



University of Pennsylvania
ScholarlyCommons

Technical Reports (CIS)

Department of Computer & Information Science

August 1993

Realizability, Covers, and Sheaves II. Applications to the Second-Order Lambda-Calculus

Jean H. Gallier

University of Pennsylvania, jean@cis.upenn.edu

Follow this and additional works at: https://repository.upenn.edu/cis_reports

Recommended Citation

Jean H. Gallier, "Realizability, Covers, and Sheaves II. Applications to the Second-Order Lambda-Calculus", . August 1993.

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-93-47.

This paper is posted at ScholarlyCommons. https://repository.upenn.edu/cis_reports/281
For more information, please contact repository@pobox.upenn.edu.

Realizability, Covers, and Sheaves II. Applications to the Second-Order Lambda-Calculus

Abstract

We present a general method for proving properties of typed λ -terms. This method is obtained by introducing a semantic notion of realizability which uses the notion of a cover algebra (as in abstract sheaf theory, a cover algebra being a Grothendieck topology in the case of a preorder). For this, we introduce a new class of semantic structures equipped with preorders, called pre-applicative structures. These structures need not be extensional. In this framework, a general realizability theorem can be shown. Applying this theorem to the special case of the term model, yields a general theorem for proving properties of typed λ -terms, in particular, strong normalization and confluence. This approach clarifies the reducibility method by showing that the closure conditions on candidates of reducibility can be viewed as sheaf conditions. Part II of this paper applies the above approach to the second-order (polymorphic) λ -calculus $\lambda^{\rightarrow, \forall^2}$ (with types \rightarrow and \forall).

Comments

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-93-47.

Realizability, Covers, and Sheaves.

II. Applications to the Second-Order Typed λ -Calculus

Preliminary Version

Jean Gallier*

Department of Computer and Information Science
University of Pennsylvania
200 South 33rd St.
Philadelphia, PA 19104, USA
e-mail: `jean@saul.cis.upenn.edu`

August 12, 1993

Abstract. We present a general method for proving properties of typed λ -terms. This method is obtained by introducing a semantic notion of realizability which uses the notion of a cover algebra (as in abstract sheaf theory, a cover algebra being a Grothendieck topology in the case of a preorder). For this, we introduce a new class of semantic structures equipped with preorders, called pre-applicative structures. These structures need not be extensional. In this framework, a general realizability theorem can be shown. Applying this theorem to the special case of the term model, yields a general theorem for proving properties of typed λ -terms, in particular, strong normalization and confluence. This approach clarifies the reducibility method by showing that the closure conditions on candidates of reducibility can be viewed as sheaf conditions. Part II of this paper applies the above approach to the second-order (polymorphic) λ -calculus $\lambda^{\rightarrow, \forall^2}$ (with types \rightarrow and \forall).

*This research was partially supported by ONR Grant N00014-88-K-0593.

1 Introduction

The following two questions were raised in part I of this paper:

1. **What is the connection between realizability and reducibility?**
2. **Is it possible to give more “semantic” versions of proofs using reducibility?**

In part I of this paper, we gave some answers to the above questions for the simply-typed λ -calculus $\lambda^{\rightarrow, \times, +, \perp}$. We defined an abstract notion of semantic realizability using the notion of a cover algebra (a Grothendieck topology, in the case of a preorder), and we proved a semantic realizability theorem. In part II of this paper, we generalize the approach developed in part I to the second-order (polymorphic) λ -calculus (with types \rightarrow and \forall). For this, we introduce a new class of structures equipped with preorders, called pre-applicative structures. In this framework, we have a type algebra T , that we use to interpret the (syntactic) types. Then, the set of realizers $\mathbf{r}[\sigma]\mu$ associated with a type σ depends on a valuation μ that assigns a pair $\langle s, S \rangle$ to every type variable, where s is an element of the type algebra T , and S is the s -component of some sheaf $\mathcal{S} = (S_s)_{s \in T}$. In this setting, it turns out that the family $(\mathbf{r}[\sigma]\mu)_{\sigma \in \mathcal{T}}$ of sets of realizers associated with the types, is itself a sheaf. Actually, we consider abstract properties \mathcal{P} of these sets of realizers. The main theorem is the following: provided that the abstract property \mathcal{P} satisfies some fairly simple conditions (P1)-(P5), if $\Gamma \triangleright M : \sigma$ and $\rho(y) \in \mathbf{r}[\delta]\mu$ for every $y : \delta \in \Gamma$, then the meaning $\mathcal{A}[\Gamma \triangleright M : \sigma]\rho$ of $\Gamma \triangleright M : \sigma$ is a realizer of σ that satisfies the property \mathcal{P} . As a corollary, considering a suitable term model for the second-order lambda calculus, we obtain simple proofs for strong normalization and confluence. This approach sheds some new light on the reducibility method and the conditions on the candidates of reducibility. These conditions can be viewed as sheaf conditions.

As in part I, in order to understand what motivated the definition of the semantic structures used in this paper, it is useful to review the definition of an applicative structure for the second-order (polymorphic) λ -calculus. In order to deal with second-order types, first, we need to provide an interpretation of the type variables. Thus, as in Breazu-Tannen and Coquand [1], we assume that we have an *algebra of types* T , which consists of a quadruple

$$\langle T, \rightarrow, [T \Rightarrow T], \forall \rangle,$$

where T is a nonempty set of types, $\rightarrow : T \times T \rightarrow T$ is a binary operations on T , $[T \Rightarrow T]$ is a nonempty set of functions from T to T , and \forall is a function $\forall : [T \Rightarrow T] \rightarrow T$.

Intuitively, given a valuation $\theta : \mathcal{V} \rightarrow T$ (where \mathcal{V} is the set of type variables), a type $\sigma \in T$ will be interpreted as an element $[\sigma]\theta$ of T . Then, a *second-order applicative structure* is defined as a tuple

$$\langle T, (A^s)_{s \in T}, (\mathbf{app}^{s,t})_{s,t \in T}, (\mathbf{tapp}^\Phi)_{\Phi \in [T \rightarrow T]} \rangle,$$

where

T is an algebra of types;

$(A^s)_{s \in T}$ is a family of nonempty sets called *carriers*,

$(\mathbf{app}^{s,t})_{s,t \in T}$ is a family of *application operators*, where each $\mathbf{app}^{s,t}$ is a total function $\mathbf{app}^{s,t} : A^{s \rightarrow t} \times A^s \rightarrow A^t$;

$(\text{tapp}^\Phi)_{\Phi \in [T \rightarrow T]}$ is a family of *type-application operators*, where each tapp^Φ is a total function $\text{tapp}^\Phi: A^{\forall(\Phi)} \times T \rightarrow \coprod (A^{\Phi(s)})_{s \in T}$, such that $\text{tapp}^\Phi(f, t) \in A^{\Phi(t)}$, for every $f \in A^{\forall(\Phi)}$, and every $t \in T$.

In order to define second-order applicative structures using operators like fun and abst , we need to define the curried version tfun^Φ of $\text{tapp}^\Phi: A^{\forall(\Phi)} \times T \rightarrow \coprod (A^{\Phi(s)})_{s \in T}$. For this, we define a kind of *dependent product* $\prod_\Phi (A^s)_{s \in T}$ (see definition 3.2). Then, we have families of operators $\text{tfun}^\Phi: A^{\forall(\Phi)} \rightarrow \prod_\Phi (A^s)_{s \in T}$, and $\text{tabst}^\Phi: \prod_\Phi (A^s)_{s \in T} \rightarrow A^{\forall(\Phi)}$, for every $\Phi \in [T \Rightarrow T]$.

Part II is organized as follows. The syntax of the second-order λ -calculus $\lambda^{\rightarrow, \forall^2}$ is reviewed in section 2. Pre-applicative structures are defined in section 3. The crucial notions of \mathcal{P} -cover algebras and of \mathcal{P} -sheaves are defined in section 4. The notion of \mathcal{P} -realizability is defined in section 5. In section 6, it is shown how to interpret terms in $\lambda^{\rightarrow, \forall^2}$ in pre-applicative structures, and some examples are given. The realizability theorem for the second-order typed λ -calculus $\lambda^{\rightarrow, \forall^2}$ is shown in section 7. Section 8 contains an application of the main theorem of section 7 to prove a general theorem about terms of the system $\lambda^{\rightarrow, \forall^2}$. Section 9 contains the conclusion and some suggestions for further research.

2 Syntax of the Second-Order Typed λ -Calculus $\lambda^{\rightarrow, \times, +, \forall^2}$

In this section, we review quickly the syntax of the second-order typed λ -calculus $\lambda^{\rightarrow, \times, +, \forall^2}$. This includes a definition of the second-order types under consideration, of raw terms, or the type-checking rules for judgements, and of the reduction rules. For more details (on the subsystem $\lambda^{\rightarrow, \forall^2}$), the reader should consult Breazu-Tannen and Coquand [1].

Let \mathcal{T} denote the set of second-order types. This set comprises type variables X , type constants k , and compound types $(\sigma \rightarrow \tau)$, $(\sigma \times \tau)$, $(\sigma + \tau)$, and $\forall X. \sigma$. It is assumed that we have a set TC of *type constants* (also called *base types of kind \star*). We have a countably infinite set \mathcal{V} of type variables (denoted as upper case letters X, Y, Z), and a countably infinite set \mathcal{X} of term variables (denoted as lower case letters x, y, z). We denote the set of free type variables occurring in a type σ as $FTV(\sigma)$. We use the notation \star for the kind of types. Since we are only considering second-order quantification over predicate symbols (of kind \star) of arity 0, this is superfluous. However, it will occasionally be useful to consider contexts Γ in which type variables are explicitly present, since this makes the type-checking rules more uniform in the case of λ -abstraction and typed λ -abstraction. Thus, officially, a context Γ is a set $\{x_1: \sigma_1, \dots, x_n: \sigma_n\}$, where x_1, \dots, x_n are term variables, and $\sigma_1, \dots, \sigma_n$ are types. We let $\text{dom}(\Gamma) = \{x_1, \dots, x_n\}$. As usual, we assume that the variables x_j are pairwise distinct. We also assume that $x \notin \text{dom}(\Gamma)$ in a context $\Gamma, x: \sigma$. Informally, we will also consider contexts $\{X_1: \star, \dots, X_m: \star, x_1: \sigma_1, \dots, x_n: \sigma_n\}$, where X_1, \dots, X_m are type variables, and x_1, \dots, x_n are term variables, with the two sets $\{X_1, \dots, X_m\}$ and $\{x_1, \dots, x_n\}$ disjoint, the variables X_i pairwise distinct, and the variables x_j pairwise distinct. We assume that $X \notin \text{dom}(\Gamma)$ in a context $\Gamma, X: \star$. For the sake of brevity, rather than writing typed λ -abstraction as $\lambda X: \star. M$, it will be written as $\lambda X. M$.

It is assumed that we have a set $Const$ of constants, together with a function $Type: Const \rightarrow \mathcal{T}$, such that every constant c is assigned a *closed type* $Type(c)$ in \mathcal{T} . The set TC of type constants, together with the set $Const$ of constants, and the function $Type$, constitute a *signature* Σ . Let us review the definition of raw terms.

Definition 2.1 The set of *raw terms* is defined inductively as follows: every variable $x \in \mathcal{X}$ is a raw term, every constant $c \in \text{Const}$ is a raw terms, and if M, N are raw terms and σ, τ are types, then (MN) , $(M\tau)$, $\lambda x:\sigma. M$, $\lambda X. M$, $\pi_1(M)$, $\pi_2(M)$, $\langle M, N \rangle$, $\text{inl}(M)$, $\text{inr}(M)$, and $[M, N]$, are raw terms.

We let $FV(M)$ denote the set of free term-variables in M . Raw terms may contain free variables and may not type-check (for example, (xx)). In order to define which raw terms type-check, we consider expressions of the form $\Gamma \triangleright M:\sigma$, called *judgements*, where Γ is a context in which all the free term variables in M are declared. A term M type-checks with type σ in the context Γ iff the judgement $\Gamma \triangleright M:\sigma$ is provable using axioms and rules summarized in the following definition.

Definition 2.2 The judgements of the polymorphic typed λ -calculus $\lambda^{\rightarrow, \times, +, \forall^2}$ are defined by the following rules.

$$\begin{array}{c}
\Gamma \triangleright x:\sigma, \quad \text{when } x:\sigma \in \Gamma, \\
\Gamma \triangleright c:\text{Type}(c), \quad \text{when } c \text{ is a constant,} \\
\frac{\Gamma, x:\sigma \triangleright M:\tau}{\Gamma \triangleright (\lambda x:\sigma. M):(\sigma \rightarrow \tau)} \quad (\text{abstraction}) \\
\frac{\Gamma \triangleright M:(\sigma \rightarrow \tau) \quad \Gamma \triangleright N:\sigma}{\Gamma \triangleright (MN):\tau} \quad (\text{application}) \\
\frac{\Gamma \triangleright M:\sigma \quad \Gamma \triangleright N:\tau}{\Gamma \triangleright \langle M, N \rangle:\sigma \times \tau} \quad (\text{pairing}) \\
\frac{\Gamma \triangleright M:\sigma \times \tau}{\Gamma \triangleright \pi_1(M):\sigma} \quad (\text{projection}) \quad \frac{\Gamma \triangleright M:\sigma \times \tau}{\Gamma \triangleright \pi_2(M):\tau} \quad (\text{projection}) \\
\frac{\Gamma \triangleright M:\sigma}{\Gamma \triangleright \text{inl}(M):\sigma + \tau} \quad (\text{injection}) \quad \frac{\Gamma \triangleright M:\tau}{\Gamma \triangleright \text{inr}(M):\sigma + \tau} \quad (\text{injection}) \\
\frac{\Gamma \triangleright M:(\sigma \rightarrow \delta) \quad \Gamma \triangleright N:(\tau \rightarrow \delta)}{\Gamma \triangleright [M, N]:(\sigma + \tau) \rightarrow \delta} \quad (\text{co-pairing}) \\
\frac{\Gamma, X:\star \triangleright M:\sigma}{\Gamma \triangleright (\lambda X. M):\forall X. \sigma} \quad (\forall\text{-intro})
\end{array}$$

provided that $X \notin \bigcup_{x:\tau \in \Gamma} FTV(\tau)$;

$$\frac{\Gamma \triangleright M:\forall X. \sigma}{\Gamma \triangleright (M\tau):\sigma[\tau/X]} \quad (\forall\text{-elim})$$

The reason why we do not officially consider that a context contains type variables, is that in the rule (\forall -elim), the type τ could contain type variables not declared in Γ , and it would be necessary to have a weakening rule to add new type variables to a context (or some other mechanism to add new type variables to a context). As long as we do not deal with dependent types, this technical annoyance is most simply circumvented by assuming that type variables are not included in contexts.

Instead of using the construct $\text{case } P \text{ of } \text{inl}(x:\sigma) \Rightarrow M \mid \text{inr}(y:\tau) \Rightarrow N$, we found it more convenient and simpler to use the slightly more general construct $[M, N]$, where M is of type $\sigma \rightarrow \delta$ and N is of type $\tau \rightarrow \delta$, even when M and N are not λ -abstractions. This will be especially advantageous for the semantic treatment to follow. Then, we can define the conditional construct $\text{case } P \text{ of } \text{inl}(x:\sigma) \Rightarrow M \mid \text{inr}(y:\tau) \Rightarrow N$, where P is of type $\sigma + \tau$, as $[\lambda x:\sigma. M, \lambda y:\tau. N]P$.

Definition 2.3 The reduction rules of the system $\lambda^{\rightarrow, \times, +, \forall}$ are listed below:

$$\begin{aligned} (\lambda x:\sigma. M)N &\longrightarrow M[N/x], \\ \pi_1(\langle M, N \rangle) &\longrightarrow M, \\ \pi_2(\langle M, N \rangle) &\longrightarrow N, \\ [M, N]\text{inl}(P) &\longrightarrow MP, \\ [M, N]\text{inr}(P) &\longrightarrow NP, \\ (\lambda X. M)\tau &\longrightarrow M[\tau/X]. \end{aligned}$$

The reduction relation defined by the rules of definition 2.3 is denoted as \longrightarrow_β (even though there are reductions other than β -reduction). From now on, when we refer to a λ -term, we mean a λ -term that type-checks. We let $\Lambda_{(\sigma, \Gamma)}$ denote the set of judgements of the form $\Gamma \triangleright M:\sigma$.

3 Pre-Applicative Structures

In this section, some new semantic structures called pre-applicative structures are defined. There are various kinds of pre-applicative structures: pre-applicative β -structures, pre-applicative $\beta\eta$ -structures, extensional pre-applicative β -structures, and the corresponding so-called applicative versions. Since we are dealing with type variables, in order to interpret the types, we first need to define the notion of an algebra of (polymorphic) types. We also need to define the notion of a dependent product (see definition 3.2) in order to “curry” the map $\text{tapp}^\Phi: A^{\forall(\Phi)} \times T \rightarrow \coprod (A^{\Phi(s)})_{s \in T}$.

Definition 3.1 An *algebra of (polymorphic) types* is a tuple

$$\langle T, \rightarrow, \times, +, [T \Rightarrow T], \forall \rangle,$$

where T is a nonempty set of *types*, $\rightarrow, \times, +: T \times T \rightarrow T$ are binary operations on T , $[T \Rightarrow T]$ is a nonempty set of functions from T to T , and \forall is a function $\forall: [T \Rightarrow T] \rightarrow T$.

Intuitively, given a valuation $\theta: \mathcal{V} \rightarrow T$, a type $\sigma \in T$ will be interpreted as an element $[\sigma]\theta$ of T .

Given an indexed family of sets $(A_i)_{i \in I}$, we let $\prod (A_i)_{i \in I}$ be the product of the family $(A_i)_{i \in I}$, and $\coprod (A_i)_{i \in I}$ be the coproduct (or disjoint sum) of the family $(A_i)_{i \in I}$. The disjoint sum $\coprod (A_i)_{i \in I}$ is the set $\bigcup \{(a, i) \mid a \in A_i\}_{i \in I}$. If the sets A_i are preorders, then $\prod (A_i)_{i \in I}$ is a preorder under the product preorder, where $(a_i)_{i \in I} \preceq (b_i)_{i \in I}$ iff $a_i \preceq_i b_i$ for all $i \in I$, and $\coprod (A_i)_{i \in I}$ is a preorder under the (disjoint) sum preorder, where $\langle a, i \rangle \preceq \langle b, j \rangle$ iff $i = j$ and $a \preceq_i b$. When $I = \{1, 2\}$, we also denote $\prod (A_i)_{i \in I}$ as $A_1 \times A_2$, and $\coprod (A_i)_{i \in I}$ as $A_1 + A_2$.

Before defining a pre-applicative structure, we need to define the notion of a dependent product.

