

Proving Properties of Typed λ -Terms Using Realizability, Covers, and Sheaves

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

December 17, 2010

Abstract. The main purpose of this paper is to take apart the reducibility method in order to understand how its pieces fit together, and in particular, to recast the conditions on candidates of reducibility as sheaf conditions. There has been a feeling among experts on this subject that it should be possible to present the reducibility method using more semantic means, and that a deeper understanding would then be gained. This paper gives mathematical substance to this feeling, by presenting a generalization of the reducibility method based on a semantic notion of realizability which uses the notion of a cover algebra (as in abstract sheaf theory). A key technical ingredient is the introduction 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. Kleene's recursive realizability and a variant of Kreisel's modified realizability both fit into this framework. We are then able to prove a meta-theorem which shows that if a property of realizers satisfies some simple conditions, then it holds for the semantic interpretations of all terms. 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. The above approach is applied to the simply-typed λ -calculus (with types \rightarrow , \times , $+$, and \perp), and to the second-order (polymorphic) λ -calculus (with types \rightarrow and \forall^2), for which it yields a new theorem.

*This research was partially supported by ONR Grant N00014-93-1-1217.

1 Introduction

Kleene, Kreisel, and others ([13], [16], [26]), introduced *realizability*, a certain kind of semantics for intuitionistic logic. Realizability can be used to show that certain axioms are consistent with certain intuitionistic theories of arithmetic, or to show that certain axioms are not derivable in these theories (see Kleene [14], Troelstra [26], Troelstra and van Dalen [27], and Beeson [1]). Tait [24], introduced *reducibility* (or *computability*), as a technique for proving strong normalization for the simply-typed λ -calculus. Girard [7], introduced the method of the *candidates of reducibility* a technique for proving strong normalization for the second-order typed λ -calculus (and F_ω). Statman [23] and Mitchell [20], observed that reducibility can be used to prove other properties besides strong normalization, for example, confluence.

The above lead to some natural observations:

- There are some similarities between reducibility and realizability, but they remain somewhat implicit.
- Proofs by reducibility use an interpretation of the types, but such interpretations are very syntactical.
- Proofs by reducibility seem to involve the construction of certain kinds of models.
- Proofs by reducibility use various inductive invariants (due to Girard [6, 7], Tait [24, 25], Krivine, [17]), but it is hard to see what they have in common.

These observations suggest the following two questions which are the primary concerns 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?**

This paper provides some answers to the above questions. But before explaining our results, we would like to explain our motivations and our point of view a little more. Reducibility proofs are seductive and thrilling, but also elusive. Following these proofs step-by-step, we see that they “work” (when they are not wrong!), but I claim that most of us would still admit that they are not sure *why* these proofs work! The situation is somewhat comparable to driving a Ferrari (I suppose): the feeling of power is tremendous, but what exactly is under the hood? What kind of carburetor, what kind of valve mechanism, gives such power and flexibility?

For a number of years, I have tried to take apart the wonderful engine of the reducibility method, look inside its carburetor, etc. Mathematically, in order to make some progress, it is often necessary to understand the various axioms that are used in a complex proof. It is often necessary to understand which ingredients of a proof are incidental, and which are really crucial to the proof. For example, in reducibility proofs, since the objective is usually to prove strong normalization, conditions specific to strong normalization are usually intimately mixed with other conditions on candidates. However, this is placing somewhat of a straight-jacket on the method of reducibility, and this is also somewhat confusing. Indeed, we know that other properties besides strong normalization can be shown, even some that cannot follow from strong normalization, for

instance, head-normalization, in the case of the pure λ -calculus (for several examples, see Krivine [17]).

Similarly, properties of substitutions are usually needed in middle of reducibility proofs, and I often wondered why. Another instance of a confusing overlap is that in approaches where reducibility is generalized to apply to a general property \mathcal{P} , it is assumed that \mathcal{P} satisfies the candidate conditions. As we shall see, this is unnecessary.

This paper consists of the observations that we find worth reporting, resulting from our many attempts to take the reducibility engine apart.

First, we found that it was necessary to step away from the syntax to have a clearer view. Thus, we define an abstract notion of semantic realizability which uses the notion of a cover algebra (covering families used in abstract sheaf theory). For this, we introduce a new class of structures equipped with preorders, called pre-applicative structures. These structures need not be extensional. Kleene's recursive realizability and a variant of Kreisel's modified realizability both fit into this framework. In this setting, it turns out that the family $(\mathbf{r}[\sigma])_{\sigma \in \mathcal{T}}$ of sets of realizers associated with the types, is a sheaf. Secondly, 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 a type σ is provable and M is a proof for σ , then the meaning $\mathcal{A}[[M]]\rho$ of M is a realizer of σ that satisfies the property \mathcal{P} . As a corollary, considering the term model for the simply-typed λ -calculus (with types \rightarrow , \times , $+$, and \perp), we obtain simple proofs for strong normalization and confluence. We also extend our method to system F.

We had previously discovered that it was possible to prove a general meta-theorem for the simply-typed λ -calculus (Gallier [5]). However, this previous work is still purely syntactical, and in our opinion, the present work goes much further in clarifying the nature of the candidate conditions, and separating the semantic from the syntactic components of reducibility proofs.

In our opinion, the new light on the reducibility method is that the conditions on the candidates of reducibility are not just a lucky strike (nevertheless, we still admire Girard, Tait, and other creators of the reducibility method for their remarkable intuition). In fact, these conditions can be viewed as *sheaf conditions*. I remember vividly when this idea occurred to me on December 8, 1992, while Jim Lipton was lecturing on cover conditions for sheaf models of intuitionistic logic. For several weeks, Jim had been lecturing on realizability methods, and when he explained how cover conditions unified all these approaches, I realized that the same idea could be applied to the conditions on candidates of reducibility. From that point on, it was very natural to attempt to define semantic realizability models of the *reduction relation*, and not of the convertibility relation (which is probably what held people back). Indeed, these models are **not** models of λ -calculi in the traditional sense, since they are not models of the convertibility relation, but instead models of the *reduction relation*. This idea is actually not new, and has been explored by Girard [8], Jacobs, Margaria, and Zacchi [12], and Plotkin [22]. However, our class of models is new, and the way we use them certainly appears to be new, although the next paragraph may attenuate our claim. In any case, our method has the advantage of dissociating the more semantic components of proofs of reducibility from the purely syntactic components, which have to do with the λ -calculus under consideration.

In a recent paper, Hyland and Ong [11] 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 somewhat related, but the technical details are very different, and we are unable to make a precise comparison at this point. What we can say is that our aim is not to provide a new class of categorical models, but rather to provide a better axiomatization of the conditions that make the proof go through. For this purpose, we believe that the notion of a cover algebra is particularly well suited. Clearly, further work is needed to clarify the connection between Hyland and Ong's approach and ours.

In order to motivate our approach and to help the reader's intuition, we first sketch our approach for the simply-typed λ -calculus λ^\rightarrow .

Recall that the types and the terms of λ^\rightarrow are given by the following grammar:

$$\sigma ::= b \mid (\sigma \rightarrow \sigma)$$

$$M ::= c \mid x \mid (MM) \mid (\lambda x:\sigma. M).$$

The type-checking rules are as usual (see section 2), and we let Λ_σ denote the set of λ -terms of type σ .

It is important to observe that there are two classes of terms:

1. Those created by *introduction rules*, or *I-terms*, $\lambda x:\sigma. M$;
2. Those created by *elimination rules*, MN .

I-terms play a special role, because *the only way to create a redex* is to combine an *I*-term with some other term. Terms that are not *I*-terms, are called *simple*, or *neutral*: x, c, MN .

Girard realized the importance of simple terms (see his (CR1-CR3)-conditions in Girard [7]). However, Koletsos [15] realized the following crucial fact:

Fact: $MN \xrightarrow{*}_\beta Q$, where Q is an *I*-term, *only if* M itself reduces to an *I*-term.

Let $\mathcal{P} = (P_\sigma)_{\sigma \in \mathcal{T}}$ be a family of properties of the simply-typed λ -terms (that type-check). For example, $M \in P_\sigma$ holds iff M is strongly normalizing (SN), or $M \in P_\sigma$ holds iff confluence holds from M . In Gallier [5], we obtained the following theorem.

Theorem A. Let \mathcal{P} be a family satisfying the conditions:

- (P1) $x \in P_\sigma, c \in P_\sigma$, for every variable x and constant c of type σ .
- (P2) If $M \in P_\sigma$ and $M \rightarrow_\beta N$, then $N \in P_\sigma$.
- (P3) If M is simple, $M \in P_{\sigma \rightarrow \tau}, N \in P_\sigma$, and $(\lambda x:\sigma. M')N \in P_\tau$ whenever $M \xrightarrow{+}_\beta \lambda x:\sigma. M'$, then $MN \in P_\tau$.
- (P4) If $M \in P_\tau$, then $\lambda x:\sigma. M \in P_{\sigma \rightarrow \tau}$.
- (P5) If $N \in P_\sigma$ and $M[N/x] \in P_\tau$, then $(\lambda x:\sigma. M)N \in P_\tau$.

Then, P_σ holds for all terms of type σ , i.e. $P_\sigma = \Lambda_\sigma$, for every $\sigma \in \mathcal{T}$.

In particular, SN and confluence are easily shown to satisfy conditions (P1)-(P5), and as a corollary, we obtain that SN and confluence hold for λ^\rightarrow .

The proof of Theorem A uses a version of reducibility in which the types are interpreted as follows:

$$\begin{aligned} \llbracket \sigma \rrbracket &= P_\sigma, & \sigma \text{ a base type,} \\ \llbracket \sigma \rightarrow \tau \rrbracket &= \{M \mid M \in P_{\sigma \rightarrow \tau}, \text{ and for all } N, \\ &\quad \text{if } N \in \llbracket \sigma \rrbracket \text{ then } MN \in \llbracket \tau \rrbracket\}. \end{aligned}$$

The other crucial concept used in the proof is the notion of a \mathcal{P} -candidate, inspired by the work of Girard, Koletsos, and Mitchell.

A family $\mathcal{S} = (S_\sigma)_{\sigma \in \mathcal{T}}$ of nonempty sets of terms is a \mathcal{P} -candidate iff it satisfies the following conditions:

- (S1) $S_\sigma \subseteq P_\sigma$.
- (S2) If $M \in S_\sigma$ and $M \rightarrow_\beta N$, then $N \in S_\sigma$.
- (S3) If M is simple, $M \in P_\sigma$, and $\lambda x:\gamma. M' \in S_\sigma$ whenever $M \xrightarrow{+}_\beta \lambda x:\gamma. M'$, then $M \in S_\sigma$.

Condition (S3) can be rewritten as follows:

- (S3) If M is simple, $M \in P_\sigma$, and $Q \in S_\sigma$ whenever $M \xrightarrow{+}_\beta Q$ and Q is an I -term, then $M \in S_\sigma$.

The advantage of the above formulation is that it applies to more general calculi, as long as the notion of an I -term is well-defined.

We now take the (somewhat wild) step of relating the previous concepts to covers (in the sense of Grothendieck) and sheaves (see MacLane and Moerdijk [18]). We can think of the set

$$\{N \mid M \xrightarrow{+}_\beta Q \xrightarrow{*}_\beta N, Q \text{ an } I\text{-term}\}$$

as a *cover* of M .¹ Then, writing $\text{Cov}_\sigma(C, M)$ for “the set C covers M ”, condition (S3) can be formulated as:

- (S3) If $\text{Cov}_\sigma(C, M)$, and $C \subseteq S_\sigma$, then $M \in S_\sigma$.

We can view $\mathcal{S} = (S_\sigma)_{\sigma \in \mathcal{T}}$ as a functor

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

by letting $\mathcal{S}(M) = \{\sigma \mid M \in S_\sigma\}$, where \mathcal{LT} is basically the term model, with preorder $N \preceq M$ iff $M \xrightarrow{*}_\beta N$. Indeed, (S2) says that $\mathcal{S}(M) \subseteq \mathcal{S}(N)$ if $N \preceq M$. Then, (S3) can be formulated as:

¹When M is a simple term that is not stubborn, see section 12 for details.

(S3) If $\text{Cov}_\sigma(C, M)$, and $\sigma \in \mathcal{S}(N)$ for every $N \in C$, then $\sigma \in \mathcal{S}(M)$.

For those familiar with sheaves, this looks like a “sheaf condition”. Indeed, the covers arising in reducibility proofs satisfy some conditions defined by Grothendieck in the sixties! These are the conditions for *Grothendieck topologies on sites* (see MacLane and Moerdijk [18]).

In order to make all this clear, first, we need to define some appropriate semantic structures that will be our sites. Normally, sites are categories. Thus, we will consider semantic structures where the carriers are equipped with preorders. These preorders are a semantic version of reduction ($\xrightarrow{*}_\beta$).

In order to understand what motivated the definition of the semantic structures used in this paper, it is useful to review the usual definition of an applicative structure for the simply-typed λ -calculus (for example, as presented in Gunter [10]). For simplicity, we are restricting our attention to arrow types. Let \mathcal{T} be the set of simple types built up from some base types using the constructor \rightarrow . Given a signature Σ of function symbols, where each symbol in Σ is assigned some type in \mathcal{T} , an *applicative structure* \mathcal{A} is defined as a triple

$$\langle (A^\sigma)_{\sigma \in \mathcal{T}}, (\mathbf{app}^{\sigma, \tau})_{\sigma, \tau \in \mathcal{T}}, \text{Const} \rangle,$$

where

1. $(A^\sigma)_{\sigma \in \mathcal{T}}$ is a family of nonempty sets called *carriers*,
2. $(\mathbf{app}^{\sigma, \tau})_{\sigma, \tau \in \mathcal{T}}$ is a family of *application operators*, where each $\mathbf{app}^{\sigma, \tau}$ is a total function $\mathbf{app}^{\sigma, \tau}: A^{\sigma \rightarrow \tau} \times A^\sigma \rightarrow A^\tau$;
3. and Const is a function assigning a member of A^σ to every symbol in Σ of type σ .

The meaning of simply-typed λ -terms is usually defined using the notion of an *environment*, or *valuation*. A valuation is a function $\rho: \mathcal{X} \rightarrow \bigcup (A^\sigma)_{\sigma \in \mathcal{T}}$, where \mathcal{X} is the set of term variables. Although when nonempty carriers are considered (which is the case right now), it is not really necessary to consider judgements for interpreting λ -terms, since we are going to consider more general applicative structures, we define the semantics of terms using judgements. Recall that a judgement is an expression of the form $\Gamma \triangleright M: \sigma$, where Γ , called a context, is a set of variable declarations of the form $x_1: \sigma_1, \dots, x_n: \sigma_n$, where the x_i are pairwise distinct and the σ_i are types, M is a simply-typed λ -term, and σ is a type. There is a standard proof system that allows to type-check terms. A term M type-checks with type σ in the context Γ (where Γ contains an assignment of types to all the variables in M) iff the judgement $\Gamma \triangleright M: \sigma$ is derivable in this proof system. Given a context Γ , we say that a valuation ρ *satisfies* Γ iff $\rho(x) \in A^\sigma$ for every $x: \sigma \in \Gamma$ (in other words, ρ respects the typing of the variables declared in Γ). Then given a context Γ and a valuation ρ satisfying Γ , the meaning $\llbracket \Gamma \triangleright M: \sigma \rrbracket \rho$ of a judgement $\Gamma \triangleright M: \sigma$ is defined by induction on the derivation of $\Gamma \triangleright M: \sigma$, according to the following clauses:

1. $\llbracket \Gamma \triangleright x: \sigma \rrbracket \rho = \rho(x)$, if x is a variable;
2. $\llbracket \Gamma \triangleright c: \sigma \rrbracket \rho = \text{Const}(c)$, if c is a constant;

3. $\llbracket \Gamma \triangleright MN : \tau \rrbracket \rho = \mathbf{app}^{\sigma, \tau}(\llbracket \Gamma \triangleright M : (\sigma \rightarrow \tau) \rrbracket \rho, \llbracket \Gamma \triangleright N : \sigma \rrbracket \rho)$,
4. $\llbracket \Gamma \triangleright \lambda x : \sigma. M : (\sigma \rightarrow \tau) \rrbracket \rho = f$, where f is the unique element of $A^{\sigma \rightarrow \tau}$ such that $\mathbf{app}^{\sigma, \tau}(f, a) = \llbracket \Gamma, x : \sigma \triangleright M : \tau \rrbracket \rho[x := a]$, for every $a \in A^\sigma$.

Note that in order for the element $f \in A^{\sigma \rightarrow \tau}$ to be uniquely defined in the last clause, we need to make certain additional assumptions. First, we assume that we are considering *extensional* applicative structures, which means that for all $f, g \in A^{\sigma \rightarrow \tau}$, if $\mathbf{app}(f, a) = \mathbf{app}(g, a)$ for all $a \in A^\sigma$, then $f = g$. This condition guarantees the uniqueness of f if it exists. The second condition is more technical, and asserts that each A^σ contains enough elements so that there is an element $f \in A^{\sigma \rightarrow \tau}$ such that $\mathbf{app}^{\sigma, \tau}(f, a) = \llbracket \Gamma, x : \sigma \triangleright M : \tau \rrbracket \rho[x := a]$, for every $a \in A^\sigma$.

Note that each operator $\mathbf{app}^{\sigma, \tau} : A^{\sigma \rightarrow \tau} \times A^\sigma \rightarrow A^\tau$ induces a function $\mathbf{fun}^{\sigma, \tau} : A^{\sigma \rightarrow \tau} \rightarrow [A^\sigma \Rightarrow A^\tau]$, where $[A^\sigma \Rightarrow A^\tau]$ denotes the set of functions from A^σ to A^τ , defined such that

$$\mathbf{fun}^{\sigma, \tau}(f)(a) = \mathbf{app}^{\sigma, \tau}(f, a),$$

for all $f \in A^{\sigma \rightarrow \tau}$, and all $a \in A^\sigma$. Then, extensionality is equivalent to the fact that each $\mathbf{fun}^{\sigma, \tau}$ is injective. Note that $\mathbf{fun}^{\sigma, \tau} : A^{\sigma \rightarrow \tau} \rightarrow [A^\sigma \Rightarrow A^\tau]$ is the “curried” version of $\mathbf{app}^{\sigma, \tau} : A^{\sigma \rightarrow \tau} \times A^\sigma \rightarrow A^\tau$, and it exists because the category of sets is Cartesian-closed.

The clause defining $\llbracket \Gamma \triangleright \lambda x : \sigma. M : (\sigma \rightarrow \tau) \rrbracket \rho$ suggests that a partial map $\mathbf{abst}^{\sigma, \tau} : [A^\sigma \Rightarrow A^\tau] \rightarrow A^{\sigma \rightarrow \tau}$, “abstracting” a function $\varphi \in [A^\sigma \Rightarrow A^\tau]$ into an element $\mathbf{abst}^{\sigma, \tau}(\varphi) \in A^{\sigma \rightarrow \tau}$, can be defined. For example, the function φ defined such that $\varphi(a) = \llbracket \Gamma, x : \sigma \triangleright M : \tau \rrbracket \rho[x := a]$ would be mapped to $\llbracket \Gamma \triangleright \lambda x : \sigma. M : (\sigma \rightarrow \tau) \rrbracket \rho$. In order for the resulting structure to be a model of β -reduction, we just have to require that $\mathbf{fun}^{\sigma, \tau}$ and $\mathbf{abst}^{\sigma, \tau}$ satisfy the axiom

$$\mathbf{fun}^{\sigma, \tau}(\mathbf{abst}^{\sigma, \tau}(\varphi)) = \varphi,$$

whenever $\varphi \in [A^\sigma \Rightarrow A^\tau]$ is in the domain of $\mathbf{abst}^{\sigma, \tau}$. But now, observe that if pairs of operators $\mathbf{fun}^{\sigma, \tau}, \mathbf{abst}^{\sigma, \tau}$ satisfying the above axiom are defined, the injectivity of $\mathbf{fun}^{\sigma, \tau}$ is superfluous for defining $\llbracket \Gamma \triangleright \lambda x : \sigma. M : (\sigma \rightarrow \tau) \rrbracket \rho$.

Thus, by defining a more general kind of applicative structure using the operators $\mathbf{fun}^{\sigma, \tau}$ and $\mathbf{abst}^{\sigma, \tau}$, we can still give meanings to λ -terms, even when these structures are nonextensional. In particular, our approach is an alternative to the method where one considers applicative structures with meaning functions, as for example in Mitchell [20]. In particular, the term structure together with the meaning function defined using substitution can be seen to be an applicative structure according to our definition. In fact, this approach allows us to go further. We can assume that each carrier A^σ is equipped with a preorder \preceq^σ , and rather than considering the equality

$$\mathbf{fun}^{\sigma, \tau}(\mathbf{abst}^{\sigma, \tau}(\varphi)) = \varphi,$$

we can consider inequalities

$$\mathbf{fun}^{\sigma, \tau}(\mathbf{abst}^{\sigma, \tau}(\varphi)) \succeq \varphi.$$

This way, we can deal with intentional (nonapplicative) structures that model reduction rather than conversion. We learned from Gordon Plotkin that models of β -reduction (or $\beta\eta$ -reduction) have been considered before, in particular by Girard [8], Jacobs, Margaria, and Zacchi [12], and

Plotkin [22]. However, except for Girard who studies qualitative domains for system F, the other authors consider models of the untyped λ -calculus. A brief presentation of these models can be found at the end of section 3.

Let us now briefly discuss how to generalize the above approach to the second-order (polymorphic) λ -calculus (with types \rightarrow and \forall^2). For this, we generalize pre-applicative structures. We now 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 an application, considering a suitable term model for the second-order λ -calculus, we obtain a new theorem for proving properties of terms in $\lambda^{\rightarrow, \forall^2}$. As a corollary, 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.

In order to understand what motivated our definition of second-order pre-applicative structures, 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 [2], we assume that we have an *algebra of types*, 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 operation on T , $[T \Rightarrow T]$ is a nonempty set of functions from T to T , and \forall is a function $\forall : [T \Rightarrow T] \rightarrow T$. We hope that readers will forgive us for denoting an algebra of types $\langle T, \rightarrow, [T \Rightarrow T], \forall \rangle$ with the same symbol T .

Intuitively, given a valuation $\theta : \mathcal{V} \rightarrow T$ (where \mathcal{V} is the set of type variables), a type $\sigma \in \mathcal{T}$ 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

1. T is an algebra of types;
2. $(A^s)_{s \in T}$ is a family of nonempty sets called *carriers*,
3. $(\mathbf{app}^{s,t})_{s,t \in T}$ is a family of *application operators*, where each $\mathbf{app}^{s,t}$ is a total function $\mathbf{app}^{s,t} : A^{s \rightarrow t} \times A^s \rightarrow A^t$;
4. $(\mathbf{tapp}^\Phi)_{\Phi \in [T \rightarrow T]}$ is a family of *type-application operators*, where each \mathbf{tapp}^Φ is a total function $\mathbf{tapp}^\Phi : A^{\forall(\Phi)} \times T \rightarrow \coprod (A^{\Phi(s)})_{s \in T}$, such that $\mathbf{tapp}^\Phi(f, t) \in A^{\Phi(t)}$, 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 \mathbf{tfun}^Φ of $\mathbf{tapp}^\Phi: A^{\forall(\Phi)} \times T \rightarrow \prod(A^{\Phi(s)})_{s \in T}$. For this, we define a kind of *dependent product* $\prod_\Phi(A^s)_{s \in T}$ (see definition 14.2). Then, we have families of operators $\mathbf{tfun}^\Phi: A^{\forall(\Phi)} \rightarrow \prod_\Phi(A^s)_{s \in T}$, and $\mathbf{tabst}^\Phi: \prod_\Phi(A^s)_{s \in T} \rightarrow A^{\forall(\Phi)}$, for every $\Phi \in [T \Rightarrow T]$.

This paper is organized as follows. The syntax of the simply-typed λ -calculus $\lambda^{\rightarrow, \times, +, \perp}$ is reviewed in section 2. Pre-applicative structures for λ^{\rightarrow} are defined in section 3, and some examples are given. The crucial notions of \mathcal{P} -cover algebras and of \mathcal{P} -sheaves are defined for λ^{\rightarrow} in section 4. The notion of \mathcal{P} -realizability is defined for λ^{\rightarrow} in section 5. In section 6, it is shown how to interpret terms in λ^{\rightarrow} in pre-applicative structures. The realizability theorem for the typed λ -calculus λ^{\rightarrow} is shown in section 7. Pre-applicative structures for the typed λ -calculus $\lambda^{\rightarrow, \times, +, \perp}$ are defined in section 8. The notions of \mathcal{P} -cover algebras and \mathcal{P} -realizability are extended to $\lambda^{\rightarrow, \times, +, \perp}$ in section 9. In section 10, it is shown how to interpret terms in $\lambda^{\rightarrow, \times, +, \perp}$ in pre-applicative structures. The realizability theorem for the typed λ -calculus $\lambda^{\rightarrow, \times, +, \perp}$ is shown in section 11. Section 12 contains an application of the main theorem of section 11 to prove a general theorem about terms of the system $\lambda^{\rightarrow, \times, +, \perp}$. The syntax of the second-order λ -calculus $\lambda^{\rightarrow, \forall^2}$ is reviewed in section 13. Pre-applicative structures for $\lambda^{\rightarrow, \forall^2}$ are defined in section 14. The notions of \mathcal{P} -cover algebras and of \mathcal{P} -sheaves are defined for $\lambda^{\rightarrow, \forall^2}$ in section 15. The notion of \mathcal{P} -realizability for $\lambda^{\rightarrow, \forall^2}$ is defined in section 16. In section 17, 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 18. Section 19 contains an application of the main theorem of section 18 to prove a new general theorem for $\lambda^{\rightarrow, \forall^2}$ (theorem 19.6). Section 20 contains the conclusion and some suggestions for further research. Extensional and $\beta\eta$ pre-applicative structures are defined in section 21.

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

Let \mathcal{T} denote the set of (simple) types, consisting of base types, including the special base type \perp , and compound types $(\sigma \rightarrow \tau)$, $(\sigma \times \tau)$, and $(\sigma + \tau)$. The presentation will be simplified if we adopt the definition of simply-typed λ -terms where all the variables are explicitly assigned types once and for all. More precisely, we have a family $\mathcal{X} = (X_\sigma)_{\sigma \in \mathcal{T}}$ of variables, where each X_σ is a countably infinite set of variables of type σ , and $X_\sigma \cap X_\tau = \emptyset$ whenever $\sigma \neq \tau$. Using this definition, there is no need to drag contexts along, and the most important feature of the proof, namely the reducibility method, is easier to grasp.

Instead of using the construct **case** P of $\mathbf{inl}(x: \sigma) \Rightarrow M \mid \mathbf{inr}(y: \tau) \Rightarrow N$, it will be more convenient and simpler to use a 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 **case** P of $\mathbf{inl}(x: \sigma) \Rightarrow M \mid \mathbf{inr}(y: \tau) \Rightarrow N$, where P is of type $\sigma + \tau$, as $[\lambda x: \sigma. M, \lambda y: \tau. N]P$. The type-checking rules of the system are summarized in the following definition.

Definition 2.1 The terms of the typed λ -calculus $\lambda^{\rightarrow, \times, +, \perp}$ are defined by the following rules.

$$x: \sigma, \quad \text{when } x \in X_\sigma,$$

(we can also have $c: \sigma$, for a set of constants that have been preassigned types).

$$\frac{M: \perp}{\nabla_{\sigma}(M): \sigma} \quad (\perp\text{-elim})$$

with $\sigma \neq \perp$,

$$\frac{M: \tau}{(\lambda x: \sigma. M): \sigma \rightarrow \tau} \quad (\text{abstraction})$$

where $x \in X_{\sigma}$;

$$\frac{M: \sigma \rightarrow \tau \quad N: \sigma}{(MN): \tau} \quad (\text{application})$$

$$\frac{M: \sigma \quad N: \tau}{\langle M, N \rangle: \sigma \times \tau} \quad (\text{pairing})$$

$$\frac{M: \sigma \times \tau}{\pi_1(M): \sigma} \quad (\text{projection}) \quad \frac{M: \sigma \times \tau}{\pi_2(M): \tau} \quad (\text{projection})$$

$$\frac{M: \sigma}{\mathbf{inl}(M): \sigma + \tau} \quad (\text{injection}) \quad \frac{M: \tau}{\mathbf{inr}(M): \sigma + \tau} \quad (\text{injection})$$

$$\frac{M: (\sigma \rightarrow \delta) \quad N: (\tau \rightarrow \delta)}{[M, N]: (\sigma + \tau) \rightarrow \delta} \quad (\text{co-pairing})$$

The standard elimination rule for $+$ is:

$$\frac{P: \sigma + \tau \quad M: \delta \quad N: \delta}{(\text{case } P \text{ of } \mathbf{inl}(x: \sigma) \Rightarrow M \mid \mathbf{inr}(y: \tau) \Rightarrow N): \delta} \quad (\text{by-cases})$$

where $x \in X_{\sigma}$ and $y \in X_{\tau}$.

We can design reduction rules so that the construct $[\lambda x: \sigma. M, \lambda y: \tau. N]P$ behaves just like $\text{case } P \text{ of } \mathbf{inl}(x: \sigma) \Rightarrow M \mid \mathbf{inr}(y: \tau) \Rightarrow N$. For this, we design more atomic reduction rules for $[M, N]$. These rules do not incorporate the β -reduction step implicit in the traditional reduction rules.

Definition 2.2 The reduction rules of the system $\lambda^{\rightarrow, \times, +, \perp}$ are listed below:

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

The traditional rules for the **case** construct are

$$\begin{aligned} \text{case inl}(P) \text{ of } \text{inl}(x: \sigma) \Rightarrow M \mid \text{inr}(y: \tau) \Rightarrow N &\longrightarrow M[P/x], \\ \text{case inr}(P) \text{ of } \text{inl}(x: \sigma) \Rightarrow M \mid \text{inr}(y: \tau) \Rightarrow N &\longrightarrow N[P/y]. \end{aligned}$$

The above reduction rules can be simulated by the $[-, -]$ -rules of definition 2.2 and β -reduction as follows:

$$\begin{aligned} [\lambda x: \sigma. M, \lambda y: \tau. N] \text{inl}(P) &\longrightarrow (\lambda x: \sigma. M)P \longrightarrow_{\beta} M[P/x], \\ [\lambda x: \sigma. M, \lambda y: \tau. N] \text{inr}(P) &\longrightarrow (\lambda y: \tau. N)P \longrightarrow_{\beta} N[P/y]. \end{aligned}$$

The reduction relation defined by the rules of definition 2.2 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 Λ_{σ} denote the set of λ -terms of type σ .

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

3 Pre-Applicative Structures for λ^{\rightarrow}

In this section, some new semantic structures called pre-applicative structures are defined. In order to simplify the presentation, we restrict our attention to the type constructor \rightarrow , and we do not discuss extensional or $\beta\eta$ pre-applicative structures. We also show that the term model can be viewed as a pre-applicative β -structures.

Definition 3.1 A *pre-applicative β -structure* is a structure

$$\mathcal{A} = \langle A, \preceq, \mathbf{fun}, \mathbf{abst} \rangle,$$

where

- $A = (A^{\sigma})_{\sigma \in \mathcal{T}}$ is a family of (nonempty) sets called *carriers*;
- $(\preceq^{\sigma})_{\sigma \in \mathcal{T}}$ is a family of preorders, each \preceq^{σ} on A^{σ} ;
- $\mathbf{abst}^{\sigma, \tau}: [A^{\sigma} \Rightarrow A^{\tau}] \rightarrow A^{\sigma \rightarrow \tau}$, a family of partial operators;
- $\mathbf{fun}^{\sigma, \tau}: A^{\sigma \rightarrow \tau} \rightarrow [A^{\sigma} \Rightarrow A^{\tau}]$, a family of (total) operators.

It is assumed that \mathbf{fun} and \mathbf{abst} are monotonic. Furthermore, the following condition is satisfied

- (1) $\mathbf{fun}^{\sigma, \tau}(\mathbf{abst}^{\sigma, \tau}(\varphi)) \succeq \varphi$, whenever $\mathbf{abst}^{\sigma, \tau}(\varphi)$ is defined for $\varphi \in [A^{\sigma} \Rightarrow A^{\tau}]$;

The operators \mathbf{fun} induce (total) operators

$\mathbf{app}^{\sigma, \tau}: A^{\sigma \rightarrow \tau} \times A^{\sigma} \rightarrow A^{\tau}$, such that, for every $f \in A^{\sigma \rightarrow \tau}$ and every $a \in A^{\sigma}$,

$$\mathbf{app}^{\sigma, \tau}(f, a) = \mathbf{fun}^{\sigma, \tau}(f)(a).$$

Then, condition (1) can be written as

(1') $\mathbf{app}^{\sigma,\tau}(\mathbf{abst}^{\sigma,\tau}(\varphi), a) \succeq \varphi(a)$, for all $a \in A^\sigma$, for $\varphi \in [A^\sigma \Rightarrow A^\tau]$, whenever $\mathbf{abst}^{\sigma,\tau}(\varphi)$ is defined.

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

Intuitively, \mathcal{A} is a set of realizers. We will omit superscripts whenever possible.

When \mathcal{A} is an applicative β -structure, then, in definition 3.1, condition (1) amounts to

(1) $\mathbf{fun}^{\sigma,\tau} \circ \mathbf{abst}^{\sigma,\tau} = \mathbf{id}$ on the domain of definition of \mathbf{abst} .

In this case, \mathbf{abst} is injective and \mathbf{fun} is surjective on the domain of definition of \mathbf{abst} (and left inverse to \mathbf{abst}).

When we use a pre-applicative β -structure to interpret λ -terms, we assume that the domain of \mathbf{abst} is sufficiently large, but we have not elucidated this last condition yet. Given $M \in A^{\sigma \rightarrow \tau}$ and $N \in A^\sigma$, $\mathbf{app}(M, N)$ is also denoted as MN .

We can also define extensional pre-applicative structures and pre-applicative $\beta\eta$ -structures, but this will be done later.

Let us give an (important) example of a pre-applicative β -structure.

Definition 3.2 Let $A^\sigma = \Lambda_\sigma$ be the set of all typed λ -terms of type σ . We let \mathbf{app} be the obvious construct ($\mathbf{app}(M, N) = MN$). Define $N \preceq M$ iff $M \xrightarrow{*}_\beta N$. Finally, we need to define \mathbf{abst} . For every (type-preserving) substitution φ , for every term $M:\tau$ and for every variable x of type σ , consider the function $\varphi[x:\sigma \triangleright M:\tau]$ from A^σ to A^τ , defined such that,

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

for every $N:\sigma$. Given any such function $\varphi[x:\sigma \triangleright