrm{Ax}}, \Delta \vdash D : \psi'[t/y]$. Since $A \succ^{\mathcal{C}}_{\Gamma} ft$, by Definition 55 there are Γ', N with $\Gamma' \rightsquigarrow_{y,N,t} \Gamma$. So $N \succ^{\mathcal{C}}_{\Gamma} t$ by Definition 54 and the thinning lemma. Also $A \succ^{\mathcal{F}}_{\Gamma} \psi'[t/y]$ (Definition 55). Recursively construct a $\Gamma, \Delta, A, \psi'[t/y]$-reconstruction M of D. This is also a $\Gamma, \Delta, A, P(ft)$-reconstruction of Q.
2. $\varphi = T(t, t')$ and $A \succ^{\mathcal{C}}_{\Gamma} t'$ and $\Gamma \vdash A : s$ with $s \neq *^p$. We seek M such that $M \succ^{\mathcal{C}}_{\Gamma} t$ and $\Gamma \vdash M : A$. We have $Q = X\vec{D}$ where $(X : \psi) \in \Delta_{\mathrm{Ax}}, \Delta$ and $\mathrm{target}(\psi) = T$. Consider possible forms of ψ.
   - $(X : \psi) \in \Delta$. Then $\psi = T(x, t')$ and $t = x$ and there is $(x : C) \in \Gamma$ such that $C \succ^{\mathcal{C}}_{\Gamma} t'$. Take $M = x$.
   - $(X : \psi) \in \Delta_{\mathcal{G}_1}$ and e.g. $\psi = \forall z.T(z, f) \to \forall x.T(x, r_1) \to T(zx, r_2)$ where $f = \mathcal{G}_1(x, r_1, r_2, s)$ and $\mathrm{FV}(r_1, r_2) \subseteq \{x\}$. Then $Q = XuD_1wD_2$ and $t = uw$ and $t' = r_2[w/x]$ and $\Delta_{\mathrm{Ax}}, \Delta \vdash D_1 : T(u, f)$ and $\Delta_{\mathrm{Ax}}, \Delta \vdash D_2 : T(w, r_1)$. We may

compute Γ_0, C with $C = \Pi x : C_1.C_2$ and $C_1 \succ^{\mathcal{C}}_{\Gamma_0} r_1$ and $C_2 \succ^{\mathcal{C}}_{\Gamma_0, x:C_1} r_2$ and $C \succ^{\mathcal{C}}_{\Gamma_0} f$ (see Remark 4). One shows that $\Gamma = \Gamma_0$ may be assumed in the case $\mathrm{FV}(r_1, r_2) \subseteq \{x\}$. Recursively construct a $\Gamma, \Delta, C, T(u, f)$-reconstruction M_1 of D_1 and a Γ, Δ, C_1, r_1-reconstruction M_2 of D_2. Take $M = M_1 M_2$.

- $(X : \psi) \in \Delta_{\tau_1}$ and e.g. $\psi = T(f, g)$ where $f = \Lambda_1(x, r, t)$ and $g = \mathcal{G}_1(x, r, u, s)$ with $\mathrm{FV}(r, t, u) \subseteq \{x\}$. We may compute C, N such that $C \succ^{\mathcal{C}}_{\Gamma} r$ and $N \succ^{\mathcal{C}}_{\Gamma, x:C} t$, so $\lambda x : C.N \succ^{\mathcal{C}}_{\Gamma} f$. Take $M = \lambda x : C.N$.
- $(X : \psi) \in \Delta_E$. Then $\psi = \forall xyx'y'.E(x, x') \to E(y, y') \to T(x, y) \to T(x', y')$ and $Q = Xuu'tt'D_1D_2D_3$ where $\Delta_{\mathrm{Ax}}, \Delta \vdash D_1 : E(u, t)$ and $\Delta_{\mathrm{Ax}}, \Delta \vdash D_2 : E(u', t')$ and $\Delta_{\mathrm{Ax}}, \Delta \vdash D_3 : T(u, u')$. Recursively construct a $\Gamma, \Delta, A, E(u', t')$-reconstruction A' of D_2, then a $\Gamma, \Delta, A, T(u, u')$-reconstruction M' of D_3, then a $\Gamma, \Delta, M', E(u, t)$-reconstruction N of D_1. Take $M = N$.

3. $\varphi = E(t_0, t_1)$ and e.g. $A \succ^{\mathcal{C}}_{\Gamma} t_0$. We need to find M with $M \succ^{\mathcal{C}}_{\Gamma} t_1$ and $M =_{\beta\varepsilon} A$. Since $E(t_0, t_1)$ is an atom and Q is in η-lnf, $Q = X\vec{D}$ where $(X : \psi) \in \Delta_{\mathrm{Ax}}, \Delta$ and $\mathrm{target}(\psi) = E$. Consider possible forms of ψ.
   - $(X : \psi) \in \Delta_{\Lambda_1}$ and e.g. $\psi = \forall x.T(x, r_1) \to E(fx, r_2)$ where $f = \Lambda_1(x, r_1, r_2)$ and $\mathrm{FV}(r_1, r_2) \subseteq \{x\}$. Then $Q = XuD$ and $t_0 = fu$ and $t_1 = r_2[u/x]$ and $\Delta_{\mathrm{Ax}}, \Delta \vdash D : T(u, r_1)$. We may compute C, N such that $C_1 \succ^{\mathcal{C}}_{\Gamma} r_1$ and $C_2 \succ^{\mathcal{C}}_{\Gamma, x:C_1} r_2$ and $\lambda x : C.N \succ^{\mathcal{C}}_{\Gamma} f$. Recursively construct a Γ, Δ, C, r_1-reconstruction M_1 of D. Then $(\lambda x : C.N)M_1 \succ^{\mathcal{C}}_{\Gamma} fu$ and $A \succ^{\mathcal{C}}_{\Gamma} fu$, so $A =_{\beta\varepsilon} N[M_1/x]$, using Lemma 69. Also $N[M_1/x] \succ^{\mathcal{C}}_{\Gamma} r_2[u/x]$. Take $M = N[M_1/x]$.
   - $(X : \psi) \in \Delta_E$ and $\psi = \forall xyx'y'.E(x, x') \to E(y, y') \to E(xy, x'y')$. Then $Q = Xuwu'w'D_1D_2$ and $\Delta_{\mathrm{Ax}}, \Delta \vdash D_1 : E(u, u')$ and $\Delta_{\mathrm{Ax}}, \Delta \vdash D_2 : E(w, w')$ and $t_0 = uw$ and $t_1 = u'w'$. Since $A \succ^{\mathcal{C}}_{\Gamma} uw$, we have $A = A_1A_2$ with $A_1 \succ^{\mathcal{C}}_{\Gamma} u$ and $A_2 \succ^{\mathcal{C}}_{\Gamma} w$. Recursively construct a $\Gamma, \Delta, A_1, E(u, u')$-reconstruction B_1 of D_1 and a $\Gamma, \Delta, A_2, E(w, w')$-reconstruction B_2 of D_2. Take $M = B_1B_2$.

Cases omitted in the above sketch are trivial or similar to other cases considered.

Together with Lemma 56, Lemma 59 and Lemma 5, Theorem 75 gives us the following.

▶ **Corollary 77.** *If* $\Delta_{\mathrm{Ax}}, \lceil\Gamma\rceil \vdash \mathcal{F}_{\Gamma}(A)$ *and* $\Gamma \vdash A : *^p$ *then there exists* M *such that* $\Gamma \vdash M : A$.

We now give a rigorous proof of the soundness of the embedding.

Proof of Theorem 75. We show that every first-order proof term Q in η-lnf is reconstructible. We proceed by induction on the size of Q. First of all, note that because of Lemma 68 for any M, Γ and $x \in V^{*^p}$ and φ, t, Δ with $M \succ^{\mathcal{F}}_{\Gamma} \varphi$, $M \succ^{\mathcal{C}}_{\Gamma} t$, $\Gamma \succ \Delta$, we have $x \notin \mathrm{FV}(\varphi, t, \Delta)$. Hence we may assume that if $x \in V^{*^p}$ then x does not occur free in any individual term used in Q.

We need to consider the three cases in Definition 73.

1. Assume $\Delta_{\mathrm{Ax}}, \Delta \vdash Q : \varphi$ and $\Gamma \succ \Delta$ and $\Gamma \vdash A : *^p$ and $A \succ^{\mathcal{F}}_{\Gamma} \varphi$. We need to find M with $\Gamma \vdash M : A$. We consider possible forms of φ.
   - $\varphi = \varphi_1 \to \varphi_2$. Then $A = \Pi x : B.C$ and $\Gamma \vdash B : *^p$ and $B \succ^{\mathcal{F}}_{\Gamma} \varphi_1$ and $C \succ^{\mathcal{F}}_{\Gamma, x:B} \varphi_2$ and $Q = \lambda X : \varphi_1.Q'$. Hence $\Delta_{\mathrm{Ax}}, \Delta, X : \varphi_1 \vdash Q' : \varphi_2$. Note that $\Gamma, x : B \succ \Delta, X : \varphi_1$. Also $\Gamma, x : B \vdash C : *^p$ by Corollary 25. Thus by the inductive hypothesis there is M' with $\Gamma, x : B \vdash M' : C$. Because $\Gamma \vdash (\Pi x : B.C) : *^p$, by the abstraction rule we obtain $\Gamma \vdash (\lambda x : B.M') : (\Pi x : B.C)$. Hence take $M = \lambda x : B.M'$.
   - $\varphi = \forall x.T(x, t) \to \psi$. Then $A = \Pi x : B.C$ and $\Gamma \nvdash B : *^p$ and $B \succ^{\mathcal{C}}_{\Gamma} t$ and $C \succ^{\mathcal{F}}_{\Gamma, x:B} \psi$. Hence $Q = \lambda x \lambda X : T(x, t).Q'$, so $\Delta_{\mathrm{Ax}}, \Delta, X : T(x, t) \vdash Q' : \psi$. Note that $\Gamma, x : B \succ \Delta, X : T(x, t)$. By Corollary 25 we have $\Gamma, x : B \vdash C : *^p$. Thus by the

inductive hypothesis there is M' with $\Gamma, x : B \vdash M' : C$. Since $\Gamma \vdash A : *^p$, by the abstraction rule we obtain $\Gamma \vdash (\lambda x : B.M') : A$. Hence take $M = \lambda x : B.M'$.

- $\varphi = P(t_A)$ with $A \succ^{\mathcal{C}}_{\Gamma} t_A$. Then $Q = XD_1 \dots D_k$ where $(X : \psi) \in \Delta_{\mathrm{Ax}}, \Delta$ and $\mathrm{target}(\psi) = P$, and each D_i is a first-order proof term in η-long normal form or an individual term. By the inductive hypothesis all proof terms among $D_1, \dots, D_k$ are reconstructible. We consider possible forms of ψ.
  - $(X : \psi) \in \Delta$. By Lemma 70 there are $\Gamma_1, \Gamma_2, \Delta_1, \Delta_2$ and C such that $\Gamma_1 \succ \Delta_1$ and $\Gamma = \Gamma_1, x : C, \Gamma_2$ and $\Delta = \Delta_1, \psi, \Delta_2$ and $\Gamma_1 \vdash C : *^p$ and $C \succ^{\mathcal{F}}_{\Gamma_1} \psi$. By Lemma 72 we have $C = \Pi x_1 : A_1 \dots \Pi x_n : A_n.B$ and $P(t) \succ^{\mathbb{A}}_{\Gamma_1,\Gamma_0} \psi$ and $B \succ^{\mathcal{C}}_{\Gamma_1,\Gamma_0} t$ where $\Gamma_0 = x_1 : A_1, \dots, x_n : A_n$. We may assume that $x_1, \dots, x_n \notin \mathrm{dom}(\Gamma)$. Hence Γ, Γ_0 is a legal context and $\Gamma \supseteq \Gamma_1$, so $P(t) \succ^{\mathbb{A}}_{\Gamma;\Gamma_0} \psi$ by Corollary 63. By Lemma 74 there are $N_1, \dots, N_n$ and $u_1, \dots, u_n$ such that $\varphi = P(t_A) = P(t)[\vec{u}/\vec{x}]$, i.e. $t_A = t[\vec{u}/\vec{x}]$, and $\Gamma, \Gamma_0 \rightsquigarrow_{\vec{x},\vec{N},\vec{u}} \Gamma$. By Lemma 62 we have $B \succ^{\mathcal{C}}_{\Gamma,\Gamma_0} t$. Hence $B[\vec{N}/\vec{x}] \succ^{\mathcal{C}}_{\Gamma} t[\vec{u}/\vec{x}] = t_A$ by Corollary 66. Since also $A \succ^{\mathcal{C}}_{\Gamma} t_A$, by Lemma 69 we obtain $A =_{\varepsilon} B[\vec{N}/\vec{x}]$. Because $\Gamma, \Gamma_0 \rightsquigarrow_{\vec{x},\vec{N},\vec{u}} \Gamma$ we must have $\Gamma \vdash N_i : A_i[N_1/x_1] \dots [N_{i-1}/x_{i-1}]$ and $N_i \sim x_i$ for $i = 1, \dots, n$. Recall that $\Gamma \vdash x : \Pi x_1 : A_1 \dots \Pi x_n : A_n.B$. Hence, using the application rule n times we conclude that $\Gamma \vdash xN_1 \dots N_n : B[\vec{N}/\vec{x}]$. Thus $\Gamma \vdash xN_1 \dots N_n : A$ by the conversion rule.
  - $(X : \psi) \in \Delta_E$ and $\psi = \forall xx'.E(x, x') \to P(x) \to P(x')$. Then $k = 4$, $D_1 = t_1$, $D_2 = t_2$ are individual terms, and $\Delta_{\mathrm{Ax}}, \Delta \vdash D_3 : E(t_1, t_2)$ and $\Delta_{\mathrm{Ax}}, \Delta \vdash D_4 : P(t_1)$ and $P(t_2) = P(t_A)$. Hence $t_2 = t_A$. Because D_3 is reconstructible (by induction), there is a Γ-term B with $B \succ^{\mathcal{C}}_{\Gamma} t_1$ and $B =_{\beta\varepsilon} A$. By Lemma 31 we have $\Gamma \vdash B : *^p$. Since $\Delta_{\mathrm{Ax}}, \Delta \vdash D_4 : P(t_1)$ and $B \succ^{\mathcal{C}}_{\Gamma} t_1$ and $\Gamma \vdash B : *^p$ and $\Gamma \succ \Delta$, because D_4 is reconstructible there is M with $\Gamma \vdash M : B$. By the conversion rule also $\Gamma \vdash M : A$.
  - $(X : \psi) \in \Delta_{\Phi}$ and $\psi = \forall \vec{y}.\psi' \to P(f\vec{y})$ and $\vec{y} = \mathrm{FV}(\psi')$ and $f = \Phi(\psi')$. Then $t_A = f\vec{t}$ and $\Delta_{\mathrm{Ax}}, \Delta \vdash X\vec{t}D_k : P(f\vec{t})$ and $\Delta_{\mathrm{Ax}}, \Delta \vdash D_k : \psi'[\vec{t}/\vec{y}]$ for some individual terms $t_1, \dots, t_n$ (without loss of generality we may assume that $\mathrm{FV}(t_i) \cap \mathrm{FV}(\psi') = \emptyset$). Since $A \succ^{\mathcal{C}}_{\Gamma} f\vec{t}$, by the definition of $\succ^{\mathcal{C}}_{\Gamma}$ there exist B, C and $N_1, \dots, N_m$ and $u_1, \dots, u_m$ and Γ' such that $\Gamma' \vdash (\Pi x : B.C) : *^p$ and $\Gamma' \rightsquigarrow_{\vec{x},\vec{N},\vec{u}} \Gamma$ and $A = (\Pi x : B.C)[\vec{N}/\vec{x}] \succ^{\mathcal{F}}_{\Gamma} \psi'[\vec{u}/\vec{x}]$ and $(\Pi x : B.C)[\vec{N}/\vec{x}] \succ^{\mathcal{C}}_{\Gamma} (f\vec{y})[\vec{u}/\vec{x}] = f\vec{t}$. Let $u'_i = u_i[u_{i+1}/x_{i+1}] \dots [u_m/x_m]$. Because $f\vec{t} = (f\vec{y})[u'_1/x_1, \dots, u'_m/x_m]$ and $\vec{y} = \mathrm{FV}(\psi') = \{y_1, \dots, y_n\}$, without loss of generality we may assume $u'_i = t_i$ and $x_i = y_i$ for $i \le n$, and $x_i \notin \mathrm{FV}(\psi')$ for $i > n$. Then $\psi'[\vec{u}/\vec{x}] = \psi'[u'_1/x_1, \dots, u'_m/x_m] = \psi'[u'_1/y_1, \dots, u'_n/y_n] = \psi'[\vec{t}/\vec{y}]$. By Lemma 61 we have $\Gamma \vdash (\Pi x : B.C)[\vec{N}/\vec{x}] : *^p$. Since also $A = (\Pi x : B.C)[\vec{N}/\vec{x}] \succ^{\mathcal{F}}_{\Gamma} \psi'[\vec{t}/\vec{y}]$ and $\Delta_{\mathrm{Ax}}, \Delta \vdash D_k : \psi'[\vec{t}/\vec{y}]$, because D_k is reconstructible there exists M such that $\Gamma \vdash M : A$.

2. Assume $\Delta_{\mathrm{Ax}}, \Delta \vdash Q : T(t, t')$ and $\Gamma \succ \Delta$ and $\Gamma \vdash A : s$ and $s \neq *^p$ and $A \succ^{\mathcal{C}}_{\Gamma} t'$. We need to find M with $M \succ^{\mathcal{C}}_{\Gamma} t$ and $\Gamma \vdash M : A$. Since $T(t, t')$ is an atom and Q is in η-lnf, we have $Q = X\vec{D}$ where $(X : \psi) \in \Delta_{\mathrm{Ax}}, \Delta$ and $\mathrm{target}(\psi) = T$ and $\vec{D}$ is a sequence of first-order individual terms and reconstructible (by induction) proof terms in η-lnf. We consider possible forms of ψ.
   - $(X : \psi) \in \Delta$. By Lemma 71 we have $\psi = T(x, r)$ and there are Γ_1, Γ_2 and C such that $\Gamma = \Gamma_1, x : C, \Gamma_2$ and $C \succ^{\mathcal{C}}_{\Gamma_1} r$. Then $t = x$ and $t' = r$. By Lemma 62 we have $C \succ^{\mathcal{C}}_{\Gamma} t'$. Since also $A \succ^{\mathcal{C}}_{\Gamma} t'$, by Lemma 69 we obtain $A =_{\varepsilon} C$. Because $\Gamma \vdash x : C$ and $\Gamma \vdash A : s$, by the conversion rule $\Gamma \vdash x : A$.
   - $(X : \psi) \in \Delta_{\mathcal{G}_1}$ and $\psi = \mathbb{A}_{\Gamma'}(\forall z.T(z, f\vec{y}) \to \forall x.T(x, r_1) \to T(zx, r_2))$ where $f = \mathcal{G}_1(x, r_1, r_2, s)$ and $\vec{y} = \mathrm{FV}(r_1, r_2) \setminus \{x\}$ and $z \notin \mathrm{FV}(r_1, r_2)$ and $\Gamma' = x_1 : A_1, \dots, x_n : A_n$. Then $Q = X\vec{R}uP_1wP_2$ and there are C_1, C_2 such that $r_1 = \mathcal{C}_{\Gamma'}(C_1)$ and $r_2 =$

$\mathcal{C}_{\Gamma',x:C_1}(C_2)$ and $\Gamma' \vdash (\Pi x : C_1.C_2) : s$ and $s \neq *^p$ and $\Gamma' \nvdash C_1 : *^p$. Note that ψ is closed, because $\vec{y} = \mathrm{FV}(r_1, r_2) \setminus \{x\} \subseteq \mathrm{dom}(\Gamma')$ by Lemma 68. Hence, by Lemma 53 we may assume $\mathrm{dom}(\Gamma) \cap \mathrm{dom}(\Gamma') = \emptyset$, possibly renaming the variables in Γ', ψ, r_1, r_2 and $\Pi x : C_1.C_2$. Hence Γ, Γ' is a legal context. Let $\psi' = \forall z.T(z, f\vec{y}) \to \forall x.T(x, r_1) \to T(zx, r_2)$. By Lemma 59 and Corollary 63 we have $\psi' \succ^{\mathbb{A}}_{\Gamma;\Gamma'} \psi$. By Lemma 74 there are $N_1, \ldots, N_n$ and $u_1, \ldots, u_n$ such that $\Gamma, \Gamma' \rightsquigarrow_{\vec{x},\vec{N},\vec{u}} \Gamma$ and $\Delta_{\mathrm{Ax}}, \Delta \vdash X\vec{R} : \psi'[\vec{u}/\vec{x}]$. Note that $z \notin \vec{y}$. Hence $\Delta_{\mathrm{Ax}}, \Delta \vdash P_1 : T(u, (f\vec{y})[\vec{u}/\vec{x}])$. Let $C_i' = C_i[\vec{N}/\vec{x}]$. By Lemma 56 and Lemma 60 and Corollary 66 we have $C_1' \succ^{\mathcal{C}}_{\Gamma} r_1[\vec{u}/\vec{x}]$ and $C_2' \succ^{\mathcal{C}}_{\Gamma,x:C_1[\vec{N}/\vec{x}]} r_2[\vec{u}/\vec{x}]$. Also $\Gamma, \Gamma' \vdash (\Pi x : C_1.C_2) : s$ by the thinning lemma, and $\Gamma, \Gamma' \nvdash C_1 : *^p$ by the generation, thinning and uniqueness of types lemmas. Hence $\Pi x : C_1'.C_2' \succ^{\mathcal{C}}_{\Gamma} (f\vec{y})[\vec{u}/\vec{x}]$. We also have $\Gamma \vdash (\Pi x : C_1'.C_2') : s$ (and $s \neq *^p$) by Lemma 61. Because P_1 is reconstructible, there is M_1 with $\Gamma \vdash M_1 : (\Pi x : C_1'.C_2')$ and $M_1 \succ^{\mathcal{C}}_{\Gamma} u$. Note that $x \notin \mathrm{FV}(r_1)$ by Lemma 68, because $C_1 \succ^{\mathcal{C}}_{\Gamma'} r_1$ and $x \notin \mathrm{dom}(\Gamma')$. Since also $z \notin \mathrm{FV}(r_1)$, we have $\Delta_{\mathrm{Ax}}, \Delta \vdash P_2 : T(w, r_1[\vec{u}/\vec{x}])$. Since $\Gamma, \Gamma' \nvdash C_1 : *^p$ and $\Gamma, \Gamma' \vdash (\Pi x : C_1.C_2) : s$, by the generation lemma and Lemma 61 we have $\Gamma \vdash C_1' : s'$ for some $s' \in \mathcal{S}$, $s' \neq *^p$. Since also $C_1' \succ^{\mathcal{C}}_{\Gamma} r_1[\vec{u}/\vec{x}]$, because P_2 is reconstructible there is M_2 with $M_2 \succ^{\mathcal{C}}_{\Gamma} w$ and $\Gamma \vdash M_2 : C_1'$. By Lemma 32 we have $M_2 \sim x$. Hence $\Gamma \vdash M_1M_2 : C_2'[M_2/x]$ by the application rule. Because M_1 is not a Γ-proof (recall that $\Gamma \vdash M_1 : (\Pi x : C_1'.C_2') : s$ with $s \neq *^p$), neither is M_1M_2 by Theorem 30. Hence $M_1M_2 \succ^{\mathcal{C}}_{\Gamma} uw = t$. Since $C_2' \succ^{\mathcal{C}}_{\Gamma,x:C_1'} r_2[\vec{u}/\vec{x}]$, by Lemma 65 we have $C_2'[M_2/x] \succ^{\mathcal{C}}_{\Gamma} r_2[\vec{u}/\vec{x}][u/x] = t'$. Since also $A \succ^{\mathcal{C}}_{\Gamma} t'$, we have $C_2'[M_2/x] =_{\varepsilon} A$ by Lemma 69. Thus $\Gamma \vdash M_1M_2 : A$ by the conversion rule. Therefore, we may take $M = M_1M_2$.

- $(X : \psi) \in \Delta_{\mathcal{G}_0}$. This case is analogous to the previous one.
- $(X : \psi) \in \Delta_{\tau_1}$ and $\psi = \mathbb{A}_{\Gamma'}(T(f\vec{y}, g\vec{z}))$ where $f = \Lambda_1(x, r, u)$ and $g = \mathcal{G}_1(x, r, w, s)$ and $\vec{y} = \mathrm{FV}(r, u) \setminus \{x\}$ and $\vec{z} = \mathrm{FV}(r, w) \setminus \{x\}$ and $\Gamma' = x_1 : A_1, \ldots, x_n : A_n$. Then $Q = X\vec{R}$ and there are C_1, C_2, N such that $r = \mathcal{C}_{\Gamma'}(C_1)$ and $u = \mathcal{C}_{\Gamma',x:C_1}(N)$ and $w = \mathcal{C}_{\Gamma',x:C_1}(C_2)$ and $\Gamma' \vdash (\lambda x : C_1.N) : \Pi x : C_1.C_2 : s$ and $\Gamma' \nvdash C_1 : *^p$. By Lemma 53 we may assume that $\mathrm{dom}(\Gamma) \cap \mathrm{dom}(\Gamma') = \emptyset$, possibly renaming the variables in Γ', ψ, r, u, w and $\Pi x : C_1.C_2$ and $\lambda x : C_1.N$. Hence Γ, Γ' is a legal context. Let $\psi' = T(f\vec{y}, g\vec{z})$. By Lemma 59 and Corollary 63 we have $\psi' \succ^{\mathbb{A}}_{\Gamma;\Gamma'} \psi$. By Lemma 74 there are $N_1, \ldots, N_n$ and $u_1, \ldots, u_n$ such that $\Gamma, \Gamma' \rightsquigarrow_{\vec{x},\vec{N},\vec{u}} \Gamma$ and $\Delta_{\mathrm{Ax}}, \Delta \vdash X\vec{R} : \psi'[\vec{u}/\vec{x}]$. We thus have $t = (f\vec{y})[\vec{u}/\vec{x}]$ and $t' = (g\vec{z})[\vec{u}/\vec{x}]$. By the generation lemma, the thinning lemma and the uniqueness of types lemma $\Gamma, \Gamma' \nvdash C_1 : *^p$. By the generation lemma C_1 is a Γ'-subject and N is a $\Gamma', x : C_1$-subject. Hence $C_1[\vec{N}/\vec{x}] \succ^{\mathcal{C}}_{\Gamma} r[\vec{u}/\vec{x}]$ and $N[\vec{N}/\vec{x}] \succ^{\mathcal{C}}_{\Gamma,x:C_1[\vec{N}/\vec{x}]} u[\vec{u}/\vec{x}]$ by Lemma 56 and Corollary 66. Hence by Definition 55 we have $(\lambda x : C_1.N)[\vec{N}/\vec{x}] \succ^{\mathcal{C}}_{\Gamma} (f\vec{y})[\vec{u}/\vec{x}]$. Also $\Gamma \vdash (\lambda x : C_1.N)[\vec{N}/\vec{x}] : C[\vec{N}/\vec{x}]$ by Lemma 60, where $C = \Pi x : C_1.C_2$. We have $C_2[\vec{N}/\vec{x}] \succ^{\mathcal{C}}_{\Gamma,x:C_1[\vec{N}/\vec{x}]} u[\vec{u}/\vec{x}]$ by Lemma 56 and Corollary 66. Also $\Gamma, \Gamma' \vdash C : s$ by the thinning lemma. Thus $C[\vec{N}/\vec{x}] \succ^{\mathcal{C}}_{\Gamma} (g\vec{z})[\vec{u}/\vec{x}] = t'$ by Definition 55. Since also $A \succ^{\mathcal{C}}_{\Gamma} t'$, by Lemma 69 we obtain $A =_{\varepsilon} C[\vec{N}/\vec{x}]$. Thus $\Gamma \vdash (\lambda x : C_1.N)[\vec{N}/\vec{x}] : A$ by the conversion rule. So we may take $M = (\lambda x : C_1.N)[\vec{N}/\vec{x}]$.
- $(X : \psi) \in \Delta_{\tau_0}$. This case is analogous to the previous one.
- $(X : \psi) \in \Delta_E$. Then $\psi = \forall xyx'y'.E(x, x') \to E(y, y') \to T(x, y) \to T(x', y')$ and $Q = Xuu'tt'D_1D_2D_3$ where $\Delta_{\mathrm{Ax}}, \Delta \vdash D_1 : E(u, t)$ and $\Delta_{\mathrm{Ax}}, \Delta \vdash D_2 : E(u', t')$ and $\Delta_{\mathrm{Ax}}, \Delta \vdash D_3 : T(u, u')$. Since $A \succ^{\mathcal{C}}_{\Gamma} t'$, because D_2 is reconstructible there exists a Γ-term A' such that $A' \succ^{\mathcal{C}}_{\Gamma} u'$ and $A' =_{\beta\varepsilon} A$. Since $\Gamma \vdash A : s$ $(s \neq *^p)$, by Lemma 31 we have $\Gamma \vdash A' : s$. Hence, because D_3 is reconstructible there exists M' such that

$M' \succ^{\mathcal{C}}_{\Gamma} u$ and $\Gamma \vdash M' : A'$. Because D_1 is reconstructible there is a Γ-subject M with $M \succ^{\mathcal{C}}_{\Gamma} t$ and $M =_{\beta\varepsilon} M'$. By the uniqueness of types lemma and Theorem 30 we have $M' \not\twoheadrightarrow^*_\varepsilon \varepsilon$, hence also $M \not\twoheadrightarrow^*_\varepsilon \varepsilon$ by Lemma 18. Since M is a Γ-subject, there is B with $\Gamma \vdash M : B$. Then $B =_{\beta\varepsilon} A'$ by the second point in the uniqueness of types lemma. Since $\Gamma \vdash A : s$ and $B =_{\beta\varepsilon} A' =_{\beta\varepsilon} A$, we have $\Gamma \vdash M : A$ by the conversion rule. Therefore, we have found M with $M \succ^{\mathcal{C}}_{\Gamma} t$ and $\Gamma \vdash M : A$, as desired.

3. Assume $\Delta_{\mathrm{Ax}}, \Delta \vdash Q : E(t_0, t_1)$ and $\Gamma \succ \Delta$ and $M \succ^{\mathcal{C}}_{\Gamma} t_q$ with $q \in \{0, 1\}$. We need to find N with $N \succ^{\mathcal{C}}_{\Gamma} t_{1-q}$ and $N =_{\beta\varepsilon} M$. Since $E(t_0, t_1)$ is an atom and Q is in η-lnf, $Q = X\vec{D}$ where $(X : \psi) \in \Delta_{\mathrm{Ax}}, \Delta$ and $\mathrm{target}(\psi) = E$ and $\vec{D}$ is a sequence of first-order individual terms and reconstructible (by induction) proof terms in η-lnf. We consider possible forms of ψ.
   - $(X : \psi) \in \Delta_{\Lambda_0}$ and $\psi = \mathbb{A}_{\Gamma'}(\varphi \to E(f\vec{y}\varepsilon, r))$ where $\Gamma' = x_1 : A_1, \ldots, x_n : A_n$ and $\mathrm{dom}(\Gamma) \cap \mathrm{dom}(\Gamma') = \emptyset$ (we may assume this by Lemma 53) and $f = \Lambda_0(x, \varphi, r)$ and $\vec{y} = \mathrm{FV}(\varphi, r) \setminus \{x\}$ and there are B, C_1, C_2 with $\varphi = \mathcal{F}_{\Gamma'}(C_1)$ and $r = \mathcal{C}_{\Gamma', x:C_1}(C_2)$ and $\Gamma' \vdash (\lambda x : C_1.C_2) : B$ and $\Gamma' \vdash C_1 : *^p$ and $\Gamma' \vdash B : s$ and $s \neq *^p$. We have $\psi = \mathbb{A}_{\Gamma', x:C_1}(E(f\vec{y}\varepsilon, r))$. We may assume $x \notin \mathrm{dom}(\Gamma)$, so $\Gamma, \Gamma', x : C_1$ is a legal context. Thus by Lemma 59 and Corollary 63 we obtain $E(f\vec{y}\varepsilon, r) \succ^{\mathbb{A}}_{\Gamma;\Gamma', x:C_1} \psi$. Hence by Lemma 74 there are $N_1, \ldots, N_n, U$ and $u_1, \ldots, u_n, u$ such that $E(t_0, t_1) = E(f\vec{y}\varepsilon, r)[\vec{u}/\vec{x}][u/x]$ and $\Gamma, \Gamma', x : C_1 \rightsquigarrow_{\vec{x}, x, \vec{N}, U, \vec{u}, u} \Gamma$. Note that then also $\Gamma, \Gamma' \rightsquigarrow_{\vec{x}, \vec{N}, \vec{u}} \Gamma$ and $\Gamma, \Gamma', x : C_1 \rightsquigarrow_{\vec{x}, \vec{N}, \vec{u}} \Gamma, x : C_1[\vec{N}/\vec{x}]$. We have $C_1 \succ^{\mathcal{F}}_{\Gamma, \Gamma'} \varphi$ and $C_2 \succ^{\mathcal{C}}_{\Gamma, \Gamma', x:C_1} r$ by Lemma 56 and Lemma 62. Hence $C_1[\vec{N}/\vec{x}] \succ^{\mathcal{F}}_{\Gamma} \varphi[\vec{u}/\vec{x}]$ and $C_2[\vec{N}/\vec{x}] \succ^{\mathcal{C}}_{\Gamma, x:C_1[\vec{N}/\vec{x}]} r[\vec{u}/\vec{x}]$ by Corollary 66. Also $\Gamma, \Gamma' \vdash (\lambda x : C_1.C_2) : B : s$ and $\Gamma, \Gamma' \vdash C_1 : *^p$ by the thinning lemma. Hence $\Gamma \vdash (\lambda x : C_1.C_2)[\vec{N}/\vec{x}] : B[\vec{N}/\vec{x}] : s$ by Lemma 61, i.e., $(\lambda x : C_1.C_2)[\vec{N}/\vec{x}]$ is not a Γ-proof (by the uniqueness of types lemma, recalling that $s \neq *^p$). Thus $(\lambda x : C_1.C_2)[\vec{N}/\vec{x}] \succ^{\mathcal{C}}_{\Gamma} (f\vec{y})[\vec{u}/\vec{x}]$. Since $N_i \succ^{\mathcal{C}}_{\Gamma} u_i$ and N_i is a Γ-term and $x \notin \mathrm{dom}(\Gamma)$, by Lemma 68 and the free variable lemma we obtain $x \notin \mathrm{FV}(u_1, \ldots, u_n)$. Also $x \notin \vec{y}$. Hence $(f\vec{y})[\vec{u}/\vec{x}] = (f\vec{y})[\vec{u}/\vec{x}][u/x]$. Because $\Gamma, \Gamma', x : C_1 \rightsquigarrow_{\vec{x}, x, \vec{N}, U, \vec{u}, u} \Gamma$, we have $U \sim x$ and $\Gamma \vdash U : C_1[\vec{N}/\vec{x}] : *^p$. Hence U is a Γ-proof, and thus $U \succ^{\mathcal{C}}_{\Gamma} \varepsilon$. Because $(\lambda x : C_1.C_2)[\vec{N}/\vec{x}]$ is not a Γ-proof, neither is $((\lambda x : C_1.C_2)[\vec{N}/\vec{x}])U$, by Theorem 30. Therefore $((\lambda x : C_1.C_2)[\vec{N}/\vec{x}])U \succ^{\mathcal{C}}_{\Gamma} ((f\vec{y})[\vec{u}/\vec{x}][u/x])\varepsilon = t_0$. We also have $C_2[\vec{N}/\vec{x}][U/x] \succ^{\mathcal{C}}_{\Gamma} r[\vec{u}/\vec{x}][u/x] = t_1$ by Corollary 66. Note that $((\lambda x : C_1.C_2)[\vec{N}/\vec{x}])U =_\beta C_2[\vec{N}/\vec{x}][U/x]$. First assume $q = 0$, i.e., $M \succ^{\mathcal{C}}_{\Gamma} t_0$. Using Lemma 69 we obtain $M =_{\beta\varepsilon} C_2[\vec{N}/\vec{x}][U/x] \succ^{\mathcal{C}}_{\Gamma} t_1$. Now assume $q = 1$, i.e., $M \succ^{\mathcal{C}}_{\Gamma} t_1$. Using Lemma 69 we obtain $M =_{\beta\varepsilon} ((\lambda x : C_1.C_2)[\vec{N}/\vec{x}])U \succ^{\mathcal{C}}_{\Gamma} t_0$. Also $((\lambda x : C_1.C_2)[\vec{N}/\vec{x}])U$ and $C_2[\vec{N}/\vec{x}][U/x]$ are Γ-subjects, by the generation lemma, the application rule (recall that $U \sim x$) and the subject reduction theorem.
   - $(X : \psi) \in \Delta_{\Lambda_1}$. This case is analogous to the case $(X : \psi) \in \Delta_{\Lambda_0}$.
   - $(X : \psi) \in \Delta_E$ and $\psi = \forall x.E(x, x)$. This case follows from reflexivity of $=_{\beta\varepsilon}$.
   - $(X : \psi) \in \Delta_E$ and $\psi = \forall xy.E(x, y) \to E(y, x)$. This case follows directly from the inductive hypothesis.
   - $(X : \psi) \in \Delta_E$ and $\psi = \forall xyz.E(x, y) \to E(y, z) \to E(x, z)$. This case follows from the inductive hypothesis and the transitivity of $=_{\beta\varepsilon}$.
   - $(X : \psi) \in \Delta_E$ and $\psi = \forall xyx'y'.E(x, x') \to E(y, y') \to E(xy, x'y')$. Then $Q = Xuwu'w'D_1D_2$ and $\Delta_{\mathrm{Ax}}, \Delta \vdash D_1 : E(u, u')$ and $\Delta_{\mathrm{Ax}}, \Delta \vdash D_2 : E(w, w')$ and $t_0 = uw$ and $t_1 = u'w'$. Assume $q = 0$, i.e., $M \succ^{\mathcal{C}}_{\Gamma} uw$ (the case $q = 1$ is analogous). Then $M = M_1M_2$ with $M_1 \succ^{\mathcal{C}}_{\Gamma} u$ and $M_2 \succ^{\mathcal{C}}_{\Gamma} w$. Since M is a Γ-subject, by the generation lemma $\Gamma \vdash M_1 : \Pi z : A.B$ and $\Gamma \vdash M_2 : A$ and $M_2 \sim z$ for some A, B. Because D_1, D_2 are reconstructible there are Γ-subjects N_1, N_2 such that $N_i =_{\beta\varepsilon} M_i$ and $N_1 \succ^{\mathcal{C}}_{\Gamma} u'$

and $N_2 \succ^{\mathcal{C}}_{\Gamma} w'$. Because $\mathrm{nf}_{\varepsilon}(M) \neq \varepsilon$, also $\mathrm{nf}_{\varepsilon}(M_1) \neq \varepsilon$. Hence by the second point in the uniqueness of types lemma, the generation lemma and the conversion rule $\Gamma \vdash N_1 : \Pi z : A.B$. Without loss of generality we may assume $\mathrm{nf}_{\varepsilon}(M_2) \neq \varepsilon$, because otherwise $u' = w' = \varepsilon$ and we may take $N_2 = M_2$. If $\mathrm{nf}_{\varepsilon}(M_2) \neq \varepsilon$ then analogously as with M_1 we conclude $\Gamma \vdash N_2 : A$. Note that also $N_2 \sim z$, because $M_2 \sim z$ and $M_2 =_{\beta\varepsilon} N_2$. Hence $N_1 N_2$ is a Γ-subject by the application rule. Using Theorem 30 we may also conclude that $N_1 N_2$ is not a Γ-proof. Thus $M =_{\beta\varepsilon} N_1 N_2 \succ^{\mathcal{C}}_{\Gamma} u'w' = t_1$.

◀

6 Conclusions and related work

Below we make a few remarks on the embedding, the soundness proof and related work.

▶ Remark. In the literature there are various translations of languages with dependent types to less expressive logics, but as far as we know none of them are both shallow, include the Calculus of Constructions as the source formalism, and target first-order logic. The paper [16] defines a deep embedding of the Calculus of Constructions into a higher-order logic and shows it complete. In [17] a similar deep embedding of LF into a fragment of higher-order logic is shown sound and complete. The paper [21] shows how to simulate dependent types in higher-order logic.

In [27] a translation from first-order logic with dependent types into ordinary first-order logic is shown sound by model-theoretic methods. The aim of [27] is also to use the translation with first-order ATPs. The logic is much simpler than dependent type theory – it allows dependent types, but not function types, i.e., no λ-abstraction or partial application is possible.

The paper [29] defines a sound and complete deep embedding Tri of Martin-Löf's type theory into first-order logic. The embedding is deep in the sense that e.g. $b \in B$ is translated to $\mathrm{In}(b, B)$, so b is not erased. For a fragment F_2, which essentially disallows dependent function types as arguments, the translation may be optimized to a shallow one, i.e., $\mathrm{In}(b, B)$ is optimised to $\mathrm{Inh}(B)$. This restriction corresponds to disallowing quantifiers on the left side of implication, which makes it possible to prove soundness and completeness of the embedding. In contrast to our approach, since there is no separate sort of propositions, all terms inhabiting types are erased, not only those intuitively representing proofs of propositions.

The general ideas behind the translations in [29, 27] are broadly similar to ours, but our work is not a direct extension of any of them.

The paper [19] defines a translation Tr from λP to FOL in order to show a conservativity result. The general idea of Tr, to translate a dependent type $\Pi x : A.B$ into a quantification and an implication, is similar to how we translate piPTS propositions. Essentially, the translation Tr is defined only for terms that "originate from" an embedding of FOL into λP, not on arbitrary λP-terms.

In [26] an essentially deep embedding from LF to the higher-order hereditary Harrop language is shown sound and complete. It is deep because even in its optimised variant the proof terms are retained as additional arguments. On the other hand, it allows to omit more type guards than our translation.

The report [1] defines and proves sound a translation from a fragment of the dependently typed F* language to intuitionistic first-order logic. The soundness proof uses a broadly similar method to the one in this paper, using induction on first-order proof terms in η-long normal form. However, the considered language fragment is essentially simpler and the soundness proof does not have to deal with the problems mentioned at the beginning of

Section 5, or with proof irrelevance. On the other hand, the target fragment of first-order logic is richer and includes conjunction and falsity.

▶ Remark. Our soundness proof relies on proof-irrelevance incorporated into the piPTS conversion rule. Proof-irrelevance is necessary for the soundness of a shallow embedding. It is an open question if the embedding is sound for the Calculus of Constructions with proof-irrelevance expressed by axioms.

▶ Remark. Note the use of the function $\mathbb{A}_\Gamma$ in the axioms in Δ_{Λ_0}, Δ_{Λ_1}, $\Delta_{\mathcal{G}_0}$ and $\Delta_{\mathcal{G}_1}$. In contrast, for the axioms in Δ_Φ the use of $\mathbb{A}_\Gamma$ is not necessary – we may simply quantify over the free variables without requiring them a priori to have the right types. This is because they all occur in the target atom $P(f\vec{y})$ which in the soundness proof is assumed to encode a well-typed term. This is not necessarily true for the axioms which use $\mathbb{A}_\Gamma$, and our soundness proof cannot be easily adapted to avoid the use of $\mathbb{A}_\Gamma$.

Nonetheless, we expect that the use of $\mathbb{A}_\Gamma$ could be avoided without compromising soundness. For instance, the axioms in Δ_{Λ_1} could be $\forall\vec{y}x.T(x,r) \to E(f\vec{y}x,t)$ or even $\forall\vec{y}x.E(f\vec{y}x,t)$. We expect the embedding to remain sound after this modification, because we would essentially omit the type information only for free variables of subterms that are "lifted out" of already well-typed terms. The problems that arise in the study of such a modified embedding are broadly similar to problems that arise in the study of systems of illative combinatory logic [3, 13] or the "liberal" Pure Type Systems from [10]. Domain-free Pure Type Systems [7], domain-free variants of the Calculus of Inductive Constructions [4], the Implicit Calculus of Constructions [23] and generally the work on ignoring computationally irrelevant information also seem related.

In fact, in the practical translation from [15] we omit type information for free variables of the terms "lifted-out" by the translation. This may increase the success rate in some circumstances, as then the formulas are simpler and the ATPs do not need to prove too many well-typedness conditions. See [15, Section 5.6].

▶ Remark. In [8, 11] it is shown that in a translation from (polymorphic) many-sorted classical first-order logic to untyped classical first-order logic much of the type information may be omitted using monotonicity inference. The methods of the cited papers are model-theoretic, so they are probably not useful in our setting. Nonetheless, it is an interesting problem to investigate the possibility of adapting monotonicity inference to embeddings of constructive dependent type theory into first-order logic.

References

1 A. Aguirre. Towards a provably correct encoding from F^* to SMT. Technical report, INRIA, 2016.

2 H. Barendregt. Lambda calculi with types. In *Handbook of Logic in Computer Science*, volume 2, pages 118–310. Oxford University Press, 1992.

3 H. Barendregt, M. Bunder, and W. Dekkers. Systems of illative combinatory logic complete for first-order propositional and predicate calculus. *J. Symb. Logic*, 58(3):769–788, 1993.

4 B. Barras and B. Grégoire. On the role of type decorations in the calculus of inductive constructions. In *CSL 2005*, pages 151–166, 2005.

5 G. Barthe. The relevance of proof-irrelevance. In *ICALP'98*, pages 755–768, 1998.

6 G. Barthe, J. Hatcliff, and M.H. Sørensen. A notion of classical pure type system. *Electr. Notes Theor. Comput. Sci.*, 6:4–59, 1997.

7 G. Barthe and M.H. Sørensen. Domain-free pure type systems. *J. Funct. Program.*, 10(5):417–452, 2000.

8 J. Blanchette, S. Böhme, A. Popescu, and N. Smallbone. Encoding monomorphic and polymorphic types. *Logical Methods in Computer Science*, 12(4), 2016.

9 J. Blanchette, C. Kaliszyk, L. Paulson, and J. Urban. Hammering towards QED. *J. Formalized Reasoning*, 9(1):101–148, 2016.

10 M. Bunder and W. Dekkers. Pure type systems with more liberal rules. *J. Symb. Logic*, 66(4):1561–1580, 2001.

11 K. Claessen, A. Lillieström, and N. Smallbone. Sort it out with monotonicity. In *CADE 2011*, pages 207–221. Springer, 2011.

12 T. Coquand and H. Herbelin. A-translation and looping combinators in pure type systems. *J. Funct. Program.*, 4(1):77–88, 1994.

13 Ł. Czajka. Higher-order illative combinatory logic. *J. Symb. Logic*, 73(3):837–872, 2013.

14 Ł. Czajka and C. Kaliszyk. Goal translation for a hammer for Coq (extended abstract). In *HaTT 2016*, volume 210 of *EPTCS*, pages 13–20, 2016.

15 Ł. Czajka and C. Kaliszyk. Hammer for Coq: Automation for dependent type theory. *J. Autom. Reasoning*, 61(1-4):423–453, 2018.

16 A. Felty. Encoding the calculus of constructions in a higher-order logic. In *LICS '93*, pages 233–244, 1993.

17 A. Felty and D. Miller. Encoding a dependent-type lambda-calculus in a logic programming language. In *CADE '90*, pages 221–235, 1990.

18 H. Geuvers. *Logics and Type Systems*. PhD thesis, University of Nijmegen, 1993.

19 H. Geuvers and E. Barendsen. Some logical and syntactical observations concerning the first-order dependent type system lambda-P. *Mathematical Structures in Computer Science*, 9(4):335–359, 1999.

20 H. Geuvers and M.-J. Nederhof. Modular proof of strong normalization for the Calculus of Constructions. *J. Funct. Program.*, 1(2):155–189, 1991.

21 B. Jacobs and T. Melham. Translating dependent type theory into higher order logic. In *TLCA '93*, pages 209–229, 1993.

22 L. Kovács and A. Voronkov. First-order theorem proving and Vampire. In *CAV 2013*, pages 1–35, 2013.

23 A. Miquel. The implicit calculus of constructions. In *TLCA 2001*, pages 344–359, 2001.

24 A. Miquel and B. Werner. The not so simple proof-irrelevant model of CC. In *TYPES 2002*, volume 2646 of *LNCS*. Springer, 2003.

25 S. Schulz. System description: E 1.8. In *LPAR 2013*, pages 735–743, 2013.

26 Z. Snow, D. Baelde, and G. Nadathur. A meta-programming approach to realizing dependently typed logic programming. In *PPDP '10*, pages 187–198, 2010.

27 K. Sojakova and F. Rabe. Translating a dependently-typed logic to first-order logic. In *WADT 2008*, pages 326–341, 2008.

28 M.H. Sørensen and P. Urzyczyn. *Lectures on the Curry-Howard Isomorphism*, volume 149 of *Studies in Logic and the Foundations of Mathematics*. Elsevier, 2006.

29 T. Tammet and J.M. Smith. Optimized encodings of fragments of type theory in first order logic. In *TYPES '95*, pages 265–287, 1995.

30 B. Werner. On the strength of proof-irrelevant type theories. *Logical Methods in Computer Science*, 4(3:13):1–20, 2008.

A Properties of proof-irrelevant Pure Type Systems

In this appendix we develop the meta-theory of proof-irrelevant Pure Type Systems. The development follows [2, Section 5.2] and is mostly standard, except for one difficulty caused by the mismatch between $\beta\varepsilon$-reduction used in the conversion rule and β-reduction for which the subject reduction theorem holds.

First, we need some lemmas concerning ε-reduction, β-reduction and $\beta\varepsilon$-reduction. Note that $\to_\varepsilon$ is not closed under substitutions. As a result, neither is $\to_\beta$, because it depends on $\to_\varepsilon$ through the side condition $N \sim x$. For example, $M = (\lambda x^{*^p} : A.x^{*^p})y^{*^p} \to_\beta y^{*^p}$. But $M[*^p/y^{*^p}] \not\to_\beta *^p$ because $*^p \not\sim x^{*^p}$. However, both relations are closed under substitutions if the condition $N \sim x$ is required when substituting N for x.

▶ **Lemma 78.** *If $M \to_\varepsilon M'$ and $N \sim x$ then $M[N/x] \to_\varepsilon^* M'[N/x]$.*

Proof. Induction on M. The assumption $N \sim x$ is needed when $x \in V^{*^p}$ and $M = x \to_\varepsilon \varepsilon$. ◀

▶ **Lemma 79** (Confluence and strong normalisation of ε-reduction)**.** *ε-reduction is confluent and strongly normalising.*

Proof. It is obvious that ε-reduction is strongly normalising. One also easily checks that the reflexive closure of $\to_\varepsilon$ has the diamond property. ◀

▶ **Corollary 80.** *If $M \to_\varepsilon^* M'$ and $M \sim x$ then $M' \sim x$.*

▶ **Lemma 16.** *If $N \sim x$ then $\mathrm{nf}_\varepsilon(M[N/x]) = \mathrm{nf}_\varepsilon(M)[\mathrm{nf}_\varepsilon(N)/x]$.*

Proof. Note that $M[N/x] \to_\varepsilon^* \mathrm{nf}_\varepsilon(M)[\mathrm{nf}_\varepsilon(N)/x]$ by Lemma 78. It suffices to show that the latter term is in ε-normal form. Otherwise, $\mathrm{nf}_\varepsilon(M)$ must have a subterm of the form xt or $\lambda y.x$, and $\mathrm{nf}_\varepsilon(N) = \varepsilon$. But then $x \in V^{*^p}$, which contradicts the fact that $\mathrm{nf}_\varepsilon(M)$ is in ε-normal form. ◀

▶ **Lemma 81.** *If $M \sim x$ and $N \sim y$ then $M[N/y] \sim x$.*

Proof. Follows directly from Lemma 16. ◀

▶ **Lemma 82.** *If $M \to_\beta M'$ and $N \sim x$ then $M[N/x] \to_\beta M'[N/x]$.*

Proof. Induction on M, using Lemma 81. ◀

▶ **Lemma 83.** *If $M \to_{\beta\varepsilon}^* M'$ and $N \to_{\beta\varepsilon}^* N'$ and $N \sim x$ then $M[N/x] \to_{\beta\varepsilon}^* M'[N'/x]$.*

Proof. Using Lemma 78 and Lemma 82 repeatedly we obtain $M[N/x] \to_{\beta\varepsilon}^* M'[N/x]$. Since $N \to_{\beta\varepsilon}^* N'$, we have $M'[N/x] \to_{\beta\varepsilon}^* M'[N'/x]$. ◀

▶ **Lemma 84.** *If $M \to_\beta M_1$ and $M \to_\varepsilon M_2$ then there is M' with $M_1 \to_\varepsilon^* M'$ and $M_2 \to_{\beta\varepsilon} M'$.*

Proof. Induction on M. The interesting case is when $M = (\lambda x : A.B)C \to_\beta B[C/x] = M_1$. Then $C \sim x$. First assume $M_2 = (\lambda x : A.B)C'$ with $C \to_\varepsilon C'$. Then $C' \sim x$ by Corollary 80. Hence $M_2 \to_\beta B[C'/x]$. We also have $B[C/x] \to_\varepsilon^* B[C'/x]$, so we may take $M' = B[C'/x]$. Now assume $M_2 = (\lambda x : A.B')C$ with $B \to_\varepsilon B'$. Then $B[C/x] \to_\varepsilon^* B'[C/x]$ by Lemma 78. Also $M_2 \to_\beta B'[C/x]$, so we may take $M' = B'[C/x]$. Finally, assume $M_2 = \varepsilon C$ where $B = \varepsilon$ and $\lambda x : A.B \to_\varepsilon \varepsilon$. Then $M_1 = B[C/x] = \varepsilon$. Since $M_2 = \varepsilon C \to_\varepsilon \varepsilon$, we may take $M' = \varepsilon$.

The remaining cases are easy. ◀

▶ **Corollary 85.** *If $M \to_\beta^* M_1$ and $M \to_\varepsilon^* M_2$ then there is M' with $M_1 \to_\varepsilon^* M'$ and $M_2 \to_{\beta\varepsilon}^* M'$.*

▶ **Lemma 86.** *If $M \to_\beta N \to_\varepsilon^* \varepsilon$ then $M \to_\varepsilon^* \varepsilon$.*

Proof. Induction on the length of the reduction $N \to_\varepsilon^* \varepsilon$.

If $M = (\lambda x.M')Q$ and $N = M'[Q/x]$ and $Q \sim x$ then $\mathrm{nf}_\varepsilon(N) = \mathrm{nf}_\varepsilon(M')[\mathrm{nf}_\varepsilon(Q)/x]$ by Lemma 16. Hence $\mathrm{nf}_\varepsilon(M')[\mathrm{nf}_\varepsilon(Q)/x] = \varepsilon$. This is possible if either $\mathrm{nf}_\varepsilon(M') = \varepsilon$, or $\mathrm{nf}_\varepsilon(M') = x$ and $\mathrm{nf}_\varepsilon(Q) = \varepsilon$. If $\mathrm{nf}_\varepsilon(M') = \varepsilon$ then $M \to_\varepsilon^* \varepsilon$. In the other case $x \in V^{*^p}$ because $Q \sim x$ and $Q \to_\varepsilon^* \varepsilon$. Hence also $M \to_\varepsilon^* \varepsilon$.

If $M = M'Q$ and $N = N'Q$ then $M' \to_\beta N' \to_\varepsilon^* \varepsilon$. Then $M' \to_\varepsilon^* \varepsilon$ by the inductive hypothesis, and thus $M \to_\varepsilon^* \varepsilon$.

Otherwise $M = \lambda x.M'$ and $N = \lambda x.N'$ and $M' \to_\beta N'$. Then $M' \to_\beta N' \to_\varepsilon^* \varepsilon$, so by the inductive hypothesis $M' \to_\varepsilon^* \varepsilon$. Hence $M \to_\varepsilon^* \varepsilon$. ◀

▶ **Corollary 87.** *If $M \to_{\beta\varepsilon}^* M'$ then $M \sim x$ iff $M' \sim x$.*

▶ **Lemma 88** (Postponement of ε-reduction)**.** *If $M \to_{\beta\varepsilon}^* M'$ then there exists N such that $M \to_\beta^* N \to_\varepsilon^* M'$.*

Proof. One shows: if $M \to_\varepsilon N \to_\beta M'$ then there is N' with $M \to_\beta N' \to_\varepsilon^* M'$. This follows easily, using Lemma 78, because ε-reduction cannot create or duplicate β-redexes. ◀

▶ **Corollary 89** (β-reduction requests ε-reduction)**.** *If $M \to_\beta^* M_1$ and $M \to_\varepsilon^* M_2$ then there are M_2', M' with $M_1 \to_\varepsilon^* M'$ and $M_2 \to_\beta^* M_2' \to_\varepsilon^* M'$.*

▶ **Lemma 90** (Confluence of β-reduction)**.** *If $M \to_\beta^* M_1$ and $M \to_\beta^* M_2$ then there exists M' such that $M_1 \to_\beta^* M'$ and $M_2 \to_\beta^* M'$.*

Proof. By a straightforward adaptation of the Tait–Martin-Löf method. The parallel reduction relation $\to_1$ is defined as follows:

- $x \to_1 x$, $s \to_1 s$, $\varepsilon \to_1 \varepsilon$,
- if $M \to_1 M'$ and $N \to_1 N'$ and $N \sim x$ then $(\lambda x : A.M)N \to_1 M'[N'/x]$,
- if $M \to_1 M'$ and $N \to_1 N'$ then $MN \to_1 M'N'$,
- if $A \to_1 A'$ and $M \to_1 M'$ then $\lambda x : A.M \to_1 \lambda x : A'.M'$,
- if $A \to_1 A'$ and $M \to_1 M'$ then $\Pi x : A.M \to_1 \Pi x : A'.M'$.

One then shows:

1. if $M \to_1 M'$ and $N \to_1 N'$ and $N \sim x$ then $M[N/x] \to_1 M'[N'/x]$,
2. if $M \to_1 M_1$ and $M \to_1 M_2$ then there exists M' with $M_1 \to_1 M'$ and $M_2 \to_1 M'$.

The first point is shown by induction on M, using Lemma 81 when $M = (\lambda y : A.M_1)M_2 \to_1 M_1'[M_2'/y] = M'$. The second point is shown by a standard argument, using the first point and Corollary 87. Confluence of β-reduction then follows from the second point, because $\to_\beta \subseteq \to_1 \subseteq \to_\beta^*$. ◀

▶ **Lemma 17** (Confluence of $\beta\varepsilon$-reduction)**.** *If $M \to_{\beta\varepsilon}^* M_1$ and $M \to_{\beta\varepsilon}^* M_2$ then there exists M' such that $M_1 \to_{\beta\varepsilon}^* M'$ and $M_2 \to_{\beta\varepsilon}^* M'$.*

Proof. This follows from the confluence of β- and ε-reduction and the fact that β-reduction requests ε-reduction. More precisely, one shows that $\to_\beta^* \cdot \to_\varepsilon^*$ has the diamond property. See Figure 3. ◀

▶ **Corollary 91.** *If $M =_{\beta\varepsilon} M'$ and $N \sim x$ and $N =_{\beta\varepsilon} N'$ then $M[N/x] =_{\beta\varepsilon} M'[N'/x]$.*

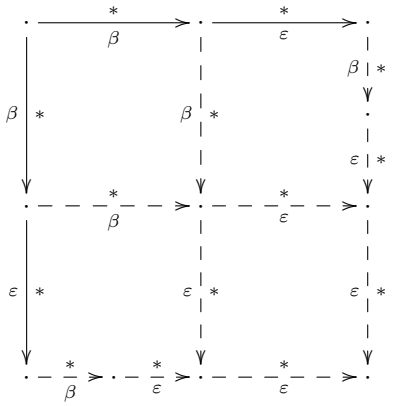

Figure 3 Confluence of $\beta\varepsilon$-reduction.

Note that confluence of $\beta\varepsilon$-reduction on arbitrary preterms would fail if we did not restrict β-reduction as in Definition 10. For example, for $M = (\lambda x^{*^p} : A.x^{*^p})*^p$ we would have $M \to_\varepsilon^* \varepsilon$ and $M \to_\beta *^p$.

▶ **Lemma 18.** *If $M =_{\beta\varepsilon} N$ then $M \to_\varepsilon^* \varepsilon$ is equivalent to $N \to_\varepsilon^* \varepsilon$.*

Proof. Suppose $M \to_\varepsilon^* \varepsilon$. By confluence of $\beta\varepsilon$-reduction $N \to_{\beta\varepsilon}^* \varepsilon$. So $N \to_\beta^* N' \to_\varepsilon^* \varepsilon$ by Lemma 88. Now by repeatedly applying Lemma 86 we obtain $N \to_\varepsilon^* \varepsilon$. ◀

▶ **Lemma 19.** *If N does not contain ε and $M \to_{\beta\varepsilon}^* N$ then $M \to_\beta^* N$.*

Proof. By postponement of ε-reduction there is M' with $M \to_\beta^* M' \to_\varepsilon^* N$. Because N does not contain ε, we must in fact have $M' = N$. ◀

Proofs of most of the following lemmas for ordinary PTSs may be found e.g. in [2, Section 5.2]. The proofs for piPTSs are essentially the same or very similar. We only briefly indicate how to carry out the proofs and note the differences with the standard proofs.

▶ **Lemma 20** (Free variable lemma)**.** *If $\Gamma = x_1 : A_1, \ldots, x_n : A_n$ and $\Gamma \vdash B : C$ then:*
1. *the $x_1, \ldots, x_n$ are all distinct,*
2. *$\mathrm{FV}(B), \mathrm{FV}(C) \subseteq \{x_1, \ldots, x_n\}$,*
3. *$\mathrm{FV}(A_i) \subseteq \{x_1, \ldots, x_{i-1}\}$ for $i = 1, \ldots, n$.*

Proof. Induction on the derivation $\Gamma \vdash B : C$. ◀

▶ **Lemma 21** (Start lemma)**.** *Let Γ be a legal context.*
1. *If $(s_1, s_2) \in \mathcal{A}$ then $\Gamma \vdash s_1 : s_2$.*
2. *If $(x : A) \in \Gamma$ then $\Gamma \vdash x : A$ and there is $s \in \mathcal{S}$ with $\Gamma_1 \vdash A : s$ and $x \in V^s$, where $\Gamma = \Gamma_1, x : A, \Gamma_2$.*

Proof. Since Γ is legal, $\Gamma \vdash B : C$ for some B, C. The lemma follows by induction on the length of the derivation of $\Gamma \vdash B : C$. ◀

▶ **Lemma 22** (Substitution lemma)**.** *If $\Gamma, x : A, \Gamma' \vdash B : C$ and $\Gamma \vdash D : A$ and $D \sim x$ then $\Gamma, \Gamma'[D/x] \vdash B[D/x] : C[D/x]$.*

Proof. Induction on the derivation of $\Gamma, x : A, \Gamma' \vdash B : C$. We need the assumption $D \sim x$ and Corollary 91 to treat the conversion rule. For the application rule we need Lemma 81. ◀

▶ **Lemma 23** (Thinning lemma). *If* $\Gamma \vdash A : B$ *and* $\Gamma' \supseteq \Gamma$ *is a legal context then* $\Gamma' \vdash A : B$.

Proof. Induction on the derivation of $\Gamma \vdash A : B$. ◀

▶ **Lemma 24** (Generation lemma).

1. *If* $\Gamma \vdash s : A$ *then there is* $s' \in \mathcal{S}$ *with* $A =_{\beta\varepsilon} s'$ *and* $(s, s') \in \mathcal{A}$.
2. *If* $\Gamma \vdash x : A$ *then there are* $s \in \mathcal{S}$ *and* B *such that* $A =_{\beta\varepsilon} B$ *and* $\Gamma \vdash B : s$ *and* $(x : B) \in \Gamma$ *and* $x \in V^s$.
3. *If* $\Gamma \vdash (\Pi x : A.B) : C$ *then there is* $(s_1, s_2, s_3) \in \mathcal{R}$ *with* $\Gamma \vdash A : s_1$ *and* $\Gamma, x : A \vdash B : s_2$ *and* $C =_{\beta\varepsilon} s_3$.
4. *If* $\Gamma \vdash (\lambda x : A.M) : C$ *then there are* $s \in \mathcal{S}$ *and* B *such that* $\Gamma \vdash (\Pi x : A.B) : s$ *and* $\Gamma, x : A \vdash M : B$ *and* $C =_{\beta\varepsilon} \Pi x : A.B$.
5. *If* $\Gamma \vdash MN : C$ *then there are* A, B *such that* $\Gamma \vdash M : (\Pi x : A.B)$ *and* $\Gamma \vdash N : A$ *and* $C =_{\beta\varepsilon} B[N/x]$ *and* $N \sim x$.

Proof. Completely analogous to the standard proof for ordinary PTSs, using the thinning lemma. ◀

▶ **Lemma 26** (Correctness of types lemma). *If* $\Gamma \vdash M : A$ *then there is* $s \in \mathcal{S}$ *such that* $A = s$ *or* $\Gamma \vdash A : s$.

Proof. Induction on the derivation $\Gamma \vdash M : A$. The non-obvious case is when the application rule is used. Then $M = M_1 M_2$ and $A = C[M_2/x]$ and $\Gamma \vdash M_1 : (\Pi x : B.C)$ and $\Gamma \vdash M_2 : B$ and $M_2 \sim x$. By the inductive hypothesis there is $s' \in \mathcal{S}$ such that $\Gamma \vdash (\Pi x : B.C) : s'$. By the generation lemma there is $s \in \mathcal{S}$ such that $\Gamma, x : B \vdash C : s$. Since $\Gamma \vdash M_2 : B$ and $M_2 \sim x$, by the substitution lemma we obtain $\Gamma \vdash C[M_2/x] : s$, so $\Gamma \vdash A : s$. Note that the side condition $M_2 \sim x$ in the application rule was necessary to carry out the proof. ◀

▶ **Theorem 29** (Subject reduction theorem). *If* $\Gamma \vdash A : B$ *and* $A \to_\beta^* A'$ *then* $\Gamma \vdash A' : B$.

Proof. Completely analogous to the standard proof, using the generation, correctness of types and substitution lemmas, and Corollary 91. To be able to apply the substitution lemma and Corollary 91 the side condition in the application rule is necessary. ◀

Subject reduction obviously does not hold for $\beta\varepsilon$-reduction, because ε is not meant to be typable. The following lemma is a direct consequence of subject reduction.

▶ **Lemma 92.** *If* $\Gamma \vdash M : A$ *and* $A =_{\beta\varepsilon} s$ *then* $\Gamma \vdash M : s$.

Proof. By confluence of $\beta\varepsilon$-reduction and Lemma 19 we have $A \to_\beta^* s$. By the correctness of types lemma $\Gamma \vdash A : s'$ or $A = s'$. If $A = s'$ then $s = s'$ and we are done. So assume $\Gamma \vdash A : s'$. Then $\Gamma \vdash s : s'$ by the subject reduction theorem. Hence $\Gamma \vdash M : s$ by the conversion rule. ◀

The mismatch between the β-reduction in the subject reduction theorem and the $\beta\varepsilon$-conversion in the conversion rule generates some difficulties in the meta-theory of piPTSs. In ordinary functional PTSs, it is a direct consequence of the subject reduction theorem and the uniqueness of types lemma (to be stated below) that if $\Gamma \vdash B : s$ and $B =_\beta B'$ and $\Gamma \vdash A' : B'$ then $\Gamma \vdash B' : s$. This can also be easily established for functional piPTSs, by a similar proof. But we would want a stronger analogous property with $\beta\varepsilon$-conversion instead of β-conversion. Then the standard argument breaks down, because subject reduction does not hold for $\beta\varepsilon$-reduction.

In particular, we are interested in showing that in a logical piPTS if M is a Γ-proof then $M \to_\varepsilon^* \varepsilon$. This presents a difficulty already when $M = x$. Then we have $\Gamma \vdash x : A : *^p$ for some A. But from this we cannot immediately conclude $x \in V^{*^p}$ because the derivation of $\Gamma \vdash x : A$ may end with the conversion rule. From the generation lemma we may only conclude that there are $s \in \mathcal{S}$ and $B =_{\beta\varepsilon} A$ such that $\Gamma \vdash B : s$ and $(x : B) \in \Gamma$ and $x \in V^s$. It would suffice if from $\Gamma \vdash B : s$ and $\Gamma \vdash A : *^p$ and $B =_{\beta\varepsilon} A$ we could conclude $s = *^p$. But this does not seem completely straightforward to establish without subject reduction for $\beta\varepsilon$-reduction. We will ultimately show this property using a slight sharpening of the uniqueness of types lemma for logical piPTSs.

Our next aim is to show that in a logical piPTS a Γ-type does not ε-reduce to ε. For this we need the following technical definition.

▶ **Definition 93.** We define the relation $B \rightsquigarrow_\Gamma C$ inductively:

- if $B =_{\beta\varepsilon} C$ then $B \rightsquigarrow_\Gamma C$,
- if there exist N and C' such that $\Gamma \vdash N : A$ and $N \sim x$ and $B[N/x] =_{\beta\varepsilon} C' \rightsquigarrow_\Gamma C$ then $\Pi x : A.B \rightsquigarrow_\Gamma C$.

▶ **Lemma 94.** *If $\Gamma \vdash B : s$ and $B =_{\beta\varepsilon} B' \rightsquigarrow_\Gamma C$ then $B \rightsquigarrow_\Gamma C$.*

Proof. If $B' =_{\beta\varepsilon} C$ then this is obvious. Otherwise $B' = \Pi x : A_0.B_0$ and $\Gamma \vdash N : A_0$ and $N \sim x$ and $B_0[N/x] =_{\beta\varepsilon} C' \rightsquigarrow_\Gamma C$. By the confluence of $\beta\varepsilon$-reduction we have $B = \Pi x : A_1.B_1$ with $A_1 =_{\beta\varepsilon} A_0$ and $B_1 =_{\beta\varepsilon} B_0$. Since $N \sim x$, by Corollary 91 we obtain $B_1[N/x] =_{\beta\varepsilon} B_0[N/x] =_{\beta\varepsilon} C'$. Because $\Gamma \vdash B : s$, by the generation lemma $\Gamma \vdash A_1 : s'$ for some $s' \in \mathcal{S}$. Hence $\Gamma \vdash N : A_1$ by the conversion rule. Thus $B \rightsquigarrow_\Gamma C$. ◀

▶ **Lemma 95.** *In a logical piPTS, if $\Gamma \vdash B : *^p$ and $B =_{\beta\varepsilon} B' \rightsquigarrow_\Gamma C$, then there exists C' such that $\Gamma \vdash C' : *^p$ and $C' =_{\beta\varepsilon} C$.*

Proof. Induction on the definition of $B' \rightsquigarrow_\Gamma C$. If $B' =_{\beta\varepsilon} C$ then this is obvious. Otherwise $B' = \Pi x : A_1.B_1$ and $\Gamma \vdash N : A_1$ and $N \sim x$ and $B_1[N/x] =_{\beta\varepsilon} C_1 \rightsquigarrow_\Gamma C$. By the confluence of $\beta\varepsilon$-reduction $B = \Pi x : A_0.B_0$ with $A_0 =_{\beta\varepsilon} A_1$ and $B_0 =_{\beta\varepsilon} B_1$. Because $\Gamma \vdash (\Pi x : A_0.B_0) : *^p$ and the piPTS is logical, by the generation lemma $\Gamma \vdash A_0 : s$ for some $s \in \mathcal{S}$ and $\Gamma, x : A_0 \vdash B_0 : *^p$. Since $\Gamma \vdash N : A_1$, by the conversion rule $\Gamma \vdash N : A_0$. Hence $\Gamma \vdash B_0[N/x] : *^p$ by the substitution lemma. By Corollary 91 we also have $B_0[N/x] =_{\beta\varepsilon} B_1[N/x]$. Thus $\Gamma \vdash B_0[N/x] : *^p$ and $B_0[N/x] =_{\beta\varepsilon} C_1$ and $C_1 \rightsquigarrow_\Gamma C$. We may therefore apply the inductive hypothesis to obtain C' with $\Gamma \vdash C' : *^p$ and $C' =_{\beta\varepsilon} C$. ◀

▶ **Lemma 96.** *In a logical piPTS, if $\Gamma \vdash M : s$ then $M \not\to_\varepsilon^* \varepsilon$.*

Proof. By induction on M we show that if $(\star)$ below holds for M then $M \not\to_\varepsilon^* \varepsilon$. Then taking $n = 0$ in $(\star)$ gives us the lemma.

$(\star)$ There exist $A_1, \ldots, A_n$ and $N_1, \ldots, N_n$ such that

$$\Gamma, x_1 : A_1, \ldots, x_n : A_n \vdash M : B$$

and $N_i \sim x_i$ and $\Gamma \vdash N_i : A_i[N_1/x_1] \ldots [N_{i-1}/x_{i-1}]$ for $i = 1, \ldots, n$ and

$$B[N_1/x_1] \ldots [N_n/x_n] \rightsquigarrow_\Gamma s.$$

Assume $(\star)$ and $M \to_\varepsilon^* \varepsilon$. Let $\Gamma' = \Gamma, x_1 : A_1, \ldots, x_n : A_n$. There are three possibilities.

1. $M = x \in V^{*^p}$. Then by the generation lemma there is B' with $B' =_{\beta\varepsilon} B$ and $\Gamma' \vdash B' : *^p$. Using the substitution lemma repeatedly we obtain $\Gamma \vdash B'[N_1/x_1]\dots[N_n/x_n] : *^p$. Using Corollary 91 repeatedly we obtain

   $$B'[N_1/x_1]\dots[N_n/x_n] =_{\beta\varepsilon} B[N_1/x_1]\dots[N_n/x_n] \rightsquigarrow_\Gamma s.$$

   Hence by Lemma 95 there is C with $\Gamma \vdash C : *^p$ and $C =_{\beta\varepsilon} s$. By Lemma 19 and the confluence of $\beta\varepsilon$-reduction we have $C \to_\beta^* s$. Hence by the subject reduction theorem we obtain $\Gamma \vdash s : *^p$. This contradicts the fact that the piPTS is logical.
2. $M = M_1 M_2$ with $M_1 \to_\varepsilon^* \varepsilon$. By the generation lemma there exist A_0 and B_0 such that $\Gamma' \vdash M_1 : (\Pi x : A_0.B_0)$ and $\Gamma' \vdash M_2 : A_0$ and $B =_{\beta\varepsilon} B_0[M_2/x]$ and $M_2 \sim x$. Let $M_2' = M_2[N_1/x_1]\dots[N_n/x_n]$. Using the substitution lemma repeatedly we obtain

   $$\Gamma \vdash M_2' : A_0[N_1/x_1]\dots[N_n/x_n].$$

   Also $M_2' \sim x$ by repeated use of Lemma 81. By the correctness of types and generation lemmas there is s' with $\Gamma', x : A_0 \vdash B_0 : s'$. Using the substitution lemma repeatedly with the N_i's and M_2' we obtain

   $$\Gamma \vdash B_0[N_1/x_1]\dots[N_n/x_n][M_2'/x] : s'.$$

   Using Corollary 91 repeatedly we also obtain

   $$B_0[M_2/x][N_1/x_1]\dots[N_n/x_n] =_{\beta\varepsilon} B[N_1/x_1]\dots[N_n/x_n] \rightsquigarrow_\Gamma s.$$

   By α-conversion we may assume $x \notin \mathrm{FV}(N_1, \dots, N_n)$. Thus

   $$B_0[M_2/x][N_1/x_1]\dots[N_n/x_n] = B_0[N_1/x_1]\dots[N_n/x_n][M_2'/x].$$

   Hence

   $$(\Pi x : A_0.B_0)[N_1/x_1]\dots[N_n/x_n] \rightsquigarrow_\Gamma s.$$

   Now applying the inductive hypothesis yields a contradiction.
3. $M = \lambda x : A_0.M'$ with $M' \to_\varepsilon^* \varepsilon$. By the generation lemma there are $s' \in \mathcal{S}$ and B_0 such that $\Gamma' \vdash (\Pi x : A_0.B_0) : s$ and $\Gamma', x : A_0 \vdash M' : B_0$ and $B =_{\beta\varepsilon} \Pi x : A_0.B_0$. By the confluence of $\beta\varepsilon$-reduction we have $B = \Pi x : C_0.D_0$ with $A_0 =_{\beta\varepsilon} C_0$ and $B_0 =_{\beta\varepsilon} D_0$. Let $A_0^{*^p} = A_0[N_1/x_1]\dots[N_n/x_n]$ and analogously for $B_0^{*^p}$, $C_0^{*^p}$ and $D_0^{*^p}$. Since $B[N_1/x_1]\dots[N_n/x_n] = \Pi x : C_0^{*^p}.D_0^{*^p} \rightsquigarrow_\Gamma s$, there is N with $\Gamma \vdash N : C_0^{*^p}$ and $N \sim x$ and $D_0^{*^p}[N/x] =_{\beta\varepsilon} C \rightsquigarrow_\Gamma s$ (the case $\Pi x : C_0^{*^p}.D_0^{*^p} =_{\beta\varepsilon} s$ is impossible by the confluence of $\beta\varepsilon$-reduction). By repeated use of Corollary 91 we have $A_0^{*^p} =_{\beta\varepsilon} C_0^{*^p}$ and $B_0^{*^p}[N/x] =_{\beta\varepsilon} D_0^{*^p}[N/x]$. Since $\Gamma' \vdash (\Pi x : A_0.B_0) : s'$, by the generation lemma there are $s_1, s_2 \in \mathcal{S}$ with $\Gamma' \vdash A_0 : s_1$ and $\Gamma', x : A_0 \vdash B_0 : s_2$. By repeated use of the substitution lemma $\Gamma \vdash A_0^{*^p} : s_1$. Since also $C_0^{*^p} =_{\beta\varepsilon} A_0^{*^p}$ and $\Gamma \vdash N : C_0^{*^p}$, by the conversion rule we have $\Gamma \vdash N : A_0^{*^p}$. Now by repeated use of the substitution lemma we obtain $\Gamma \vdash B_0^{*^p}[N/x] : s_2$. Since also $B_0^{*^p}[N/x] =_{\beta\varepsilon} C \rightsquigarrow s$, by Lemma 94 we obtain $B_0^{*^p}[N/x] \rightsquigarrow_\Gamma s$. Therefore, because $\Gamma', x : A_0 \vdash M' : B_0$ and $N \sim x$ and $\Gamma \vdash N : A_0[N_1/x_1]\dots[N_n/x_n]$ and $B_0[N_1/x_1]\dots[N_n/x_n][N/x] \rightsquigarrow_\Gamma s$, we may apply the inductive hypothesis to conclude $M' \not\to_\varepsilon^* \varepsilon$. This gives a contradiction.

◀

A simpler proof of Lemma 96 would be possible if we changed the definitions in one of the following two ways.

(1) In the definition of a logical piPTS, require $(s, *^p, *^p) \in \mathcal{R}$ for any $s \in \mathcal{S}$.
(2) In the definition of a piPTS, add the side condition $x \in V^{s_1}$ in the product rule, and restrict ε-reduction of lambda-abstractions to:

$$\lambda x^s : A.\varepsilon \quad \to_\varepsilon \quad \varepsilon \qquad \text{if } (s, *^p, *^p) \in \mathcal{R}$$

Then we could prove $(\star)$ below by a relatively straightforward induction, without relying on Lemma 99. Lemma 96 would then easily follow from $(\star)$.

$(\star)$ In a logical piPTS, if $\Gamma \vdash M : C$ and $M \to_\varepsilon^* \varepsilon$ then there is C' with $C' =_{\beta\varepsilon} C$ and $\Gamma \vdash M : C' : *^p$.

However, none of the changes (1) or (2) seem to allow avoiding the use of Lemma 99 in the proof of Lemma 100.

▶ **Definition 97.** An *n-ary term context* $C[\Box_1, \ldots, \Box_n]$ is a term with n holes into which some terms $N_1, \ldots, N_n$ may be substituted possibly capturing their free variables, yielding $C[N_1, \ldots, N_n]$. For example, $C[\Box_1, \Box_2] = \lambda xy.\Box_1\Box_2$ is a term context, and $C[x, xy] = \lambda xy.x(xy)$.

We write $\Gamma_1 =_\varepsilon \Gamma_2$ if $\Gamma_1 = x_1 : A_1, \ldots, x_n : A_n$ and $\Gamma_2 = x_1 : A'_1, \ldots, x_n : A'_n$ and $A_i =_\varepsilon A'_i$. The following simple lemma will be used implicitly.

▶ **Lemma 98.** *If $M =_\varepsilon M'$ then there are $x_1, \ldots, x_n$ and an n-ary term context $C[\Box_1, \ldots, \Box_n]$ such that $M = C[N_1, \ldots, N_n]$ and $M' = C[N'_1, \ldots, N'_n]$ and $N_i \to_\varepsilon^* \varepsilon$ and $N'_i \to_\varepsilon^* \varepsilon$.*

Proof. Follows from confluence of ε-reduction. ◀

For logical piPTSs we need a somewhat sharpened version of the uniqueness of types lemma.

▶ **Lemma 27** (Uniqueness of types lemma)**.**
1. *In a functional piPTS, if $\Gamma \vdash A : B$ and $\Gamma \vdash A : B'$ then $B =_{\beta\varepsilon} B'$.*
2. *In a logical piPTS, if $\Gamma \vdash M_1 : A_1$ and $\Gamma \vdash M_2 : A_2$ and $M_1 =_{\beta\varepsilon} M_2$ and $M_1 \not\to_\varepsilon^* \varepsilon$ and $M_2 \not\to_\varepsilon^* \varepsilon$ then $A_1 =_{\beta\varepsilon} A_2$.*

Proof. We show the second point. The proof of the first point is similar but simpler, and it is also completely analogous to the standard uniqueness of types proof for ordinary functional PTSs.

So assume the piPTS is logical. First, we show the following condition $(\star)$.

$(\star)$ If $\Gamma_1 \vdash M_1 : A_1$ and $\Gamma_2 \vdash M_2 : A_2$ and $M_1 =_\varepsilon M_2$ and $\Gamma_1 =_\varepsilon \Gamma_2$ and $M_1 \not\to_\varepsilon^* \varepsilon$ and $M_2 \not\to_\varepsilon^* \varepsilon$ then $A_1 =_{\beta\varepsilon} A_2$.

We proceed by induction on M_1. We have the following possibilities.

- $M_1 = s = M_2$. By the generation lemma there are $s_1, s_2 \in \mathcal{S}$ such that $A_1 =_{\beta\varepsilon} s_1$ and $A_2 =_{\beta\varepsilon} s_2$ and $(s, s_1), (s, s_2) \in \mathcal{A}$. Hence $s_1 = s_2$ because the piPTS is functional. Thus $A_1 =_{\beta\varepsilon} A_2$.
- $M_1 = x = M_2$. By the generation lemma there exist C_1, C_2 such that $A_1 =_{\beta\varepsilon} C_1$ and $A_2 =_{\beta\varepsilon} C_2$ and $(x : C_1) \in \Gamma_1$ and $(x : C_2) \in \Gamma_2$. Since $\Gamma_1 =_\varepsilon \Gamma_2$, we have $C_1 =_\varepsilon C_2$. Thus $A_1 =_{\beta\varepsilon} A_2$.
- $M_1 = \Pi x : B_1.C_1$ and $M_2 = \Pi x : B_2.C_2$ with $B_1 =_\varepsilon B_2$ and $C_1 =_\varepsilon C_2$. By the generation lemma there exist $(s_1, s_2, s_3), (s'_1, s'_2, s'_3) \in \mathcal{R}$ such that $\Gamma_1 \vdash B_1 : s_1$ and $\Gamma_1, x : B_1 \vdash C_1 : s_2$ and $\Gamma_2 \vdash B_2 : s'_1$ and $\Gamma_2, x : B_2 \vdash C_2 : s'_2$ and $A_1 =_{\beta\varepsilon} s_3$ and $A_2 =_{\beta\varepsilon} s'_3$. Note that $B_1 \not\to_\varepsilon^* \varepsilon$ and $B_2 \not\to_\varepsilon^* \varepsilon$ and $C_1 \not\to_\varepsilon^* \varepsilon$ and $C_2 \not\to_\varepsilon^* \varepsilon$, by Lemma 96. Hence, by the inductive hypothesis and the confluence of $\beta\varepsilon$-reduction $s_1 = s'_1$ and $s_2 = s'_2$. Thus $s_3 = s'_3$ because the piPTS is functional. Hence $A_1 =_{\beta\varepsilon} A_2$.

- $M_1 = \lambda x : B_1.N_1$ and $M_2 = \lambda x : B_2.N_2$ and $B_1 =_\varepsilon B_2$ and $N_1 =_\varepsilon N_2$. By the generation lemma there exist $s_1, s_2 \in \mathcal{S}$ and C_1, C_2 such that $\Gamma_i \vdash B_i : s_i$ and $\Gamma_i, x : B_i \vdash N_i : C_i$ and $A_i =_{\beta\varepsilon} \Pi x : B_i.C_i$. Note that $N_i \not\to_\varepsilon^* \varepsilon$ because $M_i \not\to_\varepsilon^* \varepsilon$. Hence, by the inductive hypothesis $C_1 =_{\beta\varepsilon} C_2$. Hence $A_1 =_{\beta\varepsilon} A_2$, because also $B_1 =_\varepsilon B_2$.
- $M_1 = N_1 N_1'$ and $M_2 = N_2 N_2'$ and $N_1 =_\varepsilon N_2$ and $N_1' =_\varepsilon N_2'$. By the generation lemma there exist $x_1, x_2, B_1, B_2, C_1, C_2$ such that $\Gamma_i \vdash N_i : (\Pi x_i : B_i.C_i)$ and $\Gamma_i \vdash N_i' : B_i$ and $N_i' \sim x_i$ and $A_i =_{\beta\varepsilon} C_i[N_i'/x_i]$. Note that $N_i \not\to_\varepsilon^* \varepsilon$ because $M_i \not\to_\varepsilon^* \varepsilon$. Hence, by the inductive hypothesis $\Pi x_1 : B_1.C_1 =_{\beta\varepsilon} \Pi x_2 : B_2.C_2$. Thus $x_1 = x_2$ and $C_1 =_{\beta\varepsilon} C_2$ by confluence of $\beta\varepsilon$-reduction. Hence $C_1[N_1'/x_1] =_{\beta\varepsilon} C_2[N_2'/x_2]$ by Corollary 91. Therefore $A_1 =_{\beta\varepsilon} A_2$.

We have thus shown $(\star)$. Now assume $\Gamma \vdash M_i : A_i$ and $M_1 =_{\beta\varepsilon} M_2$ and $M_i \not\to_\varepsilon^* \varepsilon$. By confluence of $\beta\varepsilon$-reduction and by Lemma 88 there are N_1, N_2 with $M_i \to_\beta^* N_i$ and $N_1 =_\varepsilon N_2$. By the subject reduction theorem $\Gamma \vdash N_i : A_i$. Because $M_i \to_\beta^* N_i$ and $M_i \not\to_\varepsilon^* \varepsilon$, Lemma 86 implies that $N_i \not\to_\varepsilon^* \varepsilon$. Hence by $(\star)$ we obtain $A_1 =_{\beta\varepsilon} A_2$. ◀

▶ **Lemma 99.** *In a logical piPTS, if* $\Gamma \vdash B : s$ *and* $B =_{\beta\varepsilon} B'$ *and* $\Gamma \vdash A' : B'$ *then* $\Gamma \vdash B' : s$.

Proof. By the correctness of types lemma there are two cases.

- $B' = s'$. Then $B \to_\beta^* s'$ by confluence of $\beta\varepsilon$-reduction and Lemma 19. Hence $\Gamma \vdash s' : s$ by the subject reduction theorem, i.e., $\Gamma \vdash B' : s$.
- $\Gamma \vdash B' : s'$. Note that $B \not\to_\varepsilon^* \varepsilon$ and $B' \not\to_\varepsilon^* \varepsilon$ by Lemma 96. Hence, by the second point of the uniqueness of types lemma and by confluence of $\beta\varepsilon$-reduction $s = s'$. Therefore $\Gamma \vdash B' : s$.

◀

▶ **Lemma 100.** *In a logical piPTS, if* $\Gamma \vdash M : C : *^p$ *then* $M \to_\varepsilon^* \varepsilon$.

Proof. Induction on M. We have the following cases.

- $M = s$. By the generation lemma there is $s' \in \mathcal{S}$ such that $C =_{\beta\varepsilon} s'$. By confluence of $\beta\varepsilon$-reduction and Lemma 19 we have $C \to_\beta^* s'$. By the subject reduction theorem $\Gamma \vdash s' : *^p$. This is a contradiction, because the piPTS is logical.
- $M = x$. By the generation lemma there are $s \in \mathcal{S}$ and B such that $B =_{\beta\varepsilon} C$ and $\Gamma \vdash B : s$ and $(x : B) \in \Gamma$ and $x \in V^s$. By Lemma 99 we obtain $\Gamma \vdash C : s$, and thus $s = *^p$ by the uniqueness of types lemma. So $x \in V^{*^p}$. Hence $M = x \to_\varepsilon \varepsilon$.
- $M = \Pi x : A.B$. By the generation lemma there is $s' \in \mathcal{S}$ with $C =_{\beta\varepsilon} s'$. Like in the case $M = s$, using confluence of $\beta\varepsilon$-reduction, Lemma 19 and the subject reduction theorem, we derive a contradiction.
- $M = \lambda x : A.N$. By the generation lemma there are $s \in \mathcal{S}$ and B such that $\Gamma \vdash (\Pi x : A.B) : s$ and $\Gamma, x : A \vdash N : B$ and $C =_{\beta\varepsilon} \Pi x : A.B$. By Lemma 99 we have $\Gamma \vdash C : s$, and thus $s = *^p$ by the uniqueness of types lemma. Since $\Gamma \vdash (\Pi x : A.B) : *^p$, by the generation lemma there is $(s_1, s_2, *^p) \in \mathcal{R}$ such that $\Gamma, x : A \vdash B : s_2$. Because the piPTS is logical $s_2 = *^p$. Hence $\Gamma, x : A \vdash N : B : *^p$. By the inductive hypothesis $N \to_\varepsilon^* \varepsilon$. Hence $M = \lambda x : A.N \to_\varepsilon^* \lambda x : A.\varepsilon \to_\varepsilon \varepsilon$.
- $M = M_1 M_2$. By the generation lemma there are A, B such that $\Gamma \vdash M_1 : (\Pi x : A.B)$ and $\Gamma \vdash M_2 : A$ and $C =_{\beta\varepsilon} B[M_2/x]$ and $M_2 \sim x$. By the correctness of types lemma and the generation lemma there is $(s_1, s_2, s_3) \in \mathcal{R}$ such that $\Gamma \vdash (\Pi x : A.B) : s_3$ and $\Gamma, x : A \vdash B : s_2$. Since $\Gamma \vdash M_2 : A$ and $M_2 \sim x$, by the substitution lemma $\Gamma \vdash B[M_2/x] : s_2$. By Lemma 99 we have $\Gamma \vdash C : s_2$, and thus $s_2 = *^p$ by the uniqueness of types lemma. Hence $s_3 = *^p$ because the piPTS is logical. Thus $\Gamma \vdash M_1 : (\Pi x : A.B) : *^p$, and by the inductive hypothesis we conclude $M_1 \to_\varepsilon^* \varepsilon$. Therefore $M = M_1 M_2 \to^* \varepsilon M_2 \to_\varepsilon \varepsilon$.

◀

▶ **Lemma 101.** *In a logical piPTS, if* $\Gamma \vdash M : C$ *and* $M \to_\varepsilon^* \varepsilon$ *then* $\Gamma \vdash C : *^p$.

Proof. Induction on M. There are three possibilities.

- $M = x \in V^{*^p}$. By the generation lemma there exists B such that $B =_{\beta\varepsilon} C$ and $\Gamma \vdash B : *^p$. By Lemma 99 we have $\Gamma \vdash C : *^p$.
- $M = \lambda x : A.M'$ with $M' \to_\varepsilon^* \varepsilon$. By the generation lemma there exist $s \in \mathcal{S}$ and B such that $\Gamma \vdash (\Pi x : A.B) : s$ and $\Gamma, x : A \vdash M' : B$ and $C =_{\beta\varepsilon} \Pi x : A.B$. By the generation lemma there is $(s_1, s_2, s) \in \mathcal{R}$ with $\Gamma \vdash A : s_1$ and $\Gamma, x : A \vdash B : s_2$. Since $\Gamma, x : A \vdash M' : B$ and $M' \to_\varepsilon^* \varepsilon$, by the inductive hypothesis and the uniqueness of types lemma we obtain $s_2 = *^p$. Because the piPTS is logical also $s = *^p$. Then $\Gamma \vdash C : *^p$ by Lemma 99.
- $M = M_1 M_2$ with $M_1 \to_\varepsilon^* \varepsilon$. By the generation lemma there are A, B such that $\Gamma \vdash M_1 : (\Pi x : A.B)$ and $\Gamma \vdash M_2 : A$ and $C =_{\beta\varepsilon} B[M_2/x]$ and $M_2 \sim x$. By the inductive hypothesis $\Gamma \vdash M_1 : (\Pi x : A.B) : *^p$. By the generation lemma there is $(s_1, s_2, *^p) \in \mathcal{R}$ such that $\Gamma \vdash A : s_1$ and $\Gamma, x : A \vdash B : s_2$. Because the piPTS is logical, $s_2 = *^p$. By the substitution lemma we thus obtain $\Gamma \vdash B[M_2/x] : *^p$. By Lemma 99 we have $\Gamma \vdash C : *^p$.

◀

▶ **Theorem 30.** *Assume the piPTS is logical and* M *is a* Γ*-term. Then* M *is a* Γ*-proof if and only if* $M \to_\varepsilon^* \varepsilon$.

Proof. Follows from the correctness of types lemma, Lemma 96, Lemma 100 and Lemma 101.

◀

▶ **Lemma 31.** *In a logical piPTS, if* M *is a* Γ*-term and* $M =_{\beta\varepsilon} N$ *and* $\Gamma \vdash N : s$ *then* $\Gamma \vdash M : s$.

Proof. By the correctness of types lemma either $\Gamma \vdash M : s'$ or $M = s'$ for some $s' \in \mathcal{S}$. If $\Gamma \vdash M : s'$ then $M \not\to_\varepsilon^* \varepsilon$ and $N \not\to_\varepsilon^* \varepsilon$ by Lemma 96, so $s' = s$ by the uniqueness of types lemma and confluence of $\beta\varepsilon$-reduction. If $M = s'$ then $N \to_\beta^* M = s'$ by confluence of $\beta\varepsilon$-reduction and Lemma 19. Hence $\Gamma \vdash M : s$ by the subject reduction theorem. ◀

▶ **Lemma 102.** *In a logical piPTS, if* M *is a* Γ*-proof and* $\Gamma \vdash M : A$ *then* $\Gamma \vdash A : *^p$.

Proof. Since M is a Γ-proof, $M \to_\varepsilon^* \varepsilon$ by Theorem 30. Hence $\Gamma \vdash A : *^p$ by Lemma 101. ◀

▶ **Lemma 32.** *In a logical piPTS, if* $\Gamma \vdash M : A$ *and* $\Gamma, x : A$ *is a legal context then* $M \sim x$.

Proof. Since $\Gamma, x : A$ is a legal context, by the start lemma there is $s \in \mathcal{S}$ with $x \in V^s$ and $\Gamma \vdash A : s$. First, assume $s = *^p$. Since then $\Gamma \vdash M : A : *^p$, the term M is a Γ-proof, and thus $M \to_\varepsilon^* \varepsilon$ by Theorem 30. So $M \sim x \in V^{*^p}$. If $s \neq *^p$ then M is not a Γ-proof, by Lemma 102 and the uniqueness of types lemma. Hence, then also $M \sim x \in V^s$. ◀

▶ **Lemma 34.** *In a logical piPTS,* $\Gamma \vdash^- M : N$ *is equivalent to* $\Gamma \vdash M : N$.

Proof. The implication from right to left follows by induction on the length of the derivation of $\Gamma \vdash M : N$. For the other direction we proceed by induction on the length of the derivation of $\Gamma \vdash^- M : N$. Lemma 32 is needed to handle the application rule. ◀

B Proofs for Section 5

▶ **Lemma 62.**
1. *If* $M \succ^{\mathcal{F}}_{\Gamma} \varphi$ *and* $\Gamma' \supseteq \Gamma$ *is a legal context then* $M \succ^{\mathcal{F}}_{\Gamma'} \varphi$.
2. *If* $M \succ^{\mathcal{C}}_{\Gamma} t$ *and* $\Gamma' \supseteq \Gamma$ *is a legal context then* $M \succ^{\mathcal{C}}_{\Gamma'} t$.

Proof. Induction on the definition of $M \succ^{\mathcal{F}}_{\Gamma} \varphi$ and $M \succ^{\mathcal{C}}_{\Gamma} t$. We show a few cases. The other cases are similar, trivial, or follow directly from the inductive hypothesis.

- $M = \Pi x : A.B \succ^{\mathcal{F}}_{\Gamma} \varphi_1 \to \varphi_2 = \varphi$. Then $\Gamma \vdash A : *^p$ and $A \succ^{\mathcal{F}}_{\Gamma} \varphi_1$ and $B \succ^{\mathcal{F}}_{\Gamma,x:A} \varphi_2$. By Corollary 28 we have $x \in V^{*^p}$. By the variable convention we may assume $x \notin \mathrm{dom}(\Gamma')$. By the thinning lemma $\Gamma' \vdash A : *^p$. Hence $\Gamma', x : A \supseteq \Gamma, x : A$ is a legal context. So $B \succ^{\mathcal{F}}_{\Gamma',x:A} \varphi_2$ by the inductive hypothesis. Also $A \succ^{\mathcal{F}}_{\Gamma'} \varphi_1$ by the inductive hypothesis. Thus $\Pi x : A.B \succ^{\mathcal{F}}_{\Gamma'} \varphi_1 \to \varphi_2$.
- $M = \Pi x : A.B \succ^{\mathcal{F}}_{\Gamma} \forall x.T(x,t) \to \psi = \varphi$. Then $\Gamma \nvdash A : *^p$ and $A \succ^{\mathcal{C}}_{\Gamma} t$ and $B \succ^{\mathcal{F}}_{\Gamma,x:A} \psi$. By Corollary 28 we have $x \in V^s$. By the variable convention we may assume $x \notin \mathrm{dom}(\Gamma')$. Since $\Pi x : A.B$ is a Γ-subject, by the correctness of types and the generation lemmas $\Gamma \vdash A : s$ for some $s \in \mathcal{S}$. So $\Gamma' \vdash A : s$ by the thinning lemma. Hence $\Gamma', x : A \supseteq \Gamma, x : A$ is a legal context. So $B \succ^{\mathcal{F}}_{\Gamma',x:A} \psi$ by the inductive hypothesis. Also $A \succ^{\mathcal{C}}_{\Gamma'} t$ by the inductive hypothesis. We also have $\Gamma' \nvdash A : *^p$, because otherwise $s = *^p$ by the uniqueness of types lemma. Thus $\Pi x : A.B \succ^{\mathcal{F}}_{\Gamma'} \forall x.T(x,t) \to \psi$.
- $M \succ^{\mathcal{C}}_{\Gamma} \varepsilon$ and M is a Γ-proof. Then M is a Γ'-proof by the thinning lemma, so $M \succ^{\mathcal{C}}_{\Gamma'} \varepsilon$.
- $M = M_1 M_2 \succ^{\mathcal{C}}_{\Gamma} t_1 t_2 = t$ and $M_1 \succ^{\mathcal{C}}_{\Gamma} t_1$ and $M_2 \succ^{\mathcal{C}}_{\Gamma} t_2$. We have $M_i \succ^{\mathcal{C}}_{\Gamma'} t_i$ by the inductive hypothesis. Note that M is not a Γ'-proof, because otherwise $M \to^*_{\varepsilon} \varepsilon$ by Theorem 30 and thus M would also be a Γ-proof. Hence $M_1 M_2 \succ^{\mathcal{C}}_{\Gamma'} t_1 t_2$.
- $M = (\lambda x : A.M')[\vec{N}/\vec{x}] \succ^{\mathcal{C}}_{\Gamma} (f\vec{y})[\vec{t}/\vec{x}] = t$ and $\Gamma_0 \vdash (\lambda x : A.M') : B$ and $\Gamma_0 \nvdash A : *^p$ and $f = \Lambda_1(x,r,t)$ and $\vec{y} = \mathrm{FV}(r,t) \setminus \{x\}$ and $\Gamma_0 \rightsquigarrow_{\vec{x},\vec{N},\vec{t}} \Gamma$ and $A[\vec{N}/\vec{x}] \succ^{\mathcal{C}}_{\Gamma} r[\vec{t}/\vec{x}]$ and $M'[\vec{N}/\vec{x}] \succ^{\mathcal{C}}_{\Gamma,x:A[\vec{N}/\vec{x}]} t[\vec{t}/\vec{x}]$. Then also $\Gamma_0 \rightsquigarrow_{\vec{x},\vec{N},\vec{t}} \Gamma'$ by the definition of $\rightsquigarrow$. By the variable convention we may assume $x \notin \mathrm{dom}(\Gamma')$, so $\Gamma', x : A[\vec{N}/\vec{x}] \supseteq \Gamma, x : A[\vec{N}/\vec{x}]$ is a legal context. Thus $A[\vec{N}/\vec{x}] \succ^{\mathcal{C}}_{\Gamma'} r[\vec{t}/\vec{x}]$ and $M'[\vec{N}/\vec{x}] \succ^{\mathcal{C}}_{\Gamma',x:A[\vec{N}/\vec{x}]} t[\vec{t}/\vec{x}]$ by the inductive hypothesis. Additionally, like in the case $M = M_1 M_2$, using Theorem 30 we conclude that M is not a Γ'-proof. Hence $M = (\lambda x : A.M')[\vec{N}/\vec{x}] \succ^{\mathcal{C}}_{\Gamma'} (f\vec{y})[\vec{t}/\vec{x}] = t$.

◀

▶ **Lemma 64.** *Assume* $N \sim x$. *Then* $M \to^*_{\varepsilon} \varepsilon$ *iff* $M[N/x] \to^*_{\varepsilon} \varepsilon$.

Proof. By Lemma 16 we have $\mathrm{nf}_{\varepsilon}(M[N/x]) = \mathrm{nf}_{\varepsilon}(M)[\mathrm{nf}_{\varepsilon}(N)/x]$. Thus if $\mathrm{nf}_{\varepsilon}(M) = \varepsilon$ then also $\mathrm{nf}_{\varepsilon}(M[N/x]) = \varepsilon$. Conversely, if $\mathrm{nf}_{\varepsilon}(M[N/x]) = \varepsilon$ and $\mathrm{nf}_{\varepsilon}(M) \neq \varepsilon$ then $\mathrm{nf}_{\varepsilon}(M) = x$ and $\mathrm{nf}_{\varepsilon}(N) = \varepsilon$. Then also $x \in V^{*^p}$, because $N \sim x$. This is impossible because then $x \to_{\varepsilon} \varepsilon$. ◀

▶ **Lemma 65.** *Assume* $\Gamma_1 \vdash N : A$ *and* $N \succ^{\mathcal{C}}_{\Gamma_1} t$ *and* $N \sim y$.
1. *If* $M \succ^{\mathcal{F}}_{\Gamma_1,y:A,\Gamma_2} \varphi$ *then* $M[N/y] \succ^{\mathcal{F}}_{\Gamma_1,\Gamma_2[N/y]} \varphi[t/y]$.
2. *If* $M \succ^{\mathcal{C}}_{\Gamma_1,y:A,\Gamma_2} u$ *then* $M[N/y] \succ^{\mathcal{C}}_{\Gamma_1,\Gamma_2[N/y]} u[t/y]$.

Proof. By induction on the definition of $M \succ^{\mathcal{F}}_{\Gamma_1,y:A,\Gamma_2} \varphi$ and $M \succ^{\mathcal{C}}_{\Gamma_1,y:A,\Gamma_2} u$. Again, we show a few cases.

Let $\Gamma = \Gamma_1, y : A, \Gamma_2$ and $\Gamma' = \Gamma_1, \Gamma_2[N/y]$. Note that because we implicitly assume M is a Γ-subject (Γ-proposition), by the substitution lemma $M[N/y]$ is also a Γ'-subject (Γ'-proposition). Also note that M is a Γ-proof iff $M[N/y]$ is a Γ'-proof. Indeed, this follows from Lemma 64 and Theorem 30.

- $M = \Pi x : C.B \succ^{\mathcal{F}}_{\Gamma} \varphi_1 \to \varphi_2 = \varphi$. Then $\Gamma \vdash C : *^p$ and $C \succ^{\mathcal{F}}_{\Gamma} \varphi_1$ and $B \succ^{\mathcal{F}}_{\Gamma,x:C} \varphi_2$. By the substitution lemma $\Gamma' \vdash C[N/y] : *^p$. By the inductive hypothesis $C[N/y] \succ^{\mathcal{F}}_{\Gamma'} \varphi_1[t/y]$ and $B[N/y] \succ^{\mathcal{F}}_{\Gamma',x:C[N/y]} \varphi_2[t/y]$. Hence $(\Pi x : C.B)[N/y] = \Pi x : C[N/y].B[N/y] \succ^{\mathcal{F}}_{\Gamma'} \varphi_1[t/y] \to \varphi_2[t/y] = \varphi[t/y]$.
- M is a Γ-proof and $M \succ^{\mathcal{C}}_{\Gamma} \varepsilon$. Then by the substitution lemma $M[N/y]$ is also a Γ'-proof. Hence $M[N/y] \succ^{\mathcal{C}}_{\Gamma'} \varepsilon$.
- $M = y \succ^{\mathcal{C}}_{\Gamma} y = u$. By the substitution lemma y is a Γ'-subject, so Γ' is a legal context. We also have $N \succ^{\mathcal{C}}_{\Gamma_1} t$ and $\Gamma_1 \subseteq \Gamma'$. Hence $M[N/y] = N \succ^{\mathcal{C}}_{\Gamma'} t = u[t/y]$ by Lemma 62.
- $M = M_1 M_2 \succ^{\mathcal{C}}_{\Gamma} u_1 u_2$ and $M_1 \succ^{\mathcal{C}}_{\Gamma} u_1$ and $M_2 \succ^{\mathcal{C}}_{\Gamma} u_2$. By the inductive hypothesis $M_i[N/y] \succ^{\mathcal{C}}_{\Gamma'} u_i[t/y]$. Recall that $M[N/y]$ is not a Γ'-proof, by the discussion in the second paragraph of the proof of this lemma. Therefore $M[N/y] = M_1[N/y]M_2[N/y] \succ^{\mathcal{C}}_{\Gamma'} u_1[t/y]u_2[t/y] = u[t/y]$.
- $M = (\lambda x : A.M')[\vec{N}/\vec{x}] \succ^{\mathcal{C}}_{\Gamma} (f\vec{y})[\vec{t}/\vec{x}] = u$ and $\Gamma_0 \vdash (\lambda x : A.M') : B$ and $\Gamma_0 \nvdash A : *^p$ and $f = \Lambda_1(x, r_1, r_2)$ and $\vec{y} = \mathrm{FV}(r_1, r_2) \setminus \{x\}$ and $\Gamma_0 \rightsquigarrow_{\vec{x},\vec{N},\vec{t}} \Gamma$ and $A[\vec{N}/\vec{x}] \succ^{\mathcal{C}}_{\Gamma} r_1[\vec{t}/\vec{x}]$ and $M'[\vec{N}/\vec{x}] \succ^{\mathcal{C}}_{\Gamma,x:A[\vec{N}/\vec{x}]} r_2[\vec{t}/\vec{x}]$. Then by definition also $\Gamma_0 \rightsquigarrow_{\vec{x},y,\vec{N},N,\vec{t},t} \Gamma'$. Moreover, we obtain $A[\vec{N}/\vec{x}][N/y] \succ^{\mathcal{C}}_{\Gamma'} r_1[\vec{t}/\vec{x}][t/y]$ and $M'[\vec{N}/\vec{x}][N/y] \succ^{\mathcal{C}}_{\Gamma',x:A[\vec{N}/\vec{x}][N/y]} r_2[\vec{t}/\vec{x}][t/y]$ by the inductive hypothesis. Hence, recalling that $M[N/y]$ is not a Γ'-proof,

$$M[N/y] = (\lambda x : A.M')[\vec{N}/\vec{x}][N/y] \succ^{\mathcal{C}}_{\Gamma'} (f\vec{y})[\vec{t}/\vec{x}][t/y] = u[t/y].$$

◀

▶ **Lemma 67.** *Assume $y \in V^{*^p}$.*

1. *If $M \succ^{\mathcal{F}}_{\Gamma} \varphi$ then $y \notin \mathrm{FV}(\varphi)$.*
2. *If $M \succ^{\mathcal{C}}_{\Gamma} t$ then $y \notin \mathrm{FV}(t)$.*

Proof. Induction on the definition of $M \succ^{\mathcal{F}}_{\Gamma} \varphi$ and $M \succ^{\mathcal{C}}_{\Gamma} t$. Note that if y is a Γ-subject then y is a Γ-proof, by the generation and start lemmas. Hence, the case $M = y \succ^{\mathcal{C}}_{\Gamma} y = t$ is impossible. ◀

▶ **Lemma 68.**

1. *If $M \succ^{\mathcal{F}}_{\Gamma} \varphi$ then $\mathrm{FV}(\varphi) = \mathrm{FV}(\mathrm{nf}_{\varepsilon}(M))$.*
2. *If $M \succ^{\mathcal{C}}_{\Gamma} t$ then $\mathrm{FV}(t) = \mathrm{FV}(\mathrm{nf}_{\varepsilon}(M))$.*

Proof. Induction on the definition of $M \succ^{\mathcal{F}}_{\Gamma} \varphi$ and $M \succ^{\mathcal{C}}_{\Gamma} t$, using Lemma 67. We show a few cases. The other cases are similar, trivial, or follow directly from the inductive hypothesis.

- $M = \Pi x : A.B \succ^{\mathcal{F}}_{\Gamma} \varphi_1 \to \varphi_2 = \varphi$. Then $\Gamma \vdash A : *^p$ and $A \succ^{\mathcal{F}}_{\Gamma} \varphi_1$ and $B \succ^{\mathcal{F}}_{\Gamma,x:A} \varphi_2$. By the inductive hypothesis $\mathrm{FV}(\mathrm{nf}_{\varepsilon}(A)) = \mathrm{FV}(\varphi_1)$ and $\mathrm{FV}(\mathrm{nf}_{\varepsilon}(B)) = \mathrm{FV}(\varphi_2)$. Note that $\mathrm{nf}_{\varepsilon}(M) = \Pi x : \mathrm{nf}_{\varepsilon}(A).\mathrm{nf}_{\varepsilon}(B)$, so $\mathrm{FV}(\mathrm{nf}_{\varepsilon}(M)) = \mathrm{FV}(\mathrm{nf}_{\varepsilon}(A)) \cup (\mathrm{FV}(\mathrm{nf}_{\varepsilon}(B)) \setminus \{x\})$. Since $\Gamma \vdash A : *^p$, we have $x \in V^{*^p}$ by Corollary 28. Thus $x \notin \mathrm{FV}(\varphi_2)$ by Lemma 67, so $\mathrm{FV}(\varphi_2) = \mathrm{FV}(\mathrm{nf}_{\varepsilon}(B)) \setminus \{x\}$. Hence $\mathrm{FV}(\mathrm{nf}_{\varepsilon}(M)) = \mathrm{FV}(\mathrm{nf}_{\varepsilon}(A)) \cup (\mathrm{FV}(\mathrm{nf}_{\varepsilon}(B)) \setminus \{x\}) = \mathrm{FV}(\varphi_1) \cup \mathrm{FV}(\varphi_2) = \mathrm{FV}(\varphi)$.
- $M = \Pi x : A.B \succ^{\mathcal{F}}_{\Gamma} \forall x.T(x,t) \to \psi = \varphi$. Then $\Gamma \vdash A : *^p$ and $A \succ^{\mathcal{C}}_{\Gamma} t$ and $B \succ^{\mathcal{F}}_{\Gamma,x:A} \psi$. By the inductive hypothesis $\mathrm{FV}(\mathrm{nf}_{\varepsilon}(A)) = \mathrm{FV}(t)$ and $\mathrm{FV}(\mathrm{nf}_{\varepsilon}(B)) = \mathrm{FV}(\psi)$. Note that $\mathrm{nf}_{\varepsilon}(M) = \Pi x : \mathrm{nf}_{\varepsilon}(A).\mathrm{nf}_{\varepsilon}(B)$, so $\mathrm{FV}(\mathrm{nf}_{\varepsilon}(M)) = \mathrm{FV}(\mathrm{nf}_{\varepsilon}(A)) \cup (\mathrm{FV}(\mathrm{nf}_{\varepsilon}(B)) \setminus \{x\}) = (\mathrm{FV}(\mathrm{nf}_{\varepsilon}(A)) \cup \mathrm{FV}(\mathrm{nf}_{\varepsilon}(B))) \setminus \{x\}$ (by the variable convention we may assume $x \notin \mathrm{FV}(A)$). Thus $\mathrm{FV}(\varphi) = (\mathrm{FV}(t) \cup \mathrm{FV}(\psi)) \setminus \{x\} = \mathrm{FV}(\mathrm{nf}_{\varepsilon}(M))$.
- If M is a Γ-proof and $M \succ^{\mathcal{C}}_{\Gamma} \varepsilon = t$, then $\mathrm{nf}_{\varepsilon}(M) = \varepsilon$ by Theorem 30, so $\mathrm{FV}(\mathrm{nf}_{\varepsilon}(M)) = \mathrm{FV}(t)$.

- $M = x \succ^{\mathcal{C}}_{\Gamma} x = t$. Then M is not a Γ-proof, so $\mathrm{nf}_{\varepsilon}(x) = x$ by Theorem 30. Hence $\mathrm{FV}(\mathrm{nf}_{\varepsilon}(M)) = \mathrm{FV}(t)$.
- $M = M_1 M_2 \succ^{\mathcal{C}}_{\Gamma} t_1 t_2 = t$ and $M_1 \succ^{\mathcal{C}}_{\Gamma} t_1$ and $M_2 \succ^{\mathcal{C}}_{\Gamma} t_2$. In this case M is not a Γ-proof, so $M \not\to^*_{\varepsilon} \varepsilon$. Hence $\mathrm{nf}_{\varepsilon}(M) = \mathrm{nf}_{\varepsilon}(M_1)\mathrm{nf}_{\varepsilon}(M_2)$. Thus $\mathrm{FV}(\mathrm{nf}_{\varepsilon}(M)) = \mathrm{FV}(\mathrm{nf}_{\varepsilon}(M_1), \mathrm{nf}_{\varepsilon}(M_2)) = \mathrm{FV}(t_1, t_2) = \mathrm{FV}(t)$, using the inductive hypothesis.
- $M = (\lambda x : A.M')[\vec{N}/\vec{x}] \succ^{\mathcal{C}}_{\Gamma} (f\vec{y})[\vec{t}/\vec{x}]$ and $\Gamma' \vdash (\lambda x : A.M') : B$ and $\Gamma' \not\vdash A : *^p$ and $f = \Lambda_1(x, r, t)$ and $\vec{y} = \mathrm{FV}(r, t) \setminus \{x\}$ and $\Gamma' \rightsquigarrow_{\vec{x},\vec{N},\vec{t}} \Gamma$ and $A[\vec{N}/\vec{x}] \succ^{\mathcal{C}}_{\Gamma} r[\vec{t}/\vec{x}]$ and $M'[\vec{N}/\vec{x}] \succ^{\mathcal{C}}_{\Gamma, x:A[\vec{N}/\vec{x}]} t[\vec{t}/\vec{x}]$. By the inductive hypothesis $\mathrm{FV}(\mathrm{nf}_{\varepsilon}(A[\vec{N}/\vec{x}])) = \mathrm{FV}(r[\vec{t}/\vec{x}])$ and $\mathrm{FV}(\mathrm{nf}_{\varepsilon}(M'[\vec{N}/\vec{x}])) = \mathrm{FV}(t[\vec{t}/\vec{x}])$. By the variable convention we may assume $x \notin \mathrm{FV}(A, N_1, \ldots, N_n)$, so

$$\begin{array}{rcl} \mathrm{FV}(\mathrm{nf}_{\varepsilon}(M)) & = & \mathrm{FV}(\mathrm{nf}_{\varepsilon}(\lambda x : A[\vec{N}/\vec{x}].M'[\vec{N}/\vec{x}])) \\ & = & \mathrm{FV}(r[\vec{t}/\vec{x}], t[\vec{t}/\vec{x}]) \setminus \{x\}. \end{array}$$

Let $\{x_{i_1}, \ldots, x_{i_k}\} = \mathrm{FV}(r, t) \cap \{x_1, \ldots, x_n\}$ and let $t'_i = t_i[t_{i+1}/x_{i+1}] \ldots [t_n/x_n]$ for $i = 1, \ldots, n$. We then have $r[\vec{t}/\vec{x}] = r[t'_{i_1}/x_{i_1}, \ldots, t'_{i_k}/x_{i_k}]$ and $t[\vec{t}/\vec{x}] = t[t'_{i_1}/x_{i_1}, \ldots, t'_{i_k}/x_{i_k}]$. Thus

$$\mathrm{FV}(r[\vec{t}/\vec{x}], t[\vec{t}/\vec{x}]) = (\mathrm{FV}(r, t) \setminus \{x_{i_1}, \ldots, x_{i_k}\}) \cup \mathrm{FV}(t'_{i_1}, \ldots, t'_{i_k}).$$

Hence

$$\mathrm{FV}(\mathrm{nf}_{\varepsilon}(M)) = ((\mathrm{FV}(r, t) \setminus \{x_{i_1}, \ldots, x_{i_k}\}) \cup \mathrm{FV}(t'_{i_1}, \ldots, t'_{i_k})) \setminus \{x\}.$$

On the other hand, also $t = (f\vec{y})[\vec{t}/\vec{x}] = (f\vec{y})[t'_{i_1}/x_{i_1}, \ldots, t'_{i_k}/x_{i_k}]$ and $\vec{y} = \mathrm{FV}(r, t) \setminus \{x\}$. By the inductive hypothesis $\mathrm{FV}(t_i) = \mathrm{FV}(\mathrm{nf}_{\varepsilon}(N_i))$, so $x \notin \mathrm{FV}(t_i)$. Hence also $x \notin \mathrm{FV}(t'_i)$. Therefore

$$\begin{array}{rcl} \mathrm{FV}(t) & = & (\mathrm{FV}(r, t) \setminus \{x, x_{i_1}, \ldots, x_{i_k}\}) \cup \mathrm{FV}(t'_{i_1}, \ldots, t'_{i_k}) \\ & = & ((\mathrm{FV}(r, t) \setminus \{x_{i_1}, \ldots, x_{i_k}\}) \cup \mathrm{FV}(t'_{i_1}, \ldots, t'_{i_k})) \setminus \{x\} \\ & = & \mathrm{FV}(\mathrm{nf}_{\varepsilon}(M)). \end{array}$$

◀

▶ **Lemma 69.** *Assume $\Gamma =_{\varepsilon} \Gamma'$.*

1. *If $M \succ^{\mathcal{F}}_{\Gamma} \varphi$ and $M' \succ^{\mathcal{F}}_{\Gamma'} \varphi$ then $M =_{\varepsilon} M'$.*
2. *If $M \succ^{\mathcal{C}}_{\Gamma} t$ and $M' \succ^{\mathcal{C}}_{\Gamma'} t$ then $M =_{\varepsilon} M'$.*

Proof. Induction on the definition of $M \succ^{\mathcal{F}}_{\Gamma} \varphi$ and $M \succ^{\mathcal{C}}_{\Gamma} t$. We show a few cases. The other cases are similar, trivial, or follow directly from the inductive hypothesis.

- $M = \Pi x : A.B$ and $M' = \Pi x : A'.B'$ and $\varphi = \varphi_1 \to \varphi_2$. Then $A \succ^{\mathcal{F}}_{\Gamma} \varphi_1$ and $A' \succ^{\mathcal{F}}_{\Gamma} \varphi_1$ and $B \succ^{\mathcal{F}}_{\Gamma, x:A} \varphi_2$ and $B' \succ^{\mathcal{F}}_{\Gamma', x:A'} \varphi_2$. By the inductive hypothesis $A =_{\varepsilon} A'$. Hence $\Gamma, x : A =_{\varepsilon} \Gamma', x : A'$, so $B =_{\varepsilon} B'$ by the inductive hypothesis. Therefore $M =_{\varepsilon} M'$.
- $t = \varepsilon$ and M is a Γ-proof and M' is a Γ'-proof. By Theorem 30 we have $M \to^*_{\varepsilon} \varepsilon$ and $M' \to^*_{\varepsilon} \varepsilon$, so $M =_{\varepsilon} M'$.
- $t = t_1 t_2$ and $M = M_1 M_2$ and $M' = M'_1 M'_2$ and $M_i \succ^{\mathcal{C}}_{\Gamma} t_i$ and $M'_i \succ^{\mathcal{C}}_{\Gamma'} t_i$. By the inductive hypothesis $M_i =_{\varepsilon} M'_i$. Hence $M =_{\varepsilon} M'$.
- $t = (f\vec{y})[\vec{t}/\vec{x}] = (f\vec{y'})[\vec{t'}/\vec{x'}]$ and $f = \Lambda_1(x, r, u) = \Lambda_1(x', r', u')$ and $\vec{y} = \mathrm{FV}(r, u) \setminus \{x\}$ and $\vec{y'} = \mathrm{FV}(r', u') \setminus \{x'\}$ and $M = (\lambda x : A.B)[\vec{N}/\vec{x}]$ and $M' = (\lambda x : A'.B')[\vec{N'}/\vec{x'}]$ and there is a bijection $\sigma : V \to V$ such that $\sigma(y'_i) = y_i$ and $\sigma(x') = x$ and $\sigma(r') = r$ and $\sigma(u') = u$. We also have $A[\vec{N}/\vec{x}] \succ^{\mathcal{C}}_{\Gamma} r[\vec{t}/\vec{x}]$ and $B[\vec{N}/\vec{x}] \succ^{\mathcal{C}}_{\Gamma, x:A[\vec{N}/\vec{x}]} u[\vec{t}/\vec{x}]$ and $A'[\vec{N'}/\vec{x'}] \succ^{\mathcal{C}}_{\Gamma'}$

$r'[\vec{t'}/\vec{x'}]$ and $B'[\vec{N'}/\vec{x'}] \succ^{\mathcal{C}}_{\Gamma',x:A'[\vec{N'}/\vec{x'}]} u'[\vec{t'}/\vec{x'}]$. Let $p_i = t_i[t_{i+1}/x_{i+1}]\dots[t_n/x_n]$ and $p'_i = t'_i[t'_{i+1}/x'_{i+1}]\dots[t'_m/x'_m]$. We have

$$(f\vec{y})[p_1/x_1,\dots,p_n/x_n] = (f\vec{y'})[p'_1/x'_1,\dots,p'_m/x'_m].$$

Without loss of generality we may assume that there is $k \leq n$ such that $x_i = y_i$ and $x'_i = y'_i$ and $p_i = p'_i$ for $i \leq k$ (this may be always achieved by taking the "missing" p_is (p'_is) to be equal to y_is (y'_is)), and $x_i \notin \vec{y} = \mathrm{FV}(r,u)\setminus\{x\}$ for $i > k$. We may also assume there is $k \leq k' \leq m$ such that $x'_i = y'_i$ for $k < i \leq k'$ and $x'_i \notin \vec{y'} = \mathrm{FV}(r',u')\setminus\{x'\}$ for $i > k'$. Then $p'_i = y_i$ for $k < i \leq k'$. Hence, in fact we may assume $k = k'$, by taking the missing p_is equal to y_is.
By the variable convention we may also assume $x, x' \notin \{x_1,\dots,x_n,x'_1,\dots,x'_m\}$. Hence for $i > k$ we have $x_i \notin \mathrm{FV}(r,u)$ and $x'_i \notin \mathrm{FV}(r',u')$. Recall that $\sigma^{-1}(y_i) = y'_i$. Since also $p_i = p'_i$ and $x_i = \sigma(x'_i)$ for $i \leq k$:

$$\begin{aligned} r[\vec{t}/\vec{x}] &= r[p_1/x_1,\dots,p_k/x_k] \\ &= r[p'_1/\sigma(x'_1),\dots,p'_k/\sigma(x'_k)] \\ &= \sigma^{-1}(r)[p'_1/x'_1,\dots,p'_k/x'_k] \\ &= r'[p'_1/x'_1,\dots,p'_k/x'_k] \\ &= r'[\vec{t'}/\vec{x'}] \end{aligned}$$

and

$$\begin{aligned} u[\vec{t}/\vec{x}] &= u[p_1/x_1,\dots,p_k/x_k] \\ &= u[p'_1/\sigma(x'_1),\dots,p'_k/\sigma(x'_k)] \\ &= \sigma^{-1}(u)[p'_1/x'_1,\dots,p'_k/x'_k] \\ &= u'[p'_1/x'_1,\dots,p'_k/x'_k] \\ &= u'[\vec{t'}/\vec{x'}]. \end{aligned}$$

So by the inductive hypothesis $A[\vec{N}/\vec{x}] =_\varepsilon A'[\vec{N'}/\vec{x'}]$ and thus also $\Gamma, x : A[\vec{N}/\vec{x}] =_\varepsilon \Gamma', x : A'[\vec{N'}/\vec{x'}]$, so $B[\vec{N}/\vec{x}] =_\varepsilon B'[\vec{N'}/\vec{x'}]$ by applying the inductive hypothesis again. This implies that $M =_\varepsilon M'$.

◀

▶ **Lemma 74.** *Suppose $\Gamma \succ \Delta$ and $\Delta_{\mathrm{Ax}}, \Delta \vdash XQ_1\dots Q_m : \psi$, where each Q_i is either an individual term or a reconstructible proof term. Let $\Gamma_0 = x_1 : A_1,\dots,x_n : A_n$ be such that $m = \mathrm{len}_{\mathbb{A}}(\Gamma_0)$ and Γ, Γ_0 is a legal context. If $(X : \gamma) \in \Delta_{\mathrm{Ax}}, \Delta$ with $\varphi \succ^{\mathbb{A}}_{\Gamma;\Gamma_0} \gamma$, then there exist $N_1,\dots,N_n$ and $u_1,\dots,u_n$ such that $\psi = \varphi[\vec{u}/\vec{x}]$ and $\Gamma, \Gamma_0 \rightsquigarrow_{\vec{x},\vec{N},\vec{u}} \Gamma$.*

Proof. Induction on n. If $n = 0$ then $m = 0$ and $\psi = \varphi$, so we are done. Thus suppose $\Gamma_0 = \Gamma'_0, x_{n+1} : A_{n+1}$.

First assume $\Gamma, \Gamma'_0 \vdash A_{n+1} : s$ and $s \neq *^p$. Then $A_{n+1} \succ^{\mathcal{C}}_{\Gamma,\Gamma'_0} t$ and $\forall x_{n+1}.T(x_{n+1},t) \to \varphi \succ^{\mathbb{A}}_{\Gamma;\Gamma'_0} \gamma$ and $\mathrm{len}_{\mathbb{A}}(\Gamma'_0) = \mathrm{len}_{\mathbb{A}}(\Gamma_0) - 2$. Also

$$\Delta_{\mathrm{Ax}}, \Delta \vdash XQ_1\dots Q_{m-2} : \forall x_{n+1}.T(x_{n+1},r) \to \psi'$$

and $Q_{m-1} = u_{n+1}$ is an individual term and $Q_m = D$ is a reconstructible proof term such that $\Delta_{\mathrm{Ax}}, \Delta \vdash D : T(u_{n+1},r)$ and $\psi = \psi'[u_{n+1}/x_{n+1}]$. By the inductive hypothesis there exist $N_1,\dots,N_n$ and $u_1,\dots,u_n$ such that $r = t[\vec{u}/\vec{x}]$ and $\psi' = \varphi[\vec{u}/\vec{x}]$ and $\Gamma, \Gamma'_0 \rightsquigarrow_{\vec{x},\vec{N},\vec{u}} \Gamma$. Then $\Gamma, \Gamma_0 \rightsquigarrow_{\vec{x},\vec{N},\vec{u}} \Gamma, x_{n+1} : A_{n+1}[\vec{N}/\vec{x}]$ by Lemma 60. Also $A_{n+1}[\vec{N}/\vec{x}] \succ^{\mathcal{C}}_{\Gamma} r$ by Corollary 66. Since $\Gamma, \Gamma'_0 \vdash A_{n+1} : s$, we have $\Gamma \vdash A_{n+1}[\vec{N}/\vec{x}] : s$ by Lemma 61. Because we also have

$\Delta_{\mathrm{Ax}}, \Delta \vdash D : T(u_{n+1}, r)$ and D is reconstructible, by 2 in Definition 73 there is N_{n+1} with $N_{n+1} \succ^{\mathcal{C}}_{\Gamma} u_{n+1}$ and $\Gamma \vdash N_{n+1} : A_{n+1}[\vec{N}/\vec{x}]$. Also $N_{n+1} \sim x_{n+1}$ by Lemma 32. Thus $\Gamma, \Gamma_0 \rightsquigarrow_{\vec{x}, x_{n+1}, \vec{N}, N_{n+1}, \vec{u}, u_{n+1}} \Gamma$ by definition of $\rightsquigarrow$. Moreover, $\psi = \psi'[u_{n+1}/x_{n+1}] = \varphi[u_1/x_1] \dots [u_{n+1}/x_{n+1}]$.

Now assume $\Gamma, \Gamma'_0 \vdash A_{n+1} : *^p$. Then $A_{n+1} \succ^{\mathcal{F}}_{\Gamma, \Gamma'_0} \varphi'$ and $\varphi' \to \varphi \succ^{\mathbb{A}}_{\Gamma; \Gamma'_0} \gamma$ and $\mathrm{len}_{\mathbb{A}}(\Gamma'_0) = \mathrm{len}_{\mathbb{A}}(\Gamma_0) - 1$. Also

$$\Delta_{\mathrm{Ax}}, \Delta \vdash X Q_1 \dots Q_{m-1} : \alpha \to \psi$$

and $Q_m = D$ is a reconstructible proof term such that $\Delta_{\mathrm{Ax}}, \Delta \vdash D : \alpha$. By the inductive hypothesis there are $N_1, \dots, N_n$ and $u_1, \dots, u_n$ such that $\alpha = \varphi'[\vec{u}/\vec{x}]$, $\psi = \varphi[\vec{u}/\vec{x}]$ and $\Gamma, \Gamma'_0 \rightsquigarrow_{\vec{x}, \vec{N}, \vec{u}} \Gamma$. Then $\Gamma, \Gamma_0 \rightsquigarrow_{\vec{x}, \vec{N}, \vec{u}} \Gamma, x_{n+1} : A_{n+1}[\vec{N}/\vec{x}]$ by Lemma 60. Also $A_{n+1}[\vec{N}/\vec{x}] \succ^{\mathcal{F}}_{\Gamma} \varphi'[\vec{u}/\vec{x}] = \alpha$ by Corollary 66. Since $\Gamma, \Gamma'_0 \vdash A_{n+1} : *^p$, we have $\Gamma \vdash A_{n+1}[\vec{N}/\vec{x}] : *^p$ by Lemma 61. Because we also have $\Delta_{\mathrm{Ax}}, \Delta \vdash D : \alpha$ and D is reconstructible, by 1 in Definition 73 there is N_{n+1} with $\Gamma \vdash N_{n+1} : A_{n+1}[\vec{N}/\vec{x}]$. Because N_{n+1} is a Γ-proof, we have $N_{n+1} \succ^{\mathcal{C}}_{\Gamma} \varepsilon$. Also $N_{n+1} \sim x_{n+1}$ by Lemma 32. Thus $\Gamma, \Gamma_0 \rightsquigarrow_{\vec{x}, x_{n+1}, \vec{N}, N_{n+1}, \vec{u}, \varepsilon} \Gamma$ by definition of $\rightsquigarrow$. Moreover, because $x_{n+1} \in V^{*^p}$ we have $x_{n+1} \notin \mathrm{FV}(\psi)$, and thus $\psi = \varphi[u_1/x_1] \dots [u_n/x_n][\varepsilon/x_{n+1}]$. ◀