Definition 3.2 Given an algebra of types T , and a T -indexed family of preorders $\langle A^s, \preceq^s \rangle$, for every function $\Phi \in [T \Rightarrow T]$, the *dependent product* $\prod_{\Phi}(A^s)_{s \in T}$ is the cartesian product $\prod_{t \in T}(A^{\Phi(t)})$, which is also described explicitly as the set of functions in $(\prod_{\Phi}(A^{\Phi(s)})_{s \in T})^T$ defined as follows:

$$\prod_{\Phi}(A^s)_{s \in T} = \{f: T \rightarrow \prod_{\Phi}(A^{\Phi(s)})_{s \in T} \mid f(t) \in A^{\Phi(t)}, \text{ for all } t \in T\}.$$

The set $\prod_{\Phi}(A^s)_{s \in T}$ is given the preorder \preceq^{Φ} defined such that, $f \preceq^{\Phi} g$ iff $f(t) \preceq^{\Phi(t)} g(t)$, for every $t \in T$.

Given two preordered sets $\langle A^s, \preceq^s \rangle$ and $\langle A^t, \preceq^t \rangle$, we let $[A^s \Rightarrow A^t]$ be the set of monotonic functions w.r.t. \preceq^s and \preceq^t , under the pointwise preorder induced by \preceq^t defined such that, $f \preceq g$ iff $f(a) \preceq^t g(a)$ for all $a \in A^s$.

We are now ready to define the semantic structures used in this paper.

Definition 3.3 Given an algebra of types T , a *pre-applicative β -structure* is a structure

$$\mathcal{A} = \langle A, \preceq, \text{fun}, \text{abst}, \text{tfun}, \text{tabst}, \Pi, \langle -, - \rangle, \text{inl}, \text{inr}, [-, -] \rangle,$$

where

- $A = (A^s)_{s \in T}$ is a family of sets (possibly empty) called *carriers*;
- $(\preceq^s)_{s \in T}$ is a family of preorders, each \preceq^s on A^s ;
- $\text{abst}^{s,t}: [A^s \Rightarrow A^t] \rightarrow A^{s \rightarrow t}$, a family of partial operators;
- $\text{fun}^{s,t}: A^{s \rightarrow t} \rightarrow [A^s \Rightarrow A^t]$, a family of (total) operators;
- $\text{tabst}^{\Phi}: \prod_{\Phi}(A^s)_{s \in T} \rightarrow A^{\vee(\Phi)}$, a family of partial operators, for every $\Phi \in [T \Rightarrow T]$;
- $\text{tfun}^{s,t}: A^{\vee(\Phi)} \rightarrow \prod_{\Phi}(A^s)_{s \in T}$, a family of (total) operators, for every $\Phi \in [T \Rightarrow T]$;
- $\langle -, - \rangle^{s,t}: A^s \times A^t \rightarrow A^{s \times t}$, a family of partial pairing operators;
- $\Pi^{s,t}: A^{s \times t} \rightarrow A^s \times A^t$, a family of (total) projection operators;
- $[-, -]^{s,t,d}: A^{s \rightarrow d} \times A^{t \rightarrow d} \rightarrow A^{(s+t) \rightarrow d}$, a family of partial copairing operators;
- $\text{inl}^{s,t}: A^s \rightarrow A^{s+t}$, a family of (total) operators;
- $\text{inr}^{s,t}: A^t \rightarrow A^{s+t}$, a family of (total) operators.

We define $\text{cinl}: A^{(s+t) \rightarrow d} \rightarrow [A^s \Rightarrow A^d]$ and $\text{cinr}: A^{(s+t) \rightarrow d} \rightarrow [A^t \Rightarrow A^d]$ as follows: For every $h \in A^{(s+t) \rightarrow d}$,

$$\text{cinl}(h)(a) = \text{fun}(h)(\text{inl}(a)),$$

for every $a \in A^s$, and

$$\text{cinr}(h)(b) = \text{fun}(h)(\text{inr}(b)),$$

for every $b \in A^t$.

It is assumed that fun , abst , tfun , tabst , Π , $\langle -, - \rangle$, inl , inr , and $[-, -]$, are monotonic. Furthermore, the following conditions are satisfied

- (1) For all $s, t \in T$, if $A^s \neq \emptyset$ and $A^t \neq \emptyset$, then $A^{s \rightarrow t} \neq \emptyset$, and $\text{fun}^{s,t}(\text{abst}^{s,t}(\varphi)) \succeq \varphi$, whenever $\text{abst}^{s,t}(\varphi)$ is defined for $\varphi \in [A^s \Rightarrow A^t]$;

- (2) If $A^{\Phi(t)} \neq \emptyset$ for every $t \in T$, then $A^{\vee(\Phi)} \neq \emptyset$, and $\text{tfun}^\Phi(\text{tabst}^\Phi(\varphi)) \succeq \varphi$, whenever $\text{tabst}^\Phi(\varphi)$ is defined for $\varphi \in \prod_{\Phi}(A^s)_{s \in T}$;
- (3) For all $s, t \in T$, if $A^s \neq \emptyset$ and $A^t \neq \emptyset$, then $A^{s \times t} \neq \emptyset$, and $\Pi^{s,t}(\langle a, b \rangle) \succeq (a, b)$, for all $a \in A^s, b \in A^t$, whenever $\langle a, b \rangle$ is defined;
- (4) For all $s, t \in T$, if $A^s \neq \emptyset$ and $A^t \neq \emptyset$, then $A^{s+t} \neq \emptyset$, and $\text{cinl}([f, g]) \succeq \text{fun}(f)$, and $\text{cinr}([f, g]) \succeq \text{fun}(g)$, whenever $[f, g]$ is defined, for $f \in A^{s \rightarrow d}$ and $g \in A^{t \rightarrow d}$.

The operators fun induce (total) operators

$\text{app}^{s,t}: A^{s \rightarrow t} \times A^s \rightarrow A^t$, such that, for every $f \in A^{s \rightarrow t}$ and every $a \in A^s$,

$$\text{app}^{s,t}(f, a) = \text{fun}^{s,t}(f)(a).$$

Then, condition (1) can be written as

- (1') $\text{app}^{s,t}(\text{abst}^{s,t}(\varphi), a) \succeq \varphi(a)$, for every $a \in A^s$, for $\varphi \in [A^s \Rightarrow A^t]$, whenever $\text{abst}^{s,t}(\varphi)$ is defined, and condition (4) can be rewritten as
- (4') $\text{cinl}([f, g])(a) \succeq \text{app}(f, a)$, for all $a \in A^s$, and $\text{cinr}([f, g])(b) \succeq \text{app}(g, b)$, for all $b \in A^t$, whenever $[f, g]$ is defined, for $f \in A^{s \rightarrow d}$ and $g \in A^{t \rightarrow d}$.

The operators tfun induce (total) operators

$\text{tapp}^\Phi: A^{\vee(\Phi)} \times T \rightarrow \prod(A^{\Phi(s)})_{s \in T}$, such that, for every $t \in T$,

$$\text{tapp}^\Phi(f, t) = \text{tfun}^\Phi(f)(t).$$

Then, condition (2) can be written as

- (2') $\text{tapp}^\Phi(\text{tabst}^\Phi(\varphi), s) \succeq \varphi(s)$, for every $s \in T$, whenever $\text{tabst}^\Phi(\varphi)$ is defined, for $\varphi \in \prod_{\Phi}(A^s)_{s \in T}$.

Finally, $N \preceq \text{inl}(M_1)$ implies that $N = \text{inl}(N_1)$ for some $N_1 \preceq M_1$, and $N \preceq \text{inr}(M_1)$ implies that $N = \text{inr}(N_1)$ for some $N_1 \preceq M_1$.

We say that a pre-applicative β -structure is an *applicative β -structure* iff in conditions (1)-(4), \succeq is replaced by the identity relation $=$.

We will omit superscripts whenever possible. Intuitively, \mathcal{A} is a set of realizers. It is shown in section 6 how the term model can be viewed as a pre-applicative β -structure (see definition 6.5).

The projection operators Π induce projections $\pi_1^{s,t}: A^{s \times t} \rightarrow A^s$ and $\pi_2^{s,t}: A^{s \times t} \rightarrow A^t$, such that for every $a \in A^{s \times t}$, if $\Pi^{s,t}(a) = (a_1, a_2)$, then

$$\pi_1^{s,t}(a) = a_1 \quad \text{and} \quad \pi_2^{s,t}(a) = a_2.$$

When \mathcal{A} is an applicative β -structure, then, in definition 3.3, conditions (1)-(4) amounts to

- (1) $\text{fun}^{s,t} \circ \text{abst}^{s,t} = \text{id}$ on the domain of definition of abst ;
- (2) $\text{tfun}^\Phi \circ \text{tabst}^\Phi = \text{id}$ on the domain of definition of tabst ;
- (3) $\Pi^{s,t} \circ \langle -, - \rangle^{s,t} = \text{id}$ on the domain of definition of $\langle -, - \rangle$; and
- (4) $\langle \text{cinl}, \text{cinr} \rangle \circ [-, -] = \text{fun}^{s,d} \times \text{fun}^{t,d}$ on the domain of definition of $[-, -]$.

In view of (1), from (4), we get

$$\langle \text{cinl}, \text{cinr} \rangle \circ ([-, -] \circ (\text{abst}^{s,d} \times \text{abst}^{t,d})) = \text{id} \text{ on the domain of definition of } [-, -] \circ (\text{abst}^{s,d} \times \text{abst}^{t,d}).$$

In this case, abst is injective and fun is surjective on the domain of definition of abst (and left inverse to abst), tabst is injective and tfun is surjective on the domain of definition of tabst (and left inverse to tabst), $\langle -, - \rangle$ is injective and Π is surjective on the domain of definition of $\langle -, - \rangle$ (and left inverse to $\langle -, - \rangle$), $[-, -] \circ (\text{abst}^{s,d} \times \text{abst}^{t,d})$ is injective on its domain of definition, and $\langle \text{cinl}, \text{cinr} \rangle$ is surjective on this domain (and left inverse to $[-, -] \circ (\text{abst}^{s,d} \times \text{abst}^{t,d})$).

When we use a pre-applicative β -structure to interpret λ -terms, we assume that $\langle -, - \rangle$ and $[-, -]$ are total, and that the domains of abst and tabst are sufficiently large, but we have not elucidated this last condition yet. Given $M \in A^{s \rightarrow t}$ and $N \in A^s$, $\text{app}(M, N)$ is also denoted as MN , and $\text{tapp}(M, t)$ as Mt .

We now define extensional pre-applicative structures. First, we define isotonicity. Given a monotonic function $f: W_1 \rightarrow W_2$, where W_1 and W_2 are preorders, we say that f is *isotone* iff $f(w_1) \preceq f(w_2)$ implies that $w_1 \preceq w_2$, for all $w_1, w_2 \in W_1$.

Definition 3.4 A pre-applicative β -structure \mathcal{A} is *extensional* iff fun , tfun , Π , and $\langle \text{cinl}, \text{cinr} \rangle$, are isotone, and the following conditions hold:

- (1) $\text{ran}(\text{fun}) \subseteq \text{dom}(\text{abst})$;
- (2) $\text{ran}(\text{tfun}) \subseteq \text{dom}(\text{tabst})$;
- (3) $\text{ran}(\Pi) \subseteq \text{dom}(\langle -, - \rangle)$;
- (4) $\text{ran}(\langle \text{cinl}^{s,t,d}, \text{cinr}^{s,t,d} \rangle) \subseteq \text{dom}([-, -] \circ (\text{abst}^{s,d} \times \text{abst}^{t,d}))$.

When \mathcal{A} is an applicative β -structure, conditions (1)-(4) hold, and the functions fun , tfun , Π , and $\langle \text{cinl}, \text{cinr} \rangle$, are injective, we say that we have an *extensional applicative β -structure*.

When \mathcal{A} is an extensional pre-applicative β -structure, in view of condition (1), $\text{abst}(\text{fun}(f))$ is defined for any $f \in A^{s \rightarrow t}$. Observe that by condition (1) of definition 3.3, we have $\text{fun}(f) \preceq \text{fun}(\text{abst}(\text{fun}(f)))$, and since fun is isotone, this implies that

- (1) $\text{abst}(\text{fun}(f)) \succeq f$, for all $f \in A^{s \rightarrow t}$.

Similarly, we can prove that

- (2) $\text{tabst}(\text{tfun}(f)) \succeq f$, for all $f \in A^{\vee(\otimes)}$;
- (3) $\langle \pi_1(a), \pi_2(a) \rangle \succeq a$, for all $a \in A^{s \times t}$; and
- (4) $[\text{abst}(\text{cinl}(h)), \text{abst}(\text{cinr}(h))] \succeq h$, for all $h \in A^{(s+t) \rightarrow d}$.

We will call the above inequalities the *η -like rules*.

In many cases, a pre-applicative β -structure that satisfies the η -like rules is not extensional. This motivates the next definition.

Definition 3.5 A pre-applicative β -structure \mathcal{A} is a $\beta\eta$ -structure if the following conditions hold:

- (1) $\text{ran}(\text{fun}) \subseteq \text{dom}(\text{abst})$, and $\text{abst}(\text{fun}(f)) \succeq f$, for all $f \in A^{s \rightarrow t}$;
- (2) $\text{ran}(\text{tfun}) \subseteq \text{dom}(\text{tabst})$, and $\text{tabst}(\text{tfun}(f)) \succeq f$, for all $f \in A^{\vee(\Phi)}$;
- (3) $\text{ran}(\Pi) \subseteq \text{dom}(\langle -, - \rangle)$, and $\langle \pi_1(a), \pi_2(a) \rangle \succeq a$, for all $a \in A^{s \times t}$; and
- (4) $\text{ran}(\langle \text{cinl}^{s,t,d}, \text{cinr}^{s,t,d} \rangle) \subseteq \text{dom}([_, _] \circ (\text{abst}^{s,d} \times \text{abst}^{t,d}))$, and $[\text{abst}(\text{cinl}(h)), \text{abst}(\text{cinr}(h))] \succeq h$, for all $h \in A^{(s+t) \rightarrow d}$.

When \mathcal{A} is an applicative β -structure and in conditions (1)-(4), \succeq is replaced by $=$, we say that we have an *applicative $\beta\eta$ -structure*.

It is shown in section 6 how the term model can be viewed as a pre-applicative $\beta\eta$ -structure (see definition 6.7). From the remark before definition 3.5, an extensional pre-applicative β -structure is a $\beta\eta$ -structure. When \mathcal{A} is an applicative $\beta\eta$ -structure, conditions (1)-(4) of definition 3.5 amount to:

- (1) $\text{abst}^{s,t} \circ \text{fun}^{s,t} = \text{id}$;
- (2) $\text{tabst}^\Phi \circ \text{tfun}^\Phi = \text{id}$;
- (3) $\langle -, - \rangle^{s,t} \circ \Pi^{s,t} = \text{id}$; and
- (4) $([_, _] \circ (\text{abst}^{s,d} \times \text{abst}^{t,d})) \circ \langle \text{cinl}^{s,t,d}, \text{cinr}^{s,t,d} \rangle = \text{id}$.

This implies that fun , tfun , Π , and $\langle \text{cinl}, \text{cinr} \rangle$, are injective. Thus, an applicative $\beta\eta$ -structure is extensional. In this case, (together with conditions (1)-(4) of definition 3.3 in the case of an applicative β -structure), we have $\text{dom}(\text{abst}) = \text{fun}(A^{s \rightarrow t})$, fun is a bijection between $A^{s \rightarrow t}$ and a subset of $[A^s \Rightarrow A^t]$ (with inverse abst), $\text{dom}(\text{tabst}) = \text{tfun}(A^{\vee(\Phi)})$, tfun is a bijection between $A^{\vee(\Phi)}$ and a subset of $\prod_{\Phi}(A^s)_{s \in T}$ (with inverse tabst), Π is a bijection between $A^{s \times t}$ and a subset of $A^s \times A^t$ (with inverse $\langle -, - \rangle$), and $\langle \text{cinl}^{s,t,d}, \text{cinr}^{s,t,d} \rangle$ is a bijection between $A^{(s+t) \rightarrow d}$ and a subset of $[A^s \Rightarrow A^d] \times [A^t \Rightarrow A^d]$ (with inverse $[_, _] \circ (\text{abst}^{s,d} \times \text{abst}^{t,d})$).

4 \mathcal{P} -Cover Algebras and \mathcal{P} -Sheaves

In this section, we basically repeat the definitions for covers and sheaves given in part I of this paper, except that we are dealing with a more general notion of pre-applicative structure (since we also have an algebra of types T). As in part I, we define all the necessary concepts in terms of preorders, referring the interested reader to MacLane and Moerdijk [6] for a general treatment. First, we need some preliminary definitions before defining the crucial notion of a cover. From now on, unless specified otherwise, it is assumed that we are dealing with pre-applicative β -structures (and thus, we will omit the prefix β).

Definition 4.1 Given an algebra of types T and a pre-applicative structure \mathcal{A} , for any $M \in A^s$, a *sieve on M* is any subset $C \subseteq A^s$ such that, $N \preceq M$ for every $N \in C$, and whenever $N \in C$ and $Q \preceq N$, then $Q \in C$. In other words, a sieve on M is downwards closed and below M (it is an *ideal below M*). The sieve $\{N \mid N \preceq M\}$ is called the *maximal (or principal) sieve* on M . A *covering family* on a pre-applicative structure \mathcal{A} is a family Cov of binary relations Cov_s on $2^{A^s} \times A^s$,

relating subsets of A^s called *covers*, to elements of A^s . Equivalently, \mathbf{Cov} can be defined as a family of functions $\mathbf{Cov}_s: A^s \rightarrow 2^{2^{A^s}}$ assigning to every element $M \in A^s$ a set $\mathbf{Cov}(M)$ of subsets of A^s (the covers of M). Given any $M \in A^s$, the empty cover \emptyset and the principal sieve $\{N \mid N \preceq M\}$ are the *trivial* covers. We let $\mathbf{triv}(M)$ denote the set consisting of the two trivial covers of M . A cover which is not trivial is called *nontrivial*.

In the rest of this paper, we will consider binary relations $\mathcal{P} \subseteq \mathcal{A} \times T$, such that $\mathcal{P}(M, s)$ implies $M \in A^s$, and for every $s \in T$, if $A^s \neq \emptyset$, then there is some $M \in A^s$ s.t. $\mathcal{P}(M, s)$. Equivalently, \mathcal{P} can be viewed as a family $\mathcal{P} = (P_s)_{s \in T}$, where each P_s is a nonempty subset of A^s (unless $A^s = \emptyset$). The intuition behind \mathcal{P} is that it is a property of realizers. For simplicity, we define the covering conditions only for the types \rightarrow and \forall^2 (but the types \times , $+$ and \perp , can also be handled. This treatment can be readily adapted from part I).

Definition 4.2 Given an algebra of types T , let \mathcal{A} be a pre-applicative structure and let \mathcal{P} be a family $\mathcal{P} = (P_s)_{s \in T}$, where each P_s is a nonempty subset of A^s (unless $A^s = \emptyset$). A \mathcal{P} -*cover algebra* (or \mathcal{P} -*Grothendieck topology*) on \mathcal{A} is a family \mathbf{Cov} of binary relations \mathbf{Cov}_s on $2^{A^s} \times A^s$ satisfying the following properties:

- (0) $\mathbf{Cov}_s(C, M)$ implies $M \in P_s$ (equivalently, $\mathcal{P}(M, s)$).
- (1) If $\mathbf{Cov}(C, M)$, then C is a sieve on M (an ideal below M).
- (2) If $M \in P_s$, then $\mathbf{Cov}(\{N \mid N \preceq M\}, M)$ ($M \in P_s$ is covered by the principal sieve on M).
- (3) (stability) If $\mathbf{Cov}(C, M)$ and $N \preceq M$, then $\mathbf{Cov}(\{Q \mid Q \in C, Q \preceq N\}, N)$.
- (4) (transitivity) If $\mathbf{Cov}(C, M)$, D is a sieve on M , and $\mathbf{Cov}(\{Q \mid Q \in D, Q \preceq N\}, N)$ for every $N \in C$, then $\mathbf{Cov}(D, M)$.
- (5) If $\mathbf{Cov}(M) = \mathbf{triv}(M)$, then $\mathbf{Cov}(MN) = \mathbf{triv}(MN)$, and if $\mathbf{Cov}(C, M)$ and $\mathbf{Cov}(D, MN)$ with C and D nontrivial, then for every $Q \in D$, there is some $M' \in C$ such that $Q \preceq M'N$.
- (6) If $\mathbf{Cov}(M) = \mathbf{triv}(M)$, then $\mathbf{Cov}(Ms) = \mathbf{triv}(Ms)$, where $s \in T$, and if $\mathbf{Cov}(C, M)$ and $\mathbf{Cov}(D, Ms)$ with C and D nontrivial, then for every $Q \in D$, there is some $M' \in C$ such that $Q \preceq M's$.

A triple $\langle \mathcal{A}, \mathcal{P}, \mathbf{Cov} \rangle$, where \mathcal{A} is pre-applicative structure, \mathcal{P} is a property on \mathcal{A} , and \mathbf{Cov} is a \mathcal{P} -Grothendieck topology, is called a \mathcal{P} -*site*.

Condition (0) is needed to restrict attention to elements having the property \mathcal{P} . Covers only matter for these elements. Conditions (1)-(4) are the conditions for a set of sieves to be a Grothendieck topology, in the case where the base category is a preorder $\langle \mathcal{A}, \preceq \rangle$. Conditions (5)-(6) are needed to take care of the extra structure.

It should be noted that conditions (3) and (4) are in fact only needed for the treatment of the sum type $+$ (or the existential type). Also, it is not necessary to assume that covers are ideals (downwards closed), but this is not harmful.

Definition 4.3 We say that $M \in A^s$ is *simple* iff $\mathbf{Cov}(C, M)$ for at least two distinct covers C . We say that $M \in A^s$ is *stubborn* iff $\mathbf{Cov}(M) = \{\emptyset, \{Q \mid Q \preceq M\}\}$ (thus every stubborn element is simple). We say that a \mathcal{P} -site $\langle \mathcal{A}, \mathcal{P}, \mathbf{Cov} \rangle$ is *scenic* iff all elements of the form $\mathbf{app}(M, N)$ (or MN), or $\mathbf{tapp}(M, s)$ (or Ms), are simple.

From now on, we only consider scenic \mathcal{P} -sites. In order for our realizability theorem to hold, realizers will have to satisfy properties analogous to the properties (P1)-(P5) mentioned in the introduction of part I.

Definition 4.4 Given an algebra of types T , let $\langle \mathcal{A}, \mathcal{P}, \text{Cov} \rangle$ be a \mathcal{P} -site. Properties (P1)-(P3) are defined as follows:

- (P1) $\mathcal{P}(M, s)$, for some stubborn element $M \in A^s$.
- (P2) If $\mathcal{P}(M, s)$ and $M \succeq N$, then $\mathcal{P}(N, s)$.
- (P3a) If $\text{Cov}_{s \rightarrow t}(C, M)$, $\mathcal{P}(N, s)$, and $\mathcal{P}(M'N, t)$ whenever $M' \in C$, then $\mathcal{P}(MN, t)$.
- (P3b) If $\text{Cov}_{\forall(\Phi)}(C, M)$, $s \in T$, and $\mathcal{P}(M's, \Phi(s))$ whenever $M' \in C$, then $\mathcal{P}(Ms, \Phi(s))$.

From now on, we only consider relations (families) \mathcal{P} satisfying conditions (P1)-(P3) of definition 4.4. Condition (P1) says that each P_s contains some stubborn element. We are now ready for the crucial notion of a sheaf property. This property is a crucial inductive invariant with respect to the notion of realizability defined in section 5.

Definition 4.5 Given an algebra of types T , let $\langle \mathcal{A}, \mathcal{P}, \text{Cov} \rangle$ be a \mathcal{P} -site. A function $\mathcal{S}: \mathcal{A} \rightarrow 2^T$ has the *sheaf property* (or is a *\mathcal{P} -sheaf*) iff it satisfies the following conditions:

- (S1) If $s \in \mathcal{S}(M)$, then $M \in P_s$.
- (S2) If $s \in \mathcal{S}(M)$ and $M \succeq N$, then $s \in \mathcal{S}(N)$.
- (S3) If $\text{Cov}_s(C, M)$ and $s \in \mathcal{S}(N)$ for every $N \in C$, then $s \in \mathcal{S}(M)$.

A function $\mathcal{S}: \mathcal{A} \rightarrow 2^T$ as in definition 4.5 can also be viewed as a family $\mathcal{S} = (S_s)_{s \in T}$, where $S_s = \{M \in \mathcal{A} \mid s \in \mathcal{S}(M)\}$. Then, the sets S_s are called *\mathcal{P} -candidates*. The conditions of definition 4.5 are then stated as follows:

- (S1) $S_s \subseteq P_s$.
- (S2) If $M \in S_s$ and $M \succeq N$, then $N \in S_s$.
- (S3) If $\text{Cov}_s(C, M)$, and $C \subseteq S_s$, then $M \in S_s$.

This second set of conditions is slightly more convenient for proving our results. Note that according to the first definition, \mathcal{S} can also be viewed as a mapping

$$\mathcal{S}: \mathcal{A} \rightarrow \mathbf{Sets}.$$

Then, (S2) means that $M \succeq N$ implies $\mathcal{S}(M) \subseteq \mathcal{S}(N)$. Thus, \mathcal{S} is in fact a functor

$$\mathcal{S}: \mathcal{A}^{op} \rightarrow \mathbf{Sets},$$

viewing \mathcal{A}^{op} equipped with the preorder \succeq , the opposite of the preorder \preceq , as a category. As in part I, the conditions of definition 4.5 mean that this functor is a sheaf for the Grothendieck topology of definition 4.2.

Note that condition (S3) is trivial when C is the principal cover on M , since in this case, M belongs to C . Thus, condition (S3) is only interesting when M is simple, and from now on, this is what we will assume when using condition (S3). Also, since $\text{Cov}_s(C, M)$ implies that $\mathcal{P}(M, s)$, any \mathcal{P} satisfying conditions (P1)-(P3) trivially satisfies the sheaf property. Finally, note that (S3) and (P1) imply that S_s is nonempty and contains all stubborn elements in P_s (unless $A^s = \emptyset$).

By (P3a), if $M \in P_{s \rightarrow t}$ is stubborn and $N \in P_s$ is any element, then $MN \in P_t$. Furthermore, MN is also stubborn. This follows from property (5) of a cover. Thus, if $M \in P_{s \rightarrow t}$ is stubborn and $N \in P_s$ is any element, then $MN \in P_t$ is stubborn. Similarly, by (P3b) and property (6) of a cover, if $M \in P_{\forall(\Phi)}$ is stubborn and $s \in T$, then $Ms \in P_{\Phi(s)}$ is stubborn.

Definition 4.6 Given an algebra of types T and a \mathcal{P} -site $\langle \mathcal{A}, \mathcal{P}, \text{Cov} \rangle$, we let $\text{Sheaf}(\mathcal{A}, \mathcal{P})$ denote the sets of all \mathcal{P} -sheaves on $\langle \mathcal{A}, \mathcal{P}, \text{Cov} \rangle$, and

$$\text{Sheaf}(\mathcal{A}, \mathcal{P})_s = \{S_s \mid S_s \in S, \text{ for some sheaf } \mathcal{S} = (S_s)_{s \in T} \in \text{Sheaf}(\mathcal{A}, \mathcal{P})\}.$$

Since \mathcal{P} itself is a \mathcal{P} -sheaf, the set $\text{Sheaf}(\mathcal{A}, \mathcal{P})$ is nonempty. The fact that definition 4.5 is indeed a sheaf condition is shown exactly as in part I (except that a functor F is a \mathcal{P} -sheaf iff it is a sheaf, and for every $a \in \mathcal{A}$, $F(a) \subseteq T$ and $s \in F(a)$ implies that $a \in P_s$).

5 \mathcal{P} -Realizability For $\lambda \rightarrow, \forall^2$

In this section, we define a semantic notion of realizability. This notion is such that realizers are elements of some pre-applicative structure. Since types can contain type variables, we first need to define an interpretation of the types. In order to define the set of realizers of a second-order type $\forall X. \sigma$, we need to define sheaf-valuations (see definition 5.4).

Definition 5.1 Given an algebra of polymorphic types T , it is assumed that we have a function $TI: TC \rightarrow T$ assigning an element $TI(k) \in T$ to every type constant $k \in TC$. A *type valuation* is a function $\theta: \mathcal{V} \rightarrow T$. Given a type valuation θ , every type $\sigma \in T$ is interpreted as an element $[\sigma]\theta$ of T as follows:

$$\begin{aligned} [X]\theta &= \theta(X), \quad \text{where } X \text{ is a type variable,} \\ [k]\theta &= TI(k), \quad \text{where } k \text{ is a type constant,} \\ [\sigma \rightarrow \tau]\theta &= [\sigma]\theta \rightarrow [\tau]\theta, \\ [\sigma \times \tau]\theta &= [\sigma]\theta \times [\tau]\theta, \\ [\sigma + \tau]\theta &= [\sigma]\theta + [\tau]\theta, \\ [\forall X. \sigma]\theta &= \forall (\lambda t \in T. [\sigma]\theta[X := t]). \end{aligned}$$

In the above definition, $\lambda t \in T. [\sigma]\theta[X := t]$ denotes the function Φ from T to T such that $\Phi(t) = [\sigma]\theta[X := t]$ for every $t \in T$. We say that T is a *type interpretation* iff $\Phi \in [T \rightarrow T]$ for every type σ and every valuation θ .

In other words, T is a type interpretation iff $[\sigma]\theta$ is well-defined for every valuation θ . The following lemmas will be needed later.

Lemma 5.2 For every type $\sigma \in T$, and every pair of type valuations θ_1 and θ_2 , if $\theta_1(X) = \theta_2(X)$, for all $X \in FTV(\sigma)$, then $[\sigma]\theta_1 = [\sigma]\theta_2$.

Proof. A straightforward induction on σ . \square

Lemma 5.3 Given a type interpretation T , for all $\sigma, \tau \in T$, for every type valuation θ , we have

$$[[\sigma[\tau/X]]]\theta = [\sigma]\theta[X := [\tau]\theta].$$

Proof. The proof is by induction on σ . The case where $\sigma = X$ is trivial, since then $X[\tau/X] = \tau$, and

$$[X]\theta[X := [\tau]\theta] = \theta[X := [\tau]\theta](X) = [\tau]\theta.$$

The induction steps are straightforward, and we only treat the case where $\sigma = \forall Y. \sigma_1$. In this case,

$$[[\forall Y. \sigma_1][\tau/X]]\theta = \forall (\Lambda t \in T. [\sigma_1[\tau/X]]\theta[Y := t]),$$

(where the bound variable Y is renamed in a suitable fashion if necessary), and where $\Lambda t \in T$. $[\sigma_1[\tau/X]]\theta[Y := t]$ denotes the function Φ from T to T such that $\Phi(t) = [[\sigma_1[\tau/X]]]\theta[Y := t]$ for every $t \in T$. By the induction hypothesis, we have

$$\Phi(t) = [[\sigma_1[\tau/X]]]\theta[Y := t] = [\sigma_1]\theta[X := [\tau]\theta, Y := t].$$

Then, since

$$[[\forall Y. \sigma_1]\theta[X := [\tau]\theta] = \forall (\Lambda t \in T. [\sigma_1]\theta[X := [\tau]\theta, Y := t]),$$

we have

$$[[\forall Y. \sigma_1][\tau/X]]\theta = [[\forall Y. \sigma_1]\theta[X := [\tau]\theta]].$$

\square

The next definition can be viewed as a semantic version of Girard's "candidats de réductibilité" (see Girard [4], Gallier [2]).

Definition 5.4 Given a type interpretation T and a pre-applicative structure \mathcal{A} , a *sheaf-valuation* is a pair $\mu = \langle \theta, \eta \rangle$, where $\theta: \mathcal{V} \rightarrow T$ is a type valuation, and $\eta: \mathcal{V} \rightarrow \bigcup \text{Sheaf}(\mathcal{A}, \mathcal{P})$ is a function called a *candidate assignment*, such that:

$\eta(X) = S_{\theta(X)}$, where $S_{\theta(X)} \in \text{Sheaf}(\mathcal{A}, \mathcal{P})_{\theta(X)}$, for some \mathcal{P} -sheaf $\mathcal{S} = (S_s)_{s \in T} \in \text{Sheaf}(\mathcal{A}, \mathcal{P})$, for every $X \in \mathcal{V}$.

Given $\mu = \langle \theta, \eta \rangle$, for any $s \in T$ and any $S \in \text{Sheaf}(\mathcal{A}, \mathcal{P})_s$, for some s -component $S = S_s$ of some \mathcal{P} -sheaf $\mathcal{S} = (S_s)_{s \in T} \in \text{Sheaf}(\mathcal{A}, \mathcal{P})$, we let $\mu[X := \langle s, S \rangle] = \langle \theta[X := s], \eta[X := S] \rangle$ be the sheaf-valuation, such that, $\theta[X := s](Y) = \theta(Y)$ for every $Y \neq X$ and $\theta[X := s](X) = s$, and $\eta[X := S](Y) = \eta(Y)$ for all $Y \neq X$, and $\eta[X := S](X) = S$.

The notion of \mathcal{P} -realizability is defined as follows.

Definition 5.5 Given an algebra of types T , let $\langle \mathcal{A}, \mathcal{P}, \text{Cov} \rangle$ be a \mathcal{P} -site. For every sheaf-valuation $\mu = \langle \theta, \eta \rangle$, the family $(\mathbf{r}[\sigma]\mu)_{\sigma \in T}$, where for every $\sigma \in T$, $\mathbf{r}[\sigma]\mu$ is the set of *realizers* of σ , is defined as follows:

$$\begin{aligned} \mathbf{r}[k]\mu &= P_{[k]\theta}, & k \text{ a constant type,} \\ \mathbf{r}[X]\mu &= \eta(X), & X \text{ a type variable,} \\ \mathbf{r}[\sigma \rightarrow \tau]\mu &= \{M \mid M \in P_{[\sigma \rightarrow \tau]\theta}, \text{ and for all } N, \text{ if } N \in \mathbf{r}[\sigma]\mu \text{ then } MN \in \mathbf{r}[\tau]\mu\}, \\ \mathbf{r}[\forall X. \sigma]\mu &= \{M \mid M \in P_{[\forall X. \sigma]\theta}, \text{ and for every } s \in T, \text{ every } S \in \text{Sheaf}(\mathcal{A}, \mathcal{P})_s, \\ & Ms \in \mathbf{r}[\sigma]\mu[X := \langle s, S \rangle]\}. \end{aligned}$$

The following lemmas will be needed later.

Lemma 5.6 For every type $\sigma \in T$, every pair of sheaf-valuations $\mu_1 = \langle \theta_1, \eta_1 \rangle$ and $\mu_2 = \langle \theta_2, \eta_2 \rangle$, if $\theta_1(X) = \theta_2(X)$ and $\eta_1(X) = \eta_2(X)$, for all $X \in FTV(\sigma)$, then $\mathbf{r}[\sigma]\mu_1 = \mathbf{r}[\sigma]\mu_2$.

Proof. A straightforward induction on σ (and using lemma 5.2). \square

Lemma 5.7 Given a type interpretation T and a \mathcal{P} -site $\langle \mathcal{A}, \mathcal{P}, \text{Cov} \rangle$, for all $\sigma, \tau \in T$, for every sheaf-valuation $\mu = \langle \theta, \eta \rangle$, we have

$$\mathbf{r}[\sigma[\tau/X]]\mu = \mathbf{r}[\sigma]\mu[X := \langle [\tau]\theta, \mathbf{r}[\tau]\mu \rangle].$$

Proof. The proof is by induction on σ . We only consider the case where $\sigma = \forall Y. \sigma_1$, the other cases being straightforward. By definition 5.5, we have

$$\begin{aligned} \mathbf{r}[(\forall Y. \sigma_1)[\tau/X]]\mu &= \{M \mid M \in P_{[(\forall Y. \sigma_1)[\tau/X]\theta}, \text{ and for every } s \in T, \text{ every } S \in \text{Sheaf}(\mathcal{A}, \mathcal{P})_s, \\ & Ms \in \mathbf{r}[\sigma_1[\tau/X]]\mu[Y := \langle s, S \rangle]\}. \end{aligned}$$

By lemma 5.3, we have

$$[(\forall Y. \sigma_1)[\tau/X]]\theta = [\forall Y. \sigma_1]\theta[X := [\tau]\theta],$$

and by the induction hypothesis, we have

$$\mathbf{r}[\sigma_1[\tau/X]]\mu[Y := \langle s, S \rangle] = \mathbf{r}[\sigma_1]\mu[Y := \langle s, S \rangle, X := \langle [\tau]\theta, \mathbf{r}[\tau]\mu \rangle].$$

However, by definition,

$$\begin{aligned} \mathbf{r}[\forall Y. \sigma_1]\mu[X := \langle [\tau]\theta, \mathbf{r}[\tau]\mu \rangle] &= \{M \mid M \in P_{[\forall Y. \sigma_1]\theta[X := [\tau]\theta]}, \text{ and for every } s \in T, \\ & \text{every } S \in \text{Sheaf}(\mathcal{A}, \mathcal{P})_s, \\ & Ms \in \mathbf{r}[\sigma_1]\mu[X := \langle [\tau]\theta, \mathbf{r}[\tau]\mu \rangle, Y := \langle s, S \rangle]\}, \end{aligned}$$

and so, we have

$$\mathbf{r}[(\forall Y. \sigma_1)[\tau/X]]\mu = \mathbf{r}[\forall Y. \sigma_1]\mu[X := \langle [\tau]\theta, \mathbf{r}[\tau]\mu \rangle].$$

\square

The following lemma shows that the notion of a \mathcal{P} -sheaf is an inductive invariant. In Gallier [2], this is the lemma we call “Girard’s trick”.¹

¹Of course, this is unfair. Girard has many tricks!

Lemma 5.8 *Given a scenic \mathcal{P} -site $\langle \mathcal{A}, \mathcal{P}, \text{Cov} \rangle$, for every sheaf valuation μ , if \mathcal{P} satisfies conditions (P1)-(P3), then the family $(\mathbf{r}[\sigma]\mu)_{\sigma \in \mathcal{T}}$ is a \mathcal{P} -sheaf, and if $A^{[\sigma]\theta} \neq \emptyset$, then each $\mathbf{r}[\sigma]\mu$ contains all stubborn elements in $P_{[\sigma]\theta}$.*

Proof. We proceed by induction on types. If σ is a base type, $\mathbf{r}[\sigma]\mu = P_{[\sigma]\theta}$, and obviously, every stubborn element in $P_{[\sigma]\theta}$ is in $\mathbf{r}[\sigma]\mu$. Since $\mathbf{r}[\sigma]\mu = P_{[\sigma]\theta}$, (S1) is trivial, (S2) follows from (P2), and (S3) is also trivial. If $\sigma = X$ is a type variable, then $\mathbf{r}[\sigma]\mu = \eta(X)$, and since $\eta(X) = S_{\theta(X)}$, where $S_{\theta(X)} \in \text{Sheaf}(\mathcal{A}, \mathcal{P})_{\theta(X)}$, (S1), (S2), and (S3) hold. The fact that every stubborn element in $P_{\theta(X)}$ is in $S_{\theta(X)}$ follows from (P1) and (S3), as we already noted earlier.

We now consider the induction step.

(S1).

(1) Type $\sigma \rightarrow \tau$. By the definition of $\mathbf{r}[\sigma \rightarrow \tau]\mu$, (S1) is trivial.

(2) Type $\forall X. \sigma$. By the definition of $\mathbf{r}[\forall X. \sigma]\mu$, (S1) is trivial.

(S2).

(1) Type $\sigma \rightarrow \tau$.

Let $M \in \mathbf{r}[\sigma \rightarrow \tau]\mu$, and assume that $M \succeq M'$. Since $M \in P_{[\sigma \rightarrow \tau]\theta}$ by (S1), we have $M' \in P_{[\sigma \rightarrow \tau]\theta}$ by (P2). For any $N \in \mathbf{r}[\sigma]\mu$, since $M \in \mathbf{r}[\sigma \rightarrow \tau]\mu$, we have $MN \in \mathbf{r}[\tau]\mu$, and since $M \succeq M'$, by monotonicity of app , we have $MN \succeq M'N$. Then, applying the induction hypothesis at type τ , (S2) holds for $\mathbf{r}[\tau]\mu$, and thus $M'N \in \mathbf{r}[\tau]\mu$. Thus, we have shown that $M' \in P_{[\sigma \rightarrow \tau]\theta}$ and that if $N \in \mathbf{r}[\sigma]\mu$, then $M'N \in \mathbf{r}[\tau]\mu$. By the definition of $\mathbf{r}[\sigma \rightarrow \tau]\mu$, this shows that $M' \in \mathbf{r}[\sigma \rightarrow \tau]\mu$, and (S2) holds at type $\sigma \rightarrow \tau$.

(2) Type $\forall X. \sigma$.

Let $M \in \mathbf{r}[\forall X. \sigma]\mu$, and assume that $M \succeq M'$. Since $M \in P_{[\forall X. \sigma]\theta}$, by (S1), we have $M' \in P_{[\forall X. \sigma]\theta}$. For every $s \in T$ and every $S \in \text{Sheaf}(\mathcal{A}, \mathcal{P})_s$, since $M \in \mathbf{r}[\forall X. \sigma]\mu$, we have $Ms \in \mathbf{r}[\sigma]\mu[X := \langle s, S \rangle]$, and since $M \succeq M'$, by monotonicity of tapp , we have $Ms \succeq M's$. Then, applying the induction hypothesis to σ and $\mu[X := \langle s, S \rangle]$, (S2) holds for $\mathbf{r}[\sigma]\mu[X := \langle s, S \rangle]$, and thus $M's \in \mathbf{r}[\sigma]\mu[X := \langle s, S \rangle]$. By the definition of $\mathbf{r}[\forall X. \sigma]\mu$, this shows that $M' \in \mathbf{r}[\forall X. \sigma]\mu$.

(S3).

(1) Type $\sigma \rightarrow \tau$.

Assume that $\text{Cov}_{[\sigma \rightarrow \tau]\theta}(C, M)$, and that $M' \in \mathbf{r}[\sigma \rightarrow \tau]\mu$ for every $M' \in C$, where M is simple. Recall that by condition (0) of definition 4.2, $\text{Cov}_{[\sigma \rightarrow \tau]\theta}(C, M)$ implies that $M \in P_{[\sigma \rightarrow \tau]\theta}$. We prove that for every N , if $N \in \mathbf{r}[\sigma]\mu$, then $MN \in \mathbf{r}[\tau]\mu$. First, we prove that $MN \in P_{[\tau]\theta}$, and for this we use (P3).

First, assume that $M \in P_{[\sigma \rightarrow \tau]\theta}$ is stubborn, and let N be in $\mathbf{r}[\sigma]\mu$. By (S1), $N \in P_{[\sigma]\theta}$. By the induction hypothesis, all stubborn elements in $P_{[\tau]\theta}$ are in $\mathbf{r}[\tau]\mu$. Since we showed that $MN \in P_{[\tau]\theta}$ is stubborn whenever $M \in P_{[\sigma \rightarrow \tau]\theta}$ is stubborn and $N \in P_{[\sigma]\theta}$, we have $M \in \mathbf{r}[\sigma \rightarrow \tau]\mu$.

Now, consider $M \in P_{[\sigma \rightarrow \tau]\theta}$ non stubborn. If $M' \in C$, then by assumption, $M' \in \mathbf{r}[\sigma \rightarrow \tau]\mu$, and for any $N \in \mathbf{r}[\sigma]\mu$, we have $M'N \in \mathbf{r}[\tau]\mu$. Since by (S1), $N \in P_{[\sigma]\theta}$ and $M'N \in P_{[\tau]\theta}$, by (P3a), we have $MN \in P_{[\tau]\theta}$. Now, there are two cases.

If τ is a base type, then $\mathbf{r}[\tau]\mu = P_{[\tau]\theta}$ and $MN \in \mathbf{r}[\tau]\mu$.

If τ is not a base type, then MN is simple (since the site is scenic). Thus, we prove that $MN \in \mathbf{r}[\tau]\mu$ using (S3) (which by induction, holds at type τ). Assume that $\mathbf{Cov}_{[\tau]\theta}(D, MN)$ for any cover D of MN . If MN is stubborn, then by the induction hypothesis, we have $MN \in \mathbf{r}[\tau]\mu$. Otherwise, since $\mathbf{Cov}_{[\sigma \rightarrow \tau]\theta}(C, M)$ and C and D are nontrivial, for every $Q \in D$, by condition (5) of definition 4.2, there is some $M' \in C$ such that $Q \preceq M'N$. Since by assumption, $M' \in \mathbf{r}[\sigma \rightarrow \tau]\mu$ whenever $M' \in C$, and $N \in \mathbf{r}[\sigma]\mu$, we conclude that $M'N \in \mathbf{r}[\tau]\mu$. By the induction hypothesis applied at type τ , by (S2), we have $Q \in \mathbf{r}[\tau]\mu$, and by (S3), we have $MN \in \mathbf{r}[\tau]\mu$.

Since $M \in P_{[\sigma \rightarrow \tau]\theta}$ and $MN \in \mathbf{r}[\tau]\mu$ whenever $N \in \mathbf{r}[\sigma]\mu$, we conclude that $M \in \mathbf{r}[\sigma \rightarrow \tau]\mu$.

(2) Type $\forall X. \sigma$.

Assume that $\mathbf{Cov}_{[\forall X. \sigma]\theta}(C, M)$, and that $M' \in \mathbf{r}[\forall X. \sigma]\mu$ for every $M' \in C$, where M is simple. Recall that by condition (0) of definition 4.2, $\mathbf{Cov}_{[\forall X. \sigma]\theta}(C, M)$ implies that $M \in P_{[\forall X. \sigma]\theta}$. We prove that for every $s \in T$ and every $S \in \mathbf{Sheaf}(\mathcal{A}, \mathcal{P})_s$, we have $Ms \in \mathbf{r}[\sigma]\mu[X := \langle s, S \rangle]$. First, we prove that $Ms \in P_{[\sigma]\theta[X := s]}$, and for this, we use (P3).

First, assume that $M \in P_{[\forall X. \sigma]\theta}$ is stubborn, and let $s \in T$. By the induction hypothesis, all stubborn elements in $P_{[\sigma]\theta[X := s]}$ are in $\mathbf{r}[\sigma]\mu[X := \langle s, S \rangle]$. Recall that we have shown that $Ms \in P_{\Phi(s)}$ is stubborn whenever $M \in P_{\forall(\Phi)}$ is stubborn. Considering the function Φ such that $\Phi(s) = [\sigma]\theta[X := s]$ for every $s \in T$, since we know that $[\forall X. \sigma]\theta = \forall(\Phi)$, then $Ms \in P_{[\sigma]\theta[X := s]}$ is stubborn whenever $M \in P_{[\forall X. \sigma]\theta}$ is stubborn, and we have $M \in \mathbf{r}[\forall X. \sigma]\mu$.

Now, consider $M \in P_{[\forall X. \sigma]\theta}$ non stubborn. If $M' \in C$, then by assumption, $M' \in \mathbf{r}[\forall X. \sigma]\mu$, and for every $s \in T$ and every $S \in \mathbf{Sheaf}(\mathcal{A}, \mathcal{P})_s$, we have $M's \in \mathbf{r}[\sigma]\mu[X := \langle s, S \rangle]$. Since by (S1), $M's \in P_{[\sigma]\theta[X := s]}$, by (P3b), we have $M's \in P_{[\sigma]\theta[X := s]}$, where (P3b) is applied to the function Φ such that $\Phi(s) = [\sigma]\theta[X := s]$ for every $s \in T$. For such a Φ , we have $[\forall X. \sigma]\theta = \forall(\Phi)$. Now, there are two cases.

If σ is a base type, then $\mathbf{r}[\sigma]\mu[X := \langle s, S \rangle] = P_{[\sigma]\theta[X := s]}$, and $M's \in \mathbf{r}[\sigma]\mu[X := \langle s, S \rangle]$.

If σ is not a base type, then $M's$ is simple (since the site is scenic). Thus, we prove that $M's \in \mathbf{r}[\sigma]\mu[X := \langle s, S \rangle]$ using (S3) (which by induction, holds for σ). Assume that $\mathbf{Cov}_{[\sigma]\theta[X := s]}(D, M's)$ for any cover D of $M's$. If $M's$ is stubborn, then by the induction hypothesis, we have $M's \in \mathbf{r}[\sigma]\mu[X := \langle s, S \rangle]$. Otherwise, since $\mathbf{Cov}_{[\forall X. \sigma]\theta}(C, M)$ and C and D are nontrivial, for every $Q \in D$, by condition (6) of definition 4.2, there is some $M' \in C$ such that $Q \preceq M's$. Since by assumption, $M' \in \mathbf{r}[\forall X. \sigma]\mu$ whenever $M' \in C$, we conclude that $M's \in \mathbf{r}[\sigma]\mu[X := \langle s, S \rangle]$. By the induction hypothesis applied at type σ , by (S2), we have $Q \in \mathbf{r}[\sigma]\mu[X := \langle s, S \rangle]$, and by (S3), we have $M's \in \mathbf{r}[\sigma]\mu[X := \langle s, S \rangle]$. \square

We will now need to relate λ -terms and realizers.

6 Interpreting λ -Terms in Pre-Applicative Structures

We show how judgements $\Gamma \triangleright M : \sigma$ are interpreted in pre-applicative structures. For this, we define valuations.

Definition 6.1 Given a type interpretation T , given a pre-applicative structure \mathcal{A} , a *valuation* is a pair $\rho = \langle \theta, \epsilon \rangle$, where $\theta: \mathcal{V} \rightarrow T$ is a type valuation, and $\epsilon: \mathcal{X} \rightarrow \bigcup (A^t)_{t \in T}$ is a partial function called an *environment*.

Given $\rho = \langle \theta, \epsilon \rangle$, for any $s \in T$ and $a \in A^s$ we let $\rho[X:=s, x:=a] = \langle \theta[X:=s], \epsilon[x:=a] \rangle$ be the valuation, such that, $\theta[X:=s](Y) = \theta(Y)$ for every $Y \neq X$ and $\theta[X:=s](X) = s$, and $\epsilon[x:=a](y) = \epsilon(y)$ for all $y \neq x$, and $\epsilon[x:=a](x) = a$.

Given a context Γ , we say that ρ *satisfies* Γ , written as $\rho \models \Gamma$ (where $\rho = \langle \theta, \epsilon \rangle$) iff

$$\epsilon(x) \in A^{[\sigma]^\theta} \quad \text{for every } x: \sigma \in \Gamma.$$

Note that if ρ satisfies a context Γ , this implies that $A^{[\sigma]^\theta} \neq \emptyset$ for every $x: \sigma \in \Gamma$. Also, conditions (1)-(4) of definition 3.3 imply that the following conditions hold:

For all types $\sigma, \tau \in T$, if $A^{[\sigma]^\theta} \neq \emptyset$ and $A^{[\tau]^\theta} \neq \emptyset$, then $A^{[\sigma \rightarrow \tau]^\theta} \neq \emptyset$, $A^{[\sigma \times \tau]^\theta} \neq \emptyset$, $A^{[\sigma + \tau]^\theta} \neq \emptyset$, and if $A^{[\sigma[\tau/X]]^\theta} \neq \emptyset$ for every $\tau \in T$, then $A^{[\forall X. \sigma]^\theta} \neq \emptyset$.

We are now ready to interpret λ -terms.

Definition 6.2 Given a type interpretation T and a pre-applicative structure \mathcal{A} , let $\mathcal{AI}: \text{Const} \rightarrow \mathcal{A}$ be a function assigning an element $\mathcal{AI}(c)$ of $A^{TI(\text{Type}(c))}$ to every constant $c \in \text{Const}$. For every valuation $\rho = \langle \theta, \epsilon \rangle$, and every context Γ , if $\rho \models \Gamma$, we define the *interpretation* (or *meaning*) $\mathcal{A}[\Gamma \triangleright M: \sigma] \rho$ of a judgement $\Gamma \triangleright M: \sigma$ inductively as follows:

$$\mathcal{A}[\Gamma \triangleright x: \sigma] \rho = \epsilon(x)$$

$$\mathcal{A}[\Gamma \triangleright c: \text{Type}(c)] \rho = \mathcal{AI}(c)$$

$$\mathcal{A}[\Gamma \triangleright MN: \tau] \rho = \text{app}^{[\sigma]^\theta, [\tau]^\theta}(\mathcal{A}[\Gamma \triangleright M: (\sigma \rightarrow \tau)] \rho, \mathcal{A}[\Gamma \triangleright N: \sigma] \rho)$$

$$\mathcal{A}[\Gamma \triangleright \lambda x: \sigma. M: (\sigma \rightarrow \tau)] \rho = \text{abst}^{[\sigma]^\theta, [\tau]^\theta}(\varphi),$$

where φ is the function defined such that,

$$\varphi(a) = \mathcal{A}[\Gamma, x: \sigma \triangleright M: \tau] \rho[x:=a], \text{ for every } a \in A^{[\sigma]^\theta}$$

$$\mathcal{A}[\Gamma \triangleright M\tau: \sigma[\tau/X]] \rho = \text{tapp}^\Phi(\mathcal{A}[\Gamma \triangleright M: \forall X. \sigma] \rho, [\tau]^\theta),$$

where Φ is the function such that $\Phi(s) = [\sigma]^\theta[X:=s]$ for every $s \in T$

$$\mathcal{A}[\Gamma \triangleright \lambda X. M: \forall X. \sigma] \rho = \text{tabst}^\Phi(\varphi),$$

where φ is the function defined such that,

$$\varphi(s) = \mathcal{A}[\Gamma, X: \star \triangleright M: \sigma] \rho[X:=s], \text{ for every } s \in T, \text{ and where } \Phi \text{ is the function such that } \Phi(s) = [\sigma]^\theta[X:=s] \text{ for every } s \in T$$

$$\mathcal{A}[\Gamma \triangleright \pi_1(M): \sigma] \rho = \pi_1(\mathcal{A}[\Gamma \triangleright M: \sigma \times \tau] \rho)$$

$$\mathcal{A}[\Gamma \triangleright \pi_2(M): \tau] \rho = \pi_2(\mathcal{A}[\Gamma \triangleright M: \sigma \times \tau] \rho)$$

$$\mathcal{A}[\Gamma \triangleright \langle M_1, M_2 \rangle: \sigma \times \tau] \rho = \langle \mathcal{A}[\Gamma \triangleright M_1: \sigma] \rho, \mathcal{A}[\Gamma \triangleright M_2: \tau] \rho \rangle$$

$$\mathcal{A}[\Gamma \triangleright \text{inl}(M): \sigma] \rho = \text{inl}(\mathcal{A}[\Gamma \triangleright M: \sigma + \tau] \rho)$$

$$\mathcal{A}[\Gamma \triangleright \text{inr}(M): \tau] \rho = \text{inr}(\mathcal{A}[\Gamma \triangleright M: \sigma + \tau] \rho)$$

$$\mathcal{A}[\Gamma \triangleright [M, N]:(\sigma + \tau) \rightarrow \delta]\rho = [\mathcal{A}[\Gamma \triangleright M:(\sigma \rightarrow \delta)]\rho, \mathcal{A}[\Gamma \triangleright N:(\tau \rightarrow \delta)]\rho].$$

We are assuming that $\langle -, - \rangle$ and $[-, -]$ are total, and that the domains of **abst** and **tabst** are sufficiently large for the above definitions to be well-defined for all ρ , and $\Gamma \triangleright M:\sigma$. In this case, we say that \mathcal{A} is a *pre-interpretation*.

The following lemma will be needed later.

Lemma 6.3 *For every pair of contexts Γ_1 and Γ_2 , for every pair of valuations $\rho_1 = \langle \theta_1, \epsilon_1 \rangle$ and $\rho_2 = \langle \theta_2, \epsilon_2 \rangle$, for every pair of judgements $\Gamma_1 \triangleright M:\sigma$ and $\Gamma_2 \triangleright M:\sigma$, if $\rho_1 \models \Gamma_1$ and $\rho_2 \models \Gamma_2$, $\Gamma_1(x) = \Gamma_2(x)$, for all $x \in FV(M)$, $\theta_1(X) = \theta_2(X)$, for all $X \in \bigcup (FTV(\tau))_{x:\tau \in \Gamma} \cup FTV(M)$, and $\epsilon_1(x) = \epsilon_2(x)$, for all $x \in FV(M)$, then*

$$\mathcal{A}[\Gamma_1 \triangleright M:\sigma]\rho_1 = \mathcal{A}[\Gamma_2 \triangleright M:\sigma]\rho_2.$$

Proof. A straightforward induction on typing derivations (and using lemma 5.2). \square

Let us give an (important) example of a pre-applicative structure. First, we review the notion of a substitution.

Definition 6.4 A substitution φ is a function $\varphi: \mathcal{V} \cup \mathcal{X} \rightarrow \mathcal{T} \cup \text{Terms}$, such that $\varphi(X) \in \mathcal{T}$ if $X \in \mathcal{V}$, $\varphi(x) \in \text{Terms}$ if $x \in \mathcal{X}$, and $\varphi(x) \neq x$ only for finitely many variables. We let $\text{dom}(\varphi) = \{x \in \mathcal{V} \cup \mathcal{X} \mid \varphi(x) \neq x\}$. We say that φ is a *type-substitution* if $\text{dom}(\varphi) \subseteq \mathcal{V}$. Given two contexts Γ and Δ , we say that φ *satisfies Γ at Δ* , denoted as $\Delta \Vdash \Gamma[\varphi]$, iff $\Delta \triangleright \varphi(x):\sigma[\varphi]$, for every $x:\sigma \in \Gamma$.

The following definition shows how the term model can be viewed as a pre-applicative β -structure.

Definition 6.5 The algebra of second-order types T is defined as follows:

$$T = \{\langle \sigma, \Gamma \rangle \mid \sigma \in \mathcal{T}, \Gamma \text{ a context}\} \cup \{\mathbf{error}\}.$$

The operations \rightarrow , \times , and $+$, are defined as follows:

$$a \rightarrow b = \langle \sigma \rightarrow \tau, \Gamma \rangle \text{ iff } a = \langle \sigma, \Gamma \rangle, b = \langle \tau, \Delta \rangle, \text{ and } \Gamma = \Delta, \text{ otherwise } \mathbf{error};$$

$$a \times b = \langle \sigma \times \tau, \Gamma \rangle \text{ iff } a = \langle \sigma, \Gamma \rangle, b = \langle \tau, \Delta \rangle, \text{ and } \Gamma = \Delta, \text{ otherwise } \mathbf{error};$$

$$a + b = \langle \sigma + \tau, \Gamma \rangle \text{ iff } a = \langle \sigma, \Gamma \rangle, b = \langle \tau, \Delta \rangle, \text{ and } \Gamma = \Delta, \text{ otherwise } \mathbf{error}.$$

We let $A^{\mathbf{error}} = \emptyset$, and $A^{(\sigma, \Gamma)}$ be the set of all provable typing judgements of the form $\Gamma \triangleright M:\sigma$. We denote $A^{(\sigma, \Gamma)}$ as A_Γ^σ . For $[T \Rightarrow T]$, we take the set of all functions Φ such that $\langle \tau, \Gamma \rangle \mapsto \langle \sigma[\tau/X], \Gamma \rangle$, where $\sigma, \tau \in \mathcal{T}$ are any types, and X is any fixed variable that does not occur in Γ (and with $\mathbf{error} \mapsto \mathbf{error}$). Then, $\forall(\Phi) = \langle \forall X. \sigma, \Gamma \rangle$.²

A type valuation is a function $\theta: \mathcal{V} \rightarrow T$, such that $\theta(X) = \langle \sigma_X, \Gamma_X \rangle$ or $\theta(X) = \mathbf{error}$ for every $X \in \mathcal{V}$, and such that the function $X \mapsto \sigma_X$ defines an (infinite) type substitution that we denote as $[\theta]$. Then, for any type $\sigma \in \mathcal{T}$, by the definition of the operations \rightarrow , \times , and $+$, either

²The choice of X is irrelevant as long as X does not occur in Γ , since X is bound in $\forall X. \sigma$.

$[\sigma]\theta = \mathbf{error}$, or $[\sigma]\theta = \langle \sigma[\theta], \Delta \rangle$ for some context Δ . A valuation $\rho = \langle \theta, \epsilon \rangle$ consists of a type valuation θ and of a partial function $\epsilon: \mathcal{X} \rightarrow \bigcup (A^\sigma)_{\sigma \in T}$. As noted just after definition 6.1, the conditions on θ require that there is some single Δ such that, $\theta(X) = \langle \sigma_X, \Delta \rangle$ iff $A_\Delta^{\sigma_X} \neq \emptyset$, for every $X \in \mathcal{V}$, and $\theta(c) = \langle \sigma_c, \Delta \rangle$ iff $A_\Delta^{\sigma_c} \neq \emptyset$, for every type constant c .³

Indeed, if $\theta(X_1) = \langle \sigma_1, \Delta_1 \rangle$, $\theta(X_2) = \langle \sigma_2, \Delta_2 \rangle$, $A_{\Delta_1}^{\sigma_1} \neq \emptyset$, $A_{\Delta_2}^{\sigma_2} \neq \emptyset$, $X_1 \neq X_2$, and $\Delta_1 \neq \Delta_2$, since $\langle \sigma_1, \Delta_1 \rangle \rightarrow \langle \sigma_2, \Delta_2 \rangle = \mathbf{error}$ and $A^{\mathbf{error}} = \emptyset$, the condition on θ would be violated. Thus, ϵ is a partial function such that $\epsilon(x)$ is of the form $\epsilon(x) = \Delta \triangleright M_x: \sigma_x$, when it is defined (where Δ is uniquely determined by θ).

Given a context Γ , according to definition 6.1, a valuation $\rho = \langle \theta, \epsilon \rangle$ satisfies Γ ($\rho \models \Gamma$) iff for every $x_i: \sigma_i \in \Gamma$, we have $\epsilon(x_i) \in A_\Delta^{\sigma_i[\theta]}$, for the fixed context Δ determined by θ , as explained above. This means that $\epsilon(x_i) = \Delta \triangleright M_i: \sigma_i[\theta]$, for some M_i . A valuation $\rho = \langle \theta, \epsilon \rangle$ such that $\rho \models \Gamma$ defines a substitution $[\epsilon]: \mathcal{X} \rightarrow \mathbf{Terms}$, such that $[\epsilon](x) = M_x$, where $\epsilon(x) = \Delta \triangleright M_x: \sigma[\theta]$, for every $x: \sigma \in \Gamma$.

Thus, the restriction of ρ to Γ defines a substitution φ as follows: $\varphi(x) = [\epsilon](x)$ for every $x \in \mathit{dom}(\Gamma)$, and $\varphi(X) = [\theta](X)$ for every $X \in \bigcup_{\sigma \in \Gamma} \mathit{FTV}(\sigma)$. Also, $\rho \models \Gamma$ is just the condition $\Delta \Vdash \Gamma[\varphi]$ of definition 6.4, where Δ is the context uniquely determined by θ .

We let Π , $\langle -, - \rangle$, inl , inr , and $[-, -]$, be the obvious. For example, $\langle \Gamma \triangleright M_1: \sigma, \Gamma \triangleright M_2: \tau \rangle = \Gamma \triangleright \langle M_1, M_2 \rangle: \sigma \times \tau$. Define $\Gamma \triangleright N: \sigma \preceq \Gamma \triangleright M: \sigma$ iff $M \xrightarrow{*}_\beta N$. Finally, we need to define fun , abst , tfun , and tabst .

We define $\mathit{fun}(\Gamma \triangleright M: \sigma \rightarrow \tau)$ as the function $[\Gamma \triangleright M: \sigma \rightarrow \tau]$ from A_Γ^σ to A_Γ^τ , such that

$$[\Gamma \triangleright M: \sigma \rightarrow \tau](\Gamma \triangleright N: \sigma) = \Gamma \triangleright MN: \tau,$$

for every $\Gamma \triangleright N: \sigma \in A_\Gamma^\sigma$.

We define $\mathit{tfun}(\Gamma \triangleright M: \forall X. \sigma)$ as the function $[\Gamma \triangleright M: \forall X. \sigma]$ from T to $\bigsqcup (A_\Gamma^\sigma)_{\sigma \in T}$, such that

$$[\Gamma \triangleright M: \forall X. \sigma](\tau) = \Gamma \triangleright M\tau: \sigma[\tau/X],$$

for every $\tau \in T$. In this case, the Φ in tfun^Φ is the function from T to T induced by σ , such that $\Phi(\tau) = \sigma[\tau/X]$ for every $\tau \in T$.

For every pair of contexts Γ, Δ , for every substitution φ such that $\Delta \Vdash (\Gamma, x: \sigma)[\varphi]$, for every judgement $\Gamma, x: \sigma \triangleright M: \tau$, consider the function $\varphi[\Gamma, x: \sigma \triangleright M: \tau]_\Delta$ from $A_\Delta^{\sigma[\varphi]}$ to $A_\Delta^{\tau[\varphi]}$, defined such that,

$$\varphi[\Gamma, x: \sigma \triangleright M: \tau]_\Delta(\Delta \triangleright N: \sigma[\varphi]) = \Delta \triangleright M[\varphi[x = N]]: \tau[\varphi],$$

for every $\Delta \triangleright N: \sigma[\varphi] \in A_\Delta^{\sigma[\varphi]}$. Given any such function $\varphi[\Gamma, x: \sigma \triangleright M: \tau]_\Delta$, we let

$$\mathit{abst}(\varphi[\Gamma, x: \sigma \triangleright M: \tau]_\Delta) = \Delta \triangleright (\lambda x: \sigma. M)[\varphi]: \sigma[\varphi] \rightarrow \tau[\varphi].$$

For every pair of contexts Γ, Δ , for every substitution φ such that $\Delta \Vdash (\Gamma, X: \star)[\varphi]$, for every judgement $\Gamma, X: \star \triangleright M: \sigma$, consider the function $\varphi[\Gamma, X: \star \triangleright M: \sigma]_\Delta$ from T to $\bigsqcup (A_\Delta^\sigma)_{\sigma \in T}$, defined such that,

$$\varphi[\Gamma, X: \star \triangleright M: \sigma]_\Delta(\tau) = \Delta \triangleright M[\varphi[X = \tau]]: \sigma[\varphi[X = \tau]],$$

³ $A_\Delta^\sigma = \emptyset$ when there is no provable judgement $\Delta \triangleright M: \sigma$ for any M .

for every $\tau \in T$.

Given any such function $\varphi[\Gamma, X: \star \triangleright M: \sigma]_{\Delta}$, we let

$$\mathbf{tabst}(\varphi[\Gamma, X: \star \triangleright M: \sigma]_{\Delta}) = \Delta \triangleright (\lambda X. M)[\varphi]: \forall X. \sigma[\varphi].$$

The pre-applicative β -structure just defined is denoted as \mathcal{LT}_{β} .

It is clear that $\varphi[\Gamma, x: \sigma \triangleright M: \tau]_{\Delta}$ is in $[A^{\sigma[\varphi]} \Rightarrow A^{\tau[\varphi]}]_{\Delta}$. Let us verify that

$$\mathbf{fun}(\mathbf{abst}(\varphi[\Gamma, x: \sigma \triangleright M: \tau]_{\Delta})) \succeq \varphi[\Gamma, x: \sigma \triangleright M: \tau]_{\Delta}.$$

Since

$$\begin{aligned} \mathbf{fun}(\mathbf{abst}(\varphi[\Gamma, x: \sigma \triangleright M: \tau]_{\Delta})) &= \mathbf{fun}(\Delta \triangleright (\lambda x: \sigma. M)[\varphi]: \sigma[\varphi] \rightarrow \tau[\varphi]), \\ \mathbf{fun}(\Delta \triangleright (\lambda x: \sigma. M)[\varphi]: \sigma[\varphi] \rightarrow \tau[\varphi]) &= [\Delta \triangleright (\lambda x: \sigma. M)[\varphi]: \sigma[\varphi] \rightarrow \tau[\varphi]], \\ [\Delta \triangleright (\lambda x: \sigma. M)[\varphi]: \sigma[\varphi] \rightarrow \tau[\varphi]](\Delta \triangleright N: \sigma[\varphi]) &= \Delta \triangleright ((\lambda x: \sigma. M)[\varphi])N: \tau[\varphi], \\ \varphi[\Gamma, x: \sigma \triangleright M: \tau]_{\Delta}(\Delta \triangleright N: \sigma[\varphi]) &= \Delta \triangleright M[\varphi[x = N]]: \tau[\varphi], \end{aligned}$$

and

$$((\lambda x: \sigma. M)[\varphi])N \longrightarrow_{\beta} M[\varphi[x = N]],$$

the inequality holds. Indeed, $(\lambda x: \sigma. M)[\varphi]$ is α -equivalent to $(\lambda y: \sigma. M[y/x])[\varphi]$ for any variable y such that $y \notin \text{dom}(\varphi)$ and $y \notin \varphi(z)$ for every $z \in \text{dom}(\varphi)$, and for such a y , $(\lambda y: \sigma. M[y/x])[\varphi] = (\lambda y: \sigma[\varphi]. M[y/x][\varphi])$. Then, for this choice of y ,

$$(\lambda y: \sigma[\varphi]. M[y/x][\varphi])N \longrightarrow_{\beta} M[y/x][\varphi][N/y] = M[\varphi[x = N]].$$

Regarding the definition of \mathbf{tabst} , letting Φ be the function from T to T induced by σ , such that $\Phi(\tau) = \sigma[\tau/X]$ for every $\tau \in T$, it is clear that $\varphi[\Gamma, X: \star \triangleright M: \sigma]_{\Delta}$ is in $\prod_{\Phi}(A_{\Delta}^{\sigma})_{s \in T}$. Let us now verify that

$$\mathbf{tfun}(\mathbf{tabst}(\varphi[\Gamma, X: \star \triangleright M: \sigma]_{\Delta})) \succeq \varphi[\Gamma, X: \star \triangleright M: \sigma]_{\Delta}.$$

Since

$$\begin{aligned} \mathbf{tfun}(\mathbf{tabst}(\varphi[\Gamma, X: \star \triangleright M: \sigma]_{\Delta})) &= \mathbf{tfun}(\Delta \triangleright (\lambda X. M)[\varphi]: \forall X. \sigma[\varphi]), \\ \mathbf{tfun}(\Delta \triangleright (\lambda X. M)[\varphi]: \forall X. \sigma[\varphi]) &= [\Delta \triangleright (\lambda X. M)[\varphi]: \forall X. \sigma[\varphi]], \\ [\Delta \triangleright (\lambda X. M)[\varphi]: \forall X. \sigma[\varphi]](\tau) &= \Delta \triangleright ((\lambda X. M)[\varphi])\tau: \sigma[\varphi][\tau/X], \\ \varphi[\Gamma, X: \star \triangleright M: \sigma]_{\Delta}(\tau) &= \Delta \triangleright M[\varphi[X = \tau]]: \sigma[\varphi[X = \tau]], \\ \sigma[\varphi][\tau/X] &= \sigma[\varphi[X = \tau]], \end{aligned}$$

(by a suitable α -renaming on X), and

$$((\lambda X. M)[\varphi])\tau \longrightarrow_{\beta} M[\varphi[X = \tau]],$$

the inequality holds (the details of the verification using α -renaming are similar to the previous case).

The other conditions of definition 3.3 are easily verified.

As we already observed, a valuation $\rho = \langle \theta, \epsilon \rangle$ for the pre-applicative structure \mathcal{LT}_β , is characterized by a single context Δ such that, $\theta(X) = \langle \sigma_X, \Delta \rangle$ iff $A_\Delta^{\sigma_X} \neq \emptyset$, and $\theta(c) = \langle \sigma_c, \Delta \rangle$ iff $A_\Delta^{\sigma_c} \neq \emptyset$, for every type constant, and ϵ is a partial function such that $\epsilon(x)$ is of the form $\epsilon(x) = \Delta \triangleright M_x : \sigma_x$, when it is defined. Also, given a context Γ , a valuation $\rho = \langle \theta, \epsilon \rangle$ satisfies Γ ($\rho \models \Gamma$) iff $\Delta \Vdash \Gamma[\varphi]$. Then, by a simple induction on the typing derivation for $\Gamma \triangleright M : \sigma$, we can show that for any valuation $\rho = \langle \theta, \epsilon \rangle$ such that $\rho \models \Gamma$, then

$$\mathcal{LT}_\beta[\Gamma \triangleright M : \sigma]\rho = \Delta \triangleright M[\varphi] : \sigma[\varphi],$$

where Δ is uniquely determined by θ , and where φ is the substitution defined by the restriction of $\rho = \langle \theta, \epsilon \rangle$ to Γ , as explained at the beginning of definition 6.5.

We now show how the structure \mathcal{LT}_β of definition 6.5 can be made into a pre-applicative $\beta\eta$ -structure. First, we recall the η -like rules.

Definition 6.6 The set of η -like reduction rules is defined as follows.

$$\begin{aligned} \lambda x : \sigma. (Mx) &\longrightarrow M, & \text{if } x \notin FV(M), \\ \lambda X. (MX) &\longrightarrow M, & \text{if } X \notin FTV(M), \\ \langle \pi_1(M), \pi_2(M) \rangle &\longrightarrow M, \\ [\lambda x : \sigma. (Minl(x)), \lambda y : \tau. (Minr(y))] &\longrightarrow M. \end{aligned}$$

We will denote the reduction relation defined by the union of the rules of definition 2.3 and of definition 6.6 as $\longrightarrow_{\beta\eta}$ (even though there are reductions other than β -reduction and η -reduction).

Definition 6.7 We define a pre-applicative structure as in definition 6.5, except that $\Gamma \triangleright M : \sigma \preceq \Gamma \triangleright N : \sigma$ iff $N \xrightarrow{*}_{\beta\eta} M$, and that **abst** and **tabst** have a larger domain of definition. First, recall the definition of **fun** and **tfun**.

fun($\Gamma \triangleright M : \sigma \rightarrow \tau$) is defined as the function $[\Gamma \triangleright M : \sigma \rightarrow \tau]$ from A_Γ^σ to A_Γ^τ , such that

$$[\Gamma \triangleright M : \sigma \rightarrow \tau](\Gamma \triangleright N : \sigma) = \Gamma \triangleright MN : \tau,$$

for every $\Gamma \triangleright N : \sigma \in A_\Gamma^\sigma$.

tfun($\Gamma \triangleright M : \forall X. \sigma$) is defined as the function $[\Gamma \triangleright M : \forall X. \sigma]$ from T to $\coprod_{\sigma \in T} (A_\Gamma^\sigma)$, such that

$$[\Gamma \triangleright M : \forall X. \sigma](\tau) = \Gamma \triangleright M\tau : \sigma[\tau/X],$$

for every $\tau \in T$. In this case, the Φ in **tfun** $^\Phi$ is the function from T to T induced by σ , such that $\Phi(\tau) = \sigma[\tau/X]$ for every $\tau \in T$.

Then, we define

$$\mathbf{abst}_\Gamma([\Gamma \triangleright M : \sigma \rightarrow \tau]) = \Gamma \triangleright \lambda x : \sigma. (Mx) : \sigma \rightarrow \tau,$$

where $x \notin FV(M)$, and

$$\mathbf{tabst}_\Gamma([\Gamma \triangleright M : \forall X. \sigma]) = \Gamma \triangleright \lambda X. (MX) : \forall X. \sigma,$$

where $X \notin FTV(M)$. The structure just defined is denoted as $\mathcal{LT}_{\beta\eta}$.

We need to check that $\mathcal{LT}_{\beta\eta}$ is a pre-applicative $\beta\eta$ -structure. Let us first verify that

$$\mathbf{fun}_\Gamma(\mathbf{abst}_\Gamma([\Gamma \triangleright M : \sigma \rightarrow \tau])) \succeq [\Gamma \triangleright M : \sigma \rightarrow \tau].$$

Since

$$\begin{aligned} \mathbf{fun}_\Gamma(\mathbf{abst}_\Gamma([\Gamma \triangleright M : \sigma \rightarrow \tau])) &= \mathbf{fun}_\Gamma(\Gamma \triangleright \lambda x : \sigma. (Mx) : \sigma \rightarrow \tau), \\ \mathbf{fun}_\Gamma(\Gamma \triangleright \lambda x : \sigma. (Mx) : \sigma \rightarrow \tau) &= [\Gamma \triangleright \lambda x : \sigma. (Mx) : \sigma \rightarrow \tau], \\ [\Gamma \triangleright \lambda x : \sigma. (Mx) : \sigma \rightarrow \tau](\Delta \triangleright N : \sigma) &= \Delta \triangleright (\lambda x : \sigma. (Mx))N : \tau, \\ [\Gamma \triangleright M : \sigma \rightarrow \tau](\Delta \triangleright N : \sigma) &= \Delta \triangleright MN : \tau, \end{aligned}$$

and

$$(\lambda x : \sigma. (Mx))N \longrightarrow_\beta MN,$$

since $x \notin FV(M)$, the inequality holds.

Let us now verify that

$$\mathbf{tfun}_\Gamma(\mathbf{tabst}_\Gamma([\Gamma \triangleright M : \forall X. \sigma])) \succeq [\Gamma \triangleright M : \forall X. \sigma].$$

Since

$$\begin{aligned} \mathbf{tfun}_\Gamma(\mathbf{tabst}_\Gamma([\Gamma \triangleright M : \forall X. \sigma])) &= \mathbf{tfun}_\Gamma(\Gamma \triangleright \lambda X. (MX) : \forall X. \sigma), \\ \mathbf{tfun}_\Gamma(\Gamma \triangleright \lambda X. (MX) : \forall X. \sigma) &= [\Gamma \triangleright \lambda X. (MX) : \forall X. \sigma], \\ [\Gamma \triangleright \lambda X. (MX) : \forall X. \sigma](\tau) &= \Delta \triangleright (\lambda X. (MX))\tau : \sigma[\tau/X], \\ [\Gamma \triangleright M : \forall X. \sigma](\tau) &= \Delta \triangleright M\tau : \sigma[\tau/X], \end{aligned}$$

and

$$(\lambda X. (MX))\tau \longrightarrow_\beta M\tau,$$

since $X \notin FTV(M)$, the inequality holds.

We also need to verify the conditions of definition 3.5.

We have $\mathbf{abst}_\Gamma(\mathbf{fun}_\Gamma(\Gamma \triangleright M : \sigma \rightarrow \tau)) = \mathbf{abst}_\Gamma([\Gamma \triangleright M : \sigma \rightarrow \tau])$, and since

$$\mathbf{abst}_\Gamma([\Gamma \triangleright M : \sigma \rightarrow \tau]) = \Gamma \triangleright \lambda x : \sigma. (Mx) : \sigma \rightarrow \tau,$$

where $x \notin FV(M)$, and by the η -like rule, $\lambda x : \sigma. (Mx) \longrightarrow_{\beta\eta} M$, we have

$$\mathbf{abst}_\Gamma(\mathbf{fun}_\Gamma(\Gamma \triangleright M : \sigma \rightarrow \tau)) \succeq \Gamma \triangleright M : \sigma \rightarrow \tau.$$

Similarly, we have $\mathbf{tabst}_\Gamma(\mathbf{tfun}_\Gamma(\Gamma \triangleright M : \forall X. \sigma)) = \mathbf{tabst}_\Gamma([\Gamma \triangleright M : \forall X. \sigma])$, and since

$$\mathbf{tabst}_\Gamma([\Gamma \triangleright M : \forall X. \sigma]) = \Gamma \triangleright \lambda X. (MX) : \forall X. \sigma,$$

where $X \notin FTV(M)$, and by the η -like rule, $\lambda X. (MX) \longrightarrow_{\beta\eta} M$, we have

$$\mathbf{tabst}_\Gamma(\mathbf{tfun}_\Gamma(\Gamma \triangleright M : \forall X. \sigma)) \succeq \Gamma \triangleright M : \forall X. \sigma.$$

The other conditions of definition 3.5, are immediately verified.

7 The Realizability Theorem

In this section, we prove the realizability lemma (lemma 7.6) for $\lambda \rightarrow, \forall^2$, and its main corollary, theorem 7.7. First, we need some conditions relating the behavior of a meaning function and covering conditions. We will also need semantic conditions analogous to the conditions (P4)-(P5) from the introduction of part I.

Definition 7.1 We say that a site $\langle \mathcal{A}, \mathcal{P}, \text{Cov} \rangle$ is well-behaved iff the following conditions hold:

- (1) For any $a \in A^s$, any $\varphi \in [A^s \Rightarrow A^t]$, if $\text{abst}(\varphi)$ exists, $\text{Cov}_t(C, \text{app}(\text{abst}(\varphi), a))$, and C is a nontrivial cover, then $c \preceq \varphi(a)$ for every $c \in C$.
- (2) For any $s \in T$, any $\varphi \in \prod_{\Phi}(A^s)_{s \in T}$, if $\text{tabst}(\varphi)$ exists, $\text{Cov}_{\Phi(s)}(C, \text{tapp}(\text{tabst}(\varphi), s))$, and C is a nontrivial cover, then $c \preceq \varphi(s)$ for every $c \in C$.

In view of definition 6.2, definition 7.1 implies the following condition.

Definition 7.2

- (1) For any $a \in A^{[\sigma]^\tau}$, if $\text{Cov}_{[\tau]^\theta}(C, \text{app}(\mathcal{A}[\Gamma \triangleright \lambda x: \sigma. M: (\sigma \rightarrow \tau)]\rho, a))$ and C is a nontrivial cover, then $c \preceq \mathcal{A}[\Gamma, x: \sigma \triangleright M: \tau]\rho[x := a]$ for every $c \in C$.
- (2) For any $s \in T$, if $\text{Cov}_{[\sigma]^\theta[X:=s]}(C, \text{tapp}(\mathcal{A}[\Gamma \triangleright \lambda X. M: \forall X. \sigma]\rho, s))$ and C is a nontrivial cover, then $c \preceq \mathcal{A}[\Gamma, X: \star \triangleright M: \sigma]\rho[X := s]$ for every $c \in C$.

For the proof of the next lemma, we need to add two new conditions (P4) and (P5) to (P1)-(P3).

Definition 7.3 Given a well-behaved site $\langle \mathcal{A}, \mathcal{P}, \text{Cov} \rangle$, properties (P4) and (P5) are defined as follows:

- (P4a) For every $a \in A^s$, if $\varphi(a) \in P_t$, where $\varphi \in [A^s \Rightarrow A^t]$ and $\text{abst}(\varphi)$ exists, then $\text{abst}(\varphi) \in P_{s \rightarrow t}$.
- (P4b) For every $s \in T$, if $\varphi(s) \in P_{\Phi(s)}$, where $\varphi \in \prod_{\Phi}(A^s)_{s \in T}$ and $\text{tabst}(\varphi)$ exists, then $\text{tabst}(\varphi) \in P_{\mathcal{V}(\Phi)}$.
- (P5a) If $a \in P_s$ and $\varphi(a) \in P_t$, where $\varphi \in [A^s \Rightarrow A^t]$ and $\text{abst}(\varphi)$ exists, then $\text{app}(\text{abst}(\varphi), a) \in P_t$.
- (P5b) If $s \in T$ and $\varphi(s) \in P_{\Phi(s)}$, where $\varphi \in \prod_{\Phi}(A^s)_{s \in T}$ and $\text{tabst}(\varphi)$ exists, then $\text{tapp}(\text{tabst}(\varphi), s) \in P_{\Phi(s)}$.

In view of definition 6.2, definition 7.3 implies the following conditions.

Definition 7.4

- (P4a) If $\mathcal{A}[\Gamma, x: \sigma \triangleright M: \tau]\rho \in P_{[\tau]^\theta}$, then $\mathcal{A}[\Gamma \triangleright \lambda x: \sigma. M: (\sigma \rightarrow \tau)]\rho \in P_{[\sigma \rightarrow \tau]^\theta}$.
- (P4b) If $\mathcal{A}[\Gamma, X: \star \triangleright M: \sigma]\rho \in P_{[\sigma]^\theta}$, then $\mathcal{A}[\Gamma \triangleright \lambda X. M: \forall X. \sigma]\rho \in P_{[\forall X. \sigma]^\theta}$.
- (P5a) If $a \in P_{[\sigma]^\theta}$ and $\mathcal{A}[\Gamma, x: \sigma \triangleright M: \tau]\rho[x := a] \in P_{[\tau]^\theta}$, then $\text{app}(\mathcal{A}[\Gamma \triangleright \lambda x: \sigma. M: (\sigma \rightarrow \tau)]\rho, a) \in P_{[\tau]^\theta}$.
- (P5b) If $s \in T$ and $\mathcal{A}[\Gamma, X: \star \triangleright M: \sigma]\rho[X := s] \in P_{[\sigma]^\theta[X:=s]}$, then $\text{tapp}(\mathcal{A}[\Gamma \triangleright \lambda X. M: \forall X. \sigma]\rho, s) \in P_{[\sigma]^\theta[X:=s]}$.

Lemma 7.5 *Given a well-behaved scenic site $\langle \mathcal{A}, \mathcal{P}, \text{Cov} \rangle$ and a family \mathcal{P} satisfying conditions (P1)-(P5), for every sheaf valuation $\mu = \langle \theta, \eta \rangle$ and every valuation $\rho = \langle \theta, \epsilon \rangle$ sharing the same type valuation θ , for every context Γ , if $\rho \models \Gamma$, then the following properties hold: (1) If $\rho(y) \in \mathbf{r}[\delta]\mu$ for every $y: \delta \in \Gamma, x: \sigma$, if for every a , ($a \in \mathbf{r}[\sigma]\mu$ implies $\mathcal{A}[\Gamma, x: \sigma \triangleright M: \tau]\rho[x = a] \in \mathbf{r}[\tau]\mu$), then $\mathcal{A}[\Gamma \triangleright \lambda x: \sigma. M: (\sigma \rightarrow \tau)]\rho \in \mathbf{r}[\sigma \rightarrow \tau]\mu$;*

(2) If $\mathcal{A}[\Gamma, X: \star \triangleright M: \sigma]\rho[X = s] \in \mathbf{r}[\sigma]\mu[X = \langle s, S \rangle]$, for every $s \in T$ and every $S \in \text{Sheaf}(\mathcal{A}, \mathcal{P})_s$, then $\mathcal{A}[\Gamma \triangleright \lambda X. M: \forall X. \sigma]\rho \in \mathbf{r}[\forall X. \sigma]\mu$.

Proof. (1) We prove that $\mathcal{A}[\Gamma \triangleright \lambda x: \sigma. M: (\sigma \rightarrow \tau)]\rho \in P_{[\sigma \rightarrow \tau]\theta}$, and that for every every a , if $a \in \mathbf{r}[\sigma]\mu$, then $\text{app}(\mathcal{A}[\Gamma \triangleright \lambda x: \sigma. M: (\sigma \rightarrow \tau)]\rho, a) \in \mathbf{r}[\tau]\mu$. We will need the fact that the sets of the form $\mathbf{r}[\sigma]\mu$ have the properties (S1)-(S3), but this follows from lemma 5.8, since (P1)-(P3) hold. First, we prove that $\mathcal{A}[\Gamma \triangleright \lambda x: \sigma. M: (\sigma \rightarrow \tau)]\rho \in P_{[\sigma \rightarrow \tau]\theta}$.

Since $\rho(y) \in \mathbf{r}[\delta]\mu$ for every $y: \delta \in \Gamma, x: \sigma$, letting $a = \rho(x)$, by the assumption of lemma 7.5, $\mathcal{A}[\Gamma, x: \sigma \triangleright M: \tau]\rho \in \mathbf{r}[\tau]\mu$. Then, by (S1), we have $\mathcal{A}[\Gamma, x: \sigma \triangleright M: \tau]\rho \in P_{[\tau]\theta}$, and by (P4a), we have $\mathcal{A}[\Gamma \triangleright \lambda x: \sigma. M: (\sigma \rightarrow \tau)]\rho \in P_{[\sigma \rightarrow \tau]\theta}$.

Next, we prove that for every every a , if $a \in \mathbf{r}[\sigma]\mu$, then $\text{app}(\mathcal{A}[\Gamma \triangleright \lambda x: \sigma. M: (\sigma \rightarrow \tau)]\rho, a) \in \mathbf{r}[\tau]\mu$. Assume that $a \in \mathbf{r}[\sigma]\mu$. Then, by the assumption of lemma 7.5, $\mathcal{A}[\Gamma, x: \sigma \triangleright M: \tau]\rho[x = a] \in \mathbf{r}[\tau]\mu$. Thus, by (S1), we have $a \in P_{[\sigma]\theta}$ and $\mathcal{A}[\Gamma, x: \sigma \triangleright M: \tau]\rho[x = a] \in P_{[\tau]\theta}$. By (P5a), we have $\text{app}(\mathcal{A}[\Gamma \triangleright \lambda x: \sigma. M: (\sigma \rightarrow \tau)]\rho, a) \in P_{[\tau]\theta}$. Now, there are two cases.

If τ is a base type, then $\mathbf{r}[\tau]\mu = P_{[\tau]\theta}$. Since $\text{app}(\mathcal{A}[\Gamma \triangleright \lambda x: \sigma. M: (\sigma \rightarrow \tau)]\rho, a) \in P_{[\tau]\theta}$, we have $\text{app}(\mathcal{A}[\Gamma \triangleright \lambda x: \sigma. M: (\sigma \rightarrow \tau)]\rho, a) \in \mathbf{r}[\tau]\mu$.

If τ is not a base type, then $\text{app}(\mathcal{A}[\Gamma \triangleright \lambda x: \sigma. M: (\sigma \rightarrow \tau)]\rho, a)$ is simple (since the site is scenic). Thus, we prove that $\text{app}(\mathcal{A}[\Gamma \triangleright \lambda x: \sigma. M: (\sigma \rightarrow \tau)]\rho, a) \in \mathbf{r}[\tau]\mu$ using (S3). By lemma 5.8, the case where $\text{app}(\mathcal{A}[\Gamma \triangleright \lambda x: \sigma. M: (\sigma \rightarrow \tau)]\rho, a)$ is stubborn is trivial.

Otherwise, assume that $\text{Cov}_{[\tau]\theta}(C, \text{app}(\mathcal{A}[\Gamma \triangleright \lambda x: \sigma. M: (\sigma \rightarrow \tau)]\rho, a))$, where C is a nontrivial cover. By condition (1) of definition 7.2, $c \preceq \mathcal{A}[\Gamma, x: \sigma \triangleright M: \tau]\rho[x = a]$ for every $c \in C$, and since by assumption, $\mathcal{A}[\Gamma, x: \sigma \triangleright M: \tau]\rho[x = a] \in \mathbf{r}[\tau]\mu$, by (S2), we have $c \in \mathbf{r}[\tau]\mu$. Since $c \in \mathbf{r}[\tau]\mu$ whenever $c \in C$, by (S3), we have $\text{app}(\mathcal{A}[\Gamma \triangleright \lambda x: \sigma. M: (\sigma \rightarrow \tau)]\rho, a) \in \mathbf{r}[\tau]\mu$.

(2) We prove that $\mathcal{A}[\Gamma \triangleright \lambda X. M: \forall X. \sigma]\rho \in P_{[\forall X. \sigma]\theta}$, and that for every $s \in T$ and every $S \in \text{Sheaf}(\mathcal{A}, \mathcal{P})_s$, $\text{tapp}(\mathcal{A}[\Gamma \triangleright \lambda X. M: \forall X. \sigma]\rho, s) \in \mathbf{r}[\sigma]\mu[X = \langle s, S \rangle]$. By lemma 5.8, since (P1)-(P3) hold, the sets of the form $\mathbf{r}[\sigma]\mu[X = \langle s, S \rangle]$ have the properties (S1)-(S3). First, we prove that $\mathcal{A}[\Gamma \triangleright \lambda X. M: \forall X. \sigma]\rho \in P_{[\forall X. \sigma]\theta}$.

By the assumption of lemma 7.5, $\mathcal{A}[\Gamma, X: \star \triangleright M: \sigma]\rho \in \mathbf{r}[\sigma]\mu[X = \langle s, S \rangle]$ for every $s \in T$ and every $S \in \text{Sheaf}(\mathcal{A}, \mathcal{P})_s$. In particular, this holds for $s = \theta(X)$ and $S = \eta(X)$, and we have $\mathcal{A}[\Gamma, X: \star \triangleright M: \sigma]\rho \in \mathbf{r}[\sigma]\mu$. Then, by (S1), we have $\mathcal{A}[\Gamma, X: \star \triangleright M: \sigma]\rho \in P_{[\sigma]\theta}$, and by (P4b), we have $\mathcal{A}[\Gamma \triangleright \lambda X. M: \forall X. \sigma]\rho \in P_{[\forall X. \sigma]\theta}$.

Next, we prove that $\text{tapp}(\mathcal{A}[\Gamma \triangleright \lambda X. M: \forall X. \sigma]\rho, s) \in \mathbf{r}[\sigma]\mu[X = \langle s, S \rangle]$, for every $s \in T$ and every $S \in \text{Sheaf}(\mathcal{A}, \mathcal{P})_s$. By the assumption of lemma 7.5, $\mathcal{A}[\Gamma, X: \star \triangleright M: \sigma]\rho[X = s] \in \mathbf{r}[\sigma]\mu[X = \langle s, S \rangle]$. Thus, by (S1), we have $\mathcal{A}[\Gamma, X: \star \triangleright M: \sigma]\rho[X = s] \in P_{[\sigma]\theta[X = s]}$. By (P5b), we have $\text{tapp}(\mathcal{A}[\Gamma \triangleright \lambda X. M: \forall X. \sigma]\rho, s) \in P_{[\sigma]\theta[X = s]}$. Now, there are two cases.

If σ is a base type, then $\mathbf{r}[\sigma]\mu[X = \langle s, S \rangle] = P_{[\sigma]\theta[X = s]}$. Since $\text{tapp}(\mathcal{A}[\Gamma \triangleright \lambda X. M: \forall X. \sigma]\rho, s) \in P_{[\sigma]\theta[X = s]}$, we have $\text{tapp}(\mathcal{A}[\Gamma \triangleright \lambda X. M: \forall X. \sigma]\rho, s) \in \mathbf{r}[\sigma]\mu[X = \langle s, S \rangle]$.

If σ is not a base type, then $\mathbf{tapp}(\mathcal{A}[\Gamma \triangleright \lambda X. M: \forall X. \sigma]\rho, s)$ is simple (since the site is scenic). Thus, we prove that $\mathbf{tapp}(\mathcal{A}[\Gamma \triangleright \lambda X. M: \forall X. \sigma]\rho, s) \in \mathbf{r}[\sigma]\mu[X: = \langle s, S \rangle]$, using (S3). The case where $\mathbf{tapp}(\mathcal{A}[\Gamma \triangleright \lambda X. M: \forall X. \sigma]\rho, s)$ is stubborn is trivial.

Otherwise, assume that $\mathbf{Cov}_{[\sigma]\theta[X:=s]}(C, \mathbf{tapp}(\mathcal{A}[\Gamma \triangleright \lambda X. M: \forall X. \sigma]\rho, s))$, where C is a nontrivial cover. By condition (2) of definition 7.2, $c \preceq \mathcal{A}[\Gamma, X: \star \triangleright M: \sigma]\rho[X: = s]$ for every $c \in C$, and since by assumption, $\mathcal{A}[\Gamma, X: \star \triangleright M: \sigma]\rho[X: = s] \in \mathbf{r}[\sigma]\mu[X: = \langle s, S \rangle]$, by (S2), we have $c \in \mathbf{r}[\sigma]\mu[X: = \langle s, S \rangle]$. Since $c \in \mathbf{r}[\sigma]\mu[X: = \langle s, S \rangle]$ whenever $c \in C$, we deduce using (S3) that we have $\mathbf{tapp}(\mathcal{A}[\Gamma \triangleright \lambda X. M: \forall X. \sigma]\rho, s) \in \mathbf{r}[\sigma]\mu[X: = \langle s, S \rangle]$. \square

We now prove the main realizability lemma for $\lambda \rightarrow, \forall^2$.

Lemma 7.6 *Given a well-behaved scenic site $\langle \mathcal{A}, \mathcal{P}, \mathbf{Cov} \rangle$ and a family \mathcal{P} satisfying conditions (P1)-(P5), for every sheaf valuation $\mu = \langle \theta, \eta \rangle$ and every valuation $\rho = \langle \theta, \epsilon \rangle$ sharing the same type valuation θ , for every context Γ , if $\rho \models \Gamma$ and $\rho(y) \in \mathbf{r}[\delta]\mu$ for every $y: \delta \in \Gamma$, then for every $\Gamma \triangleright M: \sigma$, we have $\mathcal{A}[\Gamma \triangleright M: \sigma]\rho \in \mathbf{r}[\sigma]\mu$.*

Proof. We proceed by induction on the derivation of $\Gamma \triangleright M: \sigma$. If M is a variable x , then $\mathcal{A}[\Gamma \triangleright x: \sigma]\rho = \epsilon(x) \in \mathbf{r}[\sigma]\mu$, by the assumption on ρ .

If $M = M_1 N_1$, where $\Gamma \triangleright M_1: (\sigma \rightarrow \tau)$ and $\Gamma \triangleright N_1: \sigma$, by the induction hypothesis,

$$\mathcal{A}[\Gamma \triangleright M_1: (\sigma \rightarrow \tau)]\rho \in \mathbf{r}[\sigma \rightarrow \tau]\mu \quad \text{and} \quad \mathcal{A}[\Gamma \triangleright N_1: \sigma]\rho \in \mathbf{r}[\sigma]\mu.$$

By the definition of $\mathbf{r}[\sigma \rightarrow \tau]\mu$, we get $\mathbf{app}(\mathcal{A}[\Gamma \triangleright M_1: (\sigma \rightarrow \tau)]\rho, \mathcal{A}[\Gamma \triangleright N_1: \sigma]\rho) \in \mathbf{r}[\tau]\mu$, i.e., $\mathcal{A}[\Gamma \triangleright (M_1 N_1): \tau]\rho \in \mathbf{r}[\tau]\mu$, by definition 6.2.

If $M = \lambda x: \sigma. M_1$, where $\Gamma \triangleright \lambda x: \sigma. M_1: (\sigma \rightarrow \tau)$, consider any $a \in \mathbf{r}[\sigma]\mu$ and any valuation ρ such that $\rho(y) \in \mathbf{r}[\delta]\mu$ for every $y: \delta \in \Gamma$. Note that by (S3) and (P1), $\mathbf{r}[\sigma]\mu$ is indeed nonempty. Thus, the valuation $\rho[x: = a]$ has the property that $\rho[x: = a](y) \in \mathbf{r}[\delta]\mu$ for every $y: \delta \in \Gamma, x: \sigma$. Applying the induction hypothesis to $\Gamma, x: \sigma \triangleright M_1: \tau$ and to the valuations μ , and $\rho[x: = a]$, we have

$$\mathcal{A}[\Gamma, x: \sigma \triangleright M_1: \tau]\rho[x: = a] \in \mathbf{r}[\tau]\mu.$$

Since this holds for every $a \in \mathbf{r}[\sigma]\mu$, by lemma 7.5 (1), $\mathcal{A}[\Gamma \triangleright \lambda x: \sigma. M_1: (\sigma \rightarrow \tau)]\rho \in \mathbf{r}[\sigma \rightarrow \tau]\mu$.

If $M = M_1 \tau$, where $\Gamma \triangleright M_1 \tau: \sigma[\tau/X]$ and $\Gamma \triangleright M_1: \forall X. \sigma$, by the induction hypothesis,

$$\mathcal{A}[\Gamma \triangleright M_1: \forall X. \sigma]\rho \in \mathbf{r}[\forall X. \sigma]\mu.$$

By the definition of $\mathbf{r}[\forall X. \sigma]\mu$, letting $s = [\tau]\theta$ and $S = \mathbf{r}[\tau]\mu$, we get

$$\mathbf{tapp}(\mathcal{A}[\Gamma \triangleright M_1: \forall X. \sigma]\rho, [\tau]\theta) \in \mathbf{r}[\sigma]\mu[X: = \langle s, S \rangle].$$

However, by lemma 5.7, we have

$$\mathbf{r}[\sigma[\tau/X]]\mu = \mathbf{r}[\sigma]\mu[X: = \langle [\tau]\theta, \mathbf{r}[\tau]\mu \rangle],$$

which is just

$$\mathbf{r}[\sigma[\tau/X]]\mu = \mathbf{r}[\sigma]\mu[X: = \langle s, S \rangle],$$

since $s = \llbracket \tau \rrbracket \theta$ and $S = \mathbf{r}[\llbracket \tau \rrbracket \mu]$, and thus, by definition 6.2, we have $\mathcal{A}[\Gamma \triangleright (M_1 \tau) : \sigma[\tau/X]] \rho \in \mathbf{r}[\sigma[\tau/X]] \mu$.

If $M = \lambda X. M_1$, where $\Gamma \triangleright \lambda X. M_1 : \forall X. \sigma$, consider any arbitrary $s \in T$ and any arbitrary $S \in \text{Sheaf}(\mathcal{A}, \mathcal{P})_s$. Since $X \notin \text{dom}(\Gamma)$, by lemma 5.6, we have $\mathbf{r}[\delta] \mu = \mathbf{r}[\delta] \mu[X := \langle s, S \rangle]$ for every $y: \delta \in (\Gamma, X : \star)$. Thus, we can apply the induction hypothesis to $\Gamma, X : \star \triangleright M_1 : \sigma$, and to the valuations $\mu[X := \langle s, S \rangle]$ and ρ , and we have

$$\mathcal{A}[\Gamma, X : \star \triangleright M_1 : \sigma] \rho \in \mathbf{r}[\sigma] \mu[X := \langle s, S \rangle].$$

Since the above holds for every $s \in T$ and every $S \in \text{Sheaf}(\mathcal{A}, \mathcal{P})_s$, by lemma 7.5 (2), we have $\mathcal{A}[\Gamma \triangleright \lambda X. M_1 : \forall X. \sigma] \rho \in \mathbf{r}[\forall X. \sigma] \mu$. \square

If M is a closed term of type σ , lemma 6.3 implies that $\mathcal{A}[\triangleright M : \sigma] \rho$ is independent of ρ , and thus we denote it as $\mathcal{A}[M : \sigma]$. We obtain the following important theorem for $\lambda^{\rightarrow, \forall^2}$.

Theorem 7.7 *Given a well-behaved scenic site $\langle \mathcal{A}, \mathcal{P}, \text{Cov} \rangle$ and a family \mathcal{P} satisfying conditions (P1)-(P5), for every judgement $\triangleright M : \sigma$ where M is closed, we have $\mathcal{A}[M : \sigma] \in P_{[\sigma] \theta}$. (in other words, the realizer $\mathcal{A}[M : \sigma]$ satisfies the unary predicate defined by \mathcal{P} , i.e., every provable type is realizable).*

Proof. Apply lemma 7.6 to the judgement $\triangleright M : \sigma$, to any sheaf valuation $\mu = \langle \theta, \eta \rangle$ such that $\eta(X) = P_{\theta(X)}$ for every $X \in \mathcal{V}$, and to any valuation ρ . \square

8 Applications to the System $\lambda^{\rightarrow, \forall^2}$

This section shows that theorem 7.7 can be used to prove a general theorem about terms of the system $\lambda^{\rightarrow, \forall^2}$. As a corollary, it can be shown that all terms of $\lambda^{\rightarrow, \forall^2}$ are strongly normalizing and confluent.

In order to apply theorem 7.7, we define a notion of cover for the site \mathcal{A} whose underlying pre-applicative structure is the structure \mathcal{LT}_β .

Definition 8.1 An *I-term* is a term of the form either $\lambda x : \sigma. M$, $\lambda X. M$, $\langle M, N \rangle$, $\text{inl}(M)$, $\text{inr}(M)$ or a copairing term $[M, N]$. A *simple* term (or *neutral* term) is a term that is not an I-term. Thus, a simple term is either a variable x , a constant c , an application MN , a type application $M\tau$, a projection $\pi_1(M)$ or $\pi_2(M)$. A term M is *stubborn* iff it is simple and, either M is irreducible, or M' is a simple term whenever $M \xrightarrow{+}_\beta M'$ (equivalently, M' is **not** an I-term).

We define a cover algebra on the structure \mathcal{LT}_β as follows. Let \mathcal{P} be a (unary) property of typed second-order λ -terms.

Definition 8.2 The cover algebra Cov is defined as follows:

- (1) If $\Gamma \triangleright M : \sigma \in P_{(\sigma, \Gamma)}$ and M is an I-term, then

$$\text{Cov}(\Gamma \triangleright M : \sigma) = \{ \{ \Gamma \triangleright N : \sigma \mid M \xrightarrow{*}_\beta N \} \}.$$

(2) If $\Gamma \triangleright M: \sigma \in P_{\langle \sigma, \Gamma \rangle}$ and M is a (simple and) stubborn term, then

$$\text{Cov}(\Gamma \triangleright M: \sigma) = \{\emptyset, \{\Gamma \triangleright N: \sigma \mid M \xrightarrow{*}_{\beta} N\}\}.$$

(3) If $\Gamma \triangleright M: \sigma \in P_{\langle \sigma, \Gamma \rangle}$ and M is a simple and non-stubborn term, then

$$\text{Cov}(\Gamma \triangleright M: \sigma) = \{\{\Gamma \triangleright N: \sigma \mid M \xrightarrow{*}_{\beta} N\}, \{\Gamma \triangleright N: \sigma \mid M \xrightarrow{*}_{\beta} Q \xrightarrow{*}_{\beta} N, \text{ for some I-term } Q\}\}.$$

Recall from definition 4.3 that M is simple iff it has at least two distinct covers. Thus, definition 8.2 implies that a term is simple in the sense of definition 8.1 iff it is simple in the sense of definition 4.3. Similarly a term is stubborn in the sense of definition 8.1 iff it is stubborn in the sense of definition 4.3. Also, definition 8.1 implies that \mathcal{LT}_{β} is scenic.

Properties (P1-P3) are listed below.

Definition 8.3 Properties (P1)-(P3) are defined as follows:

(P1) $\Gamma, x: \sigma \triangleright x: \sigma \in P_{\langle \sigma, \Gamma \rangle}$, $\Gamma \triangleright c: \sigma \in P_{\langle \sigma, \Gamma \rangle}$, for every variable x and constant c (such that $\text{Type}(c) = \sigma$).

(P2) If $\Gamma \triangleright M: \sigma \in P_{\langle \sigma, \Gamma \rangle}$ and $M \longrightarrow_{\beta} N$, then $\Gamma \triangleright N: \sigma \in P_{\langle \sigma, \Gamma \rangle}$.

If M is simple, then:

(P3a) If $\Gamma \triangleright M: (\sigma \rightarrow \tau) \in P_{\langle \sigma \rightarrow \tau, \Gamma \rangle}$, $\Gamma \triangleright N: \sigma \in P_{\langle \sigma, \Gamma \rangle}$, $\Gamma \triangleright (\lambda x: \sigma. M')N: \tau \in P_{\langle \tau, \Gamma \rangle}$ whenever $M \xrightarrow{+}_{\beta} \lambda x: \sigma. M'$, then $\Gamma \triangleright MN: \tau \in P_{\langle \tau, \Gamma \rangle}$.

(P3b) If $\Gamma \triangleright M: \forall X. \sigma \in P_{\langle \forall X. \sigma, \Gamma \rangle}$, $\tau \in \mathcal{T}$, $\Gamma \triangleright (\lambda X. M')\tau: \sigma[\tau/X] \in P_{\langle \sigma[\tau/X], \Gamma \rangle}$ whenever $M \xrightarrow{+}_{\beta} \lambda X. M'$, then $\Gamma \triangleright M\tau: \sigma[\tau/X] \in P_{\langle \sigma[\tau/X], \Gamma \rangle}$.

A careful reader will notice that conditions (P3) of definition 8.3 are not simply a reformulation of conditions (P3) of definition 4.4. This is because according to definition 8.2, $\Gamma \triangleright M: \sigma$, where M is a non-stubborn term, is covered by the nontrivial cover $\{\Gamma \triangleright N: \sigma \mid M \xrightarrow{*}_{\beta} Q \xrightarrow{*}_{\beta} N\}$, where Q is some I-term, but the conditions of definition 8.3 only involve reductions to I-terms. However, due to condition (P2) and the fact that a nontrivial cover is determined by the I-terms in it, the two definitions are indeed equivalent.

If $\Gamma \triangleright M: (\sigma \rightarrow \tau) \in P_{\langle \sigma \rightarrow \tau, \Gamma \rangle}$ where M is a stubborn term and $\Gamma \triangleright N: \sigma \in P_{\langle \sigma, \Gamma \rangle}$ where N is any term, then $\Gamma \triangleright MN: \tau \in P_{\langle \tau, \Gamma \rangle}$ by (P3a). Furthermore, MN is also stubborn since it is a simple term and since it can only reduce to an I-term if M itself reduces to a an I-term. Thus, if $\Gamma \triangleright M: (\sigma \rightarrow \tau) \in P_{\langle \sigma \rightarrow \tau, \Gamma \rangle}$ where M is a stubborn term and $\Gamma \triangleright N: \sigma \in P_{\langle \sigma, \Gamma \rangle}$ where N is any term, then $\Gamma \triangleright MN: \tau \in P_{\langle \tau, \Gamma \rangle}$ where MN is a stubborn term. We can show in a similar fashion that (P3b) implies that if $\Gamma \triangleright M: \forall X. \sigma \in P_{\langle \forall X. \sigma, \Gamma \rangle}$ where M is a stubborn term, then $\Gamma \triangleright M\tau: \sigma[\tau/X] \in P_{\langle \sigma[\tau/X], \Gamma \rangle}$, where $M\tau$ is a stubborn term, for any $\tau \in \mathcal{T}$.

Properties (P4-P5) are listed below.

Definition 8.4 Properties (P4) and (P5) are defined as follows:

(P4a) If $\Gamma, x: \sigma \triangleright M: \tau \in P_{\langle \tau, \Gamma \rangle}$, then $\Gamma \triangleright \lambda x: \sigma. M: (\sigma \rightarrow \tau) \in P_{\langle \sigma \rightarrow \tau, \Gamma \rangle}$.

(P4b) If $\Gamma, X: \star \triangleright M: \sigma \in P_{\langle \sigma, \Gamma \rangle}$, then $\Gamma \triangleright \lambda X. M: \forall X. \sigma \in P_{\langle \forall X. \sigma, \Gamma \rangle}$.

(P5a) If $\Gamma \triangleright N: \sigma \in P_{\langle \sigma, \Gamma \rangle}$ and $\Gamma \triangleright M[N/x]: \tau \in P_{\langle \tau, \Gamma \rangle}$, then $\Gamma \triangleright (\lambda x: \sigma. M)N: \tau \in P_{\langle \tau, \Gamma \rangle}$.

(P5b) If $\tau \in T$ and $\Gamma \triangleright M[\tau/X]: \sigma[\tau/X] \in P_{\langle \sigma[\tau/X], \Gamma \rangle}$, then $(\Gamma \triangleright \lambda X. M)\tau: \sigma[\tau/X] \in P_{\langle \sigma[\tau/X], \Gamma \rangle}$.

Again, a careful reader will notice that conditions (P5) of definition 8.4 are not simply a reformulation of conditions (P5) of definition 7.4. However, because of (P2) and the fact that a nontrivial cover is determined by the I-terms in it, the two sets of conditions are equivalent.

We now show that the conditions of definition 4.2 and the conditions of definition 7.2 hold.

Lemma 8.5 *Definition 8.2 defines a cover algebra, and the site $\langle \mathcal{LT}_\beta, \mathcal{P}, \text{Cov} \rangle$ is scenic and well-behaved.*

Proof. Conditions (0)-(4) of definition 4.2 are easily verified. Let us verify conditions (5) and (6). In these proofs, we often drop the context Γ to simplify the notation.

(5) If $\text{Cov}(M) = \{\emptyset, \{Q \mid Q \preceq M\}\}$ then $\text{Cov}(MN) = \{\emptyset, \{Q \mid Q \preceq MN\}\}$, and if $\text{Cov}(C, M)$ and $\text{Cov}(D, MN)$ with C and D nontrivial, then for every $Q \in D$, there is some $M' \in C$ such that $Q \preceq M'N$.

The first part says that if M is stubborn, then MN is stubborn, which has already been verified. If the covers C and D are nontrivial, then by definition 8.1, M and MN must be simple and non-stubborn terms. In this case, $Q \in D$ means that

$$MN \xrightarrow{+}_\beta P \xrightarrow{*}_\beta Q,$$

where P is an I-term. This can happen only if $M \xrightarrow{+}_\beta M'$, where M' itself an I-term. In this case, there is some reduction

$$MN \xrightarrow{+}_\beta M'N \xrightarrow{*}_\beta P \xrightarrow{*}_\beta Q,$$

where M' is an I-term. Since M is simple and non-stubborn, definition 8.1 implies that $M' \in C$.

(6) If $\text{Cov}(M) = \{\emptyset, \{Q \mid Q \preceq M\}\}$ then $\text{Cov}(Ms) = \{\emptyset, \{Q \mid Q \preceq Ms\}\}$, where $s \in T$, and if $\text{Cov}(C, M)$ and $\text{Cov}(D, Ms)$ with C and D nontrivial, then for every $Q \in D$, there is some $M' \in C$ such that $Q \preceq M's$.

The first part says that if M is stubborn, then $M\tau$ is stubborn, which has already been verified. If the covers C and D are nontrivial, then by definition 8.1, M and MN must be simple and non-stubborn terms. In this case, $Q \in D$ means that

$$M\tau \xrightarrow{+}_\beta P \xrightarrow{*}_\beta Q,$$

where P is an I-term. This can happen only if $M \xrightarrow{+}_\beta M'$, where M' itself an I-term. In this case, there is some reduction

$$M\tau \xrightarrow{+}_\beta M'\tau \xrightarrow{*}_\beta P \xrightarrow{*}_\beta Q,$$

where M' is an I-term. Since M is simple and non-stubborn, definition 8.1 implies that $M' \in C$.

Let us now verify the conditions of definition 7.2. First, recall that for the structure \mathcal{LT}_β , for every valuation $\rho = \langle \theta, \epsilon \rangle$ such that $\rho \models \Gamma$, there is some Δ uniquely determined by θ , such that $\Delta \Vdash \Gamma[\varphi]$, and

$$\mathcal{LT}_\beta[\Gamma \triangleright M : \sigma]\rho = \Delta \triangleright M[\varphi] : \sigma[\varphi],$$

where φ is the substitution defined by the restriction of $\rho = \langle \theta, \epsilon \rangle$ to Γ .

- (1) For any $a \in A^{[\sigma]\tau}$, if $\text{Cov}_{[\tau]\theta}(C, \text{app}(\mathcal{A}[\Gamma \triangleright \lambda x : \sigma. M : (\sigma \rightarrow \tau)]\rho, a))$ and C is a nontrivial cover, then $c \preceq \mathcal{A}[\Gamma, x : \sigma \triangleright M : \tau]\rho[x = a]$ for every $c \in C$.

We have $\text{app}(\mathcal{A}[\Gamma \triangleright \lambda x : \sigma. M : (\sigma \rightarrow \tau)]\rho, a) = \Delta \triangleright ((\lambda x : \sigma. M)[\varphi])a : \tau[\varphi]$, where φ is the substitution defined by the restriction of ρ to Γ . By definition 8.1, since C is nontrivial, $c \in C$ means that

$$((\lambda x : \sigma. M)[\varphi])a \xrightarrow{+}_\beta Q \xrightarrow{*}_\beta c,$$

for some I-term Q . This can only happen if there is a reduction

$$((\lambda x : \sigma. M)[\varphi])a \longrightarrow_\beta (M[\varphi])[a/x] \xrightarrow{*}_\beta c.$$

However, we have $(M[\varphi])[a/x] = M[\varphi[x = a]]$ (using a suitable renaming of x). By the definition of $\mathcal{LT}_\beta[\Gamma, x : \sigma \triangleright M : \tau]\rho$, we have $\mathcal{LT}_\beta[\Gamma, x : \sigma \triangleright M : \tau]\rho[x = a] = \Delta \triangleright M[\varphi[x = a]] : \tau[\varphi]$, and this part of the proof is complete.

- (2) For any $s \in \mathcal{T}$, if $\text{Cov}_{[\sigma]\theta[X:=s]}(C, \text{tapp}(\mathcal{A}[\Gamma \triangleright \lambda X. M : \forall X. \sigma]\rho, s))$ and C is a nontrivial cover, then $c \preceq \mathcal{A}[\Gamma, X : \star \triangleright M : \sigma]\rho[X = s]$ for every $c \in C$.

We have $\text{tapp}(\mathcal{A}[\Gamma \triangleright \lambda X. M : \forall X. \sigma]\rho, s) = \Delta \triangleright ((\lambda X. M)[\varphi])s : (\sigma[s/X])[\varphi]$, where φ is the substitution defined by the restriction of ρ to Γ . By definition 8.1, since C is nontrivial, $c \in C$ means that

$$((\lambda X. M)[\varphi])s \xrightarrow{+}_\beta Q \xrightarrow{*}_\beta c,$$

for some I-term Q . This can only happen if there is a reduction

$$((\lambda X. M)[\varphi])s \longrightarrow_\beta (M[\varphi])[s/X] \xrightarrow{*}_\beta c.$$

However, we have $(M[\varphi])[s/X] = M[\varphi[X = s]]$, and $(\sigma[s/X])[\varphi] = \sigma[\varphi[X = s]]$, (using a suitable renaming of X). By the definition of $\mathcal{LT}_\beta[\Gamma, X : \star \triangleright M : \sigma]\rho$, we have

$$\mathcal{LT}_\beta[\Gamma, X : \star \triangleright M : \sigma]\rho[X = s] = \Delta \triangleright M[\varphi[X = s]] : \tau[\varphi[X = s]],$$

and the proof is complete. \square

Thus, the site $\langle \mathcal{LT}_\beta, \mathcal{P}, \text{Cov} \rangle$, is scenic and well-behaved. Consequently, we can apply theorem 7.7, and get a general theorem for proving properties of terms of the system $\lambda^{\rightarrow, \forall^2}$. In fact, for the structure \mathcal{LT}_β , for a property \mathcal{P} satisfying conditions (P1)-(P5), by (P1) and (P3), every variable x is stubborn. Thus, for every context Γ , we can apply lemma 7.6 to the sheaf valuation $\mu = \langle \theta, \eta \rangle$ such that $\theta(X) = \langle X, \Gamma \rangle$ and $\eta(X) = P_X$ for every type variable, and to the valuation $\rho = \langle \theta, \epsilon \rangle$ such that $\epsilon(x) = x$ for every variable x , since by lemma 5.8, $\mathbf{r}[\delta]\mu$ contains every stubborn term, for every $x : \delta \in \Gamma$. Consequently, we have the following theorem.

Theorem 8.6 *If \mathcal{P} is a family of λ -terms satisfying conditions (P1)-(P5), then $P_{\langle\sigma, \Gamma\rangle} = \Lambda_{\langle\sigma, \Gamma\rangle}$ for every type σ (in other words, every term satisfies the unary predicate defined by \mathcal{P}).*

Proof. By lemma 8.5, the site $\langle \mathcal{L}T_\beta, \mathcal{P}, \mathbf{Cov} \rangle$ is scenic and well-behaved. By the discussion just before stating theorem 8.6, for every context Γ , if we consider the sheaf valuation $\rho = \langle \theta, \eta \rangle$ such that $\theta(X) = \langle X, \Gamma \rangle$ and $\eta(X) = P_X$ for every type variable, and the valuation $\rho = \langle \theta, \epsilon \rangle$ such that $\epsilon(x) = x$ for every variable x , we have $\rho(x) \in \mathbf{r}[\sigma]\mu$ for every $x: \delta \in \Gamma$. Thus, we can apply lemma 7.6 to any judgement $\Gamma \triangleright M: \sigma$ and to μ and ρ just defined, and we have

$$\mathcal{L}T_\beta[\Gamma \triangleright M: \sigma]\rho \in \mathbf{r}[\sigma]\mu.$$

However, in the present case, $\mathcal{L}T_\beta[\Gamma \triangleright M: \sigma]\rho = \Gamma \triangleright M: \sigma$. Thus, $\Gamma \triangleright M: \sigma \in \mathbf{r}[\sigma]\mu$, and since $\mathbf{r}[\sigma]\mu \subseteq P_{\langle\sigma, \Gamma\rangle}$, we have $\Gamma \triangleright M: \sigma \in P_{\langle\sigma, \Gamma\rangle}$, as claimed. \square

As a corollary, we can prove strong normalization and confluence. We prove strong normalization below. For simplicity of notation, instead of using judgements $\Gamma \triangleright M: \sigma$, we will use the terms M . Since we are concerned with reduction properties, this is not harmful at all.

Theorem 8.7 *The reduction relation $\xrightarrow{*}_\beta$ of the system λ^{-, \forall^2} is strongly normalizing.*

Proof. Let \mathcal{P} be the family defined such that $P_\sigma = SN_\sigma$ is the set of strongly normalizing terms of type σ . By theorem 8.6, we just have to check that \mathcal{P} satisfies conditions (P1)-(P5). First, we make the following observation that will simplify the proof. Since there is only a finite number of redexes in any term, for any term M , the reduction tree⁴ for M is finitely branching. Thus, if M is any strongly normalizing term (abbreviated as *SN term* from now on), every path in its reduction tree is finite, and since this tree is finite branching, by König's lemma, this reduction tree is finite. Thus, for any SN term M , the depth⁵ of its reduction tree is a natural number, and we will denote it as $d(M)$. We now check the conditions (P1)-(P5). (P1) and (P2) are obvious.

(P3a) Since $M \in SN_{\sigma \rightarrow \tau}$ and $N \in SN_\sigma$, $d(M)$ and $d(N)$ are finite. We prove by induction on $d(M) + d(N)$ that MN is SN. We consider all possible ways that $MN \xrightarrow{\beta} P$. Since M is simple, MN itself is not a redex, and so $P = M_1N_1$ where either $N = N_1$ and $M \xrightarrow{\beta} M_1$, or $M = M_1$ and $N \xrightarrow{\beta} N_1$.

If M_1 is simple or $M_1 = M$, $d(M_1) + d(N_1) < d(M) + d(N)$, and by the induction hypothesis, $P = M_1N_1$ is SN. Otherwise, $M_1 = \lambda x: \sigma. M'$, $N_1 = N$, by assumption $(\lambda x: \sigma. M')N$ is SN, and so P is SN. Thus, $P = M_1N_1$ is SN in all cases, and MN is SN.

(P3b) Since $M \in SN_{\forall X. \sigma}$, $d(M)$ is finite. We prove by induction on $d(M)$ that $M\tau$ is SN. We consider all possible ways that $M\tau \xrightarrow{\beta} P$. Since M is simple, $M\tau$ itself is not a redex, and so $P = M_1\tau$ where $M \xrightarrow{\beta} M_1$.

If M_1 is simple, $d(M_1) < d(M)$, and by the induction hypothesis, $P = M_1\tau$ is SN. Otherwise, $M_1 = \lambda X. M'$, by assumption $(\lambda X. M')\tau$ is SN, and so P is SN. Thus, $P = M_1\tau$ is SN in all cases, and $M\tau$ is SN.

(P4) These cases are all similar, and hold because a reduction cannot apply at the outermost level.

⁴the tree of reduction sequences from M

⁵the length of a longest path in the tree, counting the number of edges

(P4a) Any reduction from $\lambda x:\sigma. M$ must be of the form $\lambda x:\sigma. M \xrightarrow{+}_\beta \lambda x:\sigma. M'$ where $M \xrightarrow{+}_\beta M'$. We use a simple induction on $d(M)$.

(P4b) Similar to (P4a).

(P5a) Since $N \in SN_\sigma$ and $M[N/x] \in SN_\tau$, the term M itself is SN. Thus, $d(M)$ and $d(N)$ are finite. We prove by induction on $d(M) + d(N)$ that $(\lambda x:\sigma. M)N$ is SN. We consider all possible ways that $(\lambda x:\sigma. M)N \rightarrow_\beta P$. Either $P = (\lambda x:\sigma. M_1)N$ where $M \rightarrow_\beta M_1$, or $P = (\lambda x:\sigma. M)N_1$ where $N \rightarrow_\beta N_1$, or $P = M[N/x]$. In the first two cases, $d(M_1) + d(N) < d(M) + d(N)$, $d(M) + d(N_1) < d(M) + d(N)$, and by the induction hypothesis, P is SN. In the third case, by assumption $M[N/x]$ is SN. But then, P is SN in all cases, and so $(\lambda x:\sigma. M)N$ is SN.

(P5b) This case is quite similar to (P5a). Since $M[\tau/X] \in SN_{\sigma[\tau/X]}$, the term M itself is SN. Thus, $d(M)$ is finite. We prove by induction on $d(M)$ that $(\lambda X. M)\tau$ is SN. We consider all possible ways that $(\lambda X. M)\tau \rightarrow_\beta P$. Either $P = (\lambda X. M_1)\tau$ where $M \rightarrow_\beta M_1$, or $P = M[\tau/X]$. In the first case, $d(M_1) < d(M)$, and by the induction hypothesis, P is SN. In the second case, by assumption $M[\tau/X]$ is SN. But then, P is SN in all cases, and so $(\lambda X. M)\tau$ is SN. \square

Confluence can be shown exactly as in Gallier [3].

9 Conclusion and Suggestions for Further Research

A semantic notion of realizability using the notion of a cover algebra was defined and investigated. For this, we introduced a new class of semantic structures equipped with preorders, called pre-applicative structures. In this framework, we proved a general realizability theorem. Applying this theorem to the special cases of the term model for the simply-typed λ -calculus and for the second-order λ -calculus, we obtained some general theorems for proving properties of typed λ -terms, in particular, strong normalization and confluence.

This approach clarifies the reducibility method by showing that the closure conditions on candidates of reducibility can be viewed as sheaf conditions. Indeed, cover conditions provide a clear axiomatization of the conditions needed for the proof of the realizability theorem. Our approach yields a clearer separation of the semantic versus the syntactic ingredients of the proof. For example, the fact that the sheaf property is an invariant with respect to the notion of realizability is a semantic property which has little to do with λ -terms. In fact, this uses only part of the pre-applicative structure (`app`, `tapp`, π_1 , π_2 , `inl`, `inr`). On the other hand, at some point, it is necessary to interpret λ -terms in order to show what amounts to the soundness of our realizability interpretation, and it is in this part that substitution and reduction properties of λ -terms play a role. In traditional presentations of proofs using reducibility, the underlying pre-applicative structure of the term model is only implicit, and it is harder to see that substitutions are really valuations. It is also practically impossible to see that cover conditions are present.

As we mentioned in part I of this paper, Hyland and Ong [5] show how strong normalization proofs can be obtained from the construction of a modified realizability topos. Very roughly, they show how a suitable quotient of the strongly normalizing untyped terms can be made into a categorical (modified realizability) interpretation of system F. There is no doubt that Hyland and Ong's approach and our approach are related, but the technical details are very different, and we

are unable to make a precise comparison at this point. Clearly, further work is needed to clarify the connection between Hyland and Ong’s approach and ours.

We have checked that in all proofs of reducibility that we are aware of, except for a recent paper by McAllester, Kučan, and Otth [7], the conditions on sets of realizers are sheaf conditions.⁶ However, the pre-applicative structures defined in this paper are not always general enough to carry out these proofs (for example, in the case of untyped λ -terms and typing systems with intersection types). McAllester, Kučan, and Otth [7], prove various strong normalization results using another variation of the reducibility method, and we need to understand how this method relates to the method presented in this paper. We believe that nonextensional structures are interesting in their own right, and thus we think that it would be interesting to investigate classes of nonextensional structures more general than pre-applicative structures (perhaps using category theory). When dependent types are considered, we run into the problem that interpreting types requires interpreting terms. We were able to define cover conditions that seem adequate for proving a general realizability theorem, but we ran into problems in defining the meaning of terms. The problem has to do with type-conversion rules: a term no longer has a unique type, and we run into a coherence problem in attempting to define the meaning of term by induction on typing-derivations. Overcoming this difficulty seems to be the most pressing open problem. More generally, we believe that there is a deeper connection between realizability semantics and other kinds of semantics, and that the notion of a cover algebra plays a significant role in that connection. We believe that understanding this connection would be worthwhile.

Acknowledgment: I wish to express my gratitude to Jim Lipton, since I would not have been able to write this paper without his inspiring suggestions and incisive questions. I also would like to thank Philippe de Groote, Andre Scedrov, and Scott Weinstein, for some very helpful comments.

References

- [1] V. Breazu-Tannen and T. Coquand. Extensional models for polymorphism. *Theoretical Computer Science*, 59:85–114, 1988.
- [2] Jean H. Gallier. On Girard’s “candidats de reductibilité”. In P. Odifreddi, editor, *Logic And Computer Science*, pages 123–203. Academic Press, London, New York, May 1990.
- [3] Jean H. Gallier. On the correspondence between proofs and λ -terms. In P. DeGroote, editor, *The Curry-Howard Isomorphism*, Cahiers du Centre de Logique. Université Catholique de Louvain, 1993. To appear.
- [4] Jean-Yves Girard. *Interprétation fonctionnelle et élimination des coupures de l’arithmétique d’ordre supérieur*. PhD thesis, Université de Paris VII, June 1972. Thèse de Doctorat d’Etat.
- [5] J. M. E. Hyland and C.-H. L. Ong. Modified realizability topos and strong normalization proofs. In M. Bezem and J.F. Groote, editors, *Typed Lambda Calculi and Applications*, volume 664 of *Lecture Notes in Computer Science*, pages 179–194. Springer Verlag, 1993.
- [6] S. MacLane and I. Moerdijk. *Sheaves in Geometry and Logic*. Springer Verlag, New York, 1992.

⁶We need to examine more closely the approach in [7] to determine whether it fits into our framework.

- [7] D. McAllester, J. Kučan, and D. F. Otth. A proof of strong normalization for F_2 , F_ω , and beyond. Technical report, MIT, Boston, Mass, 1993. Draft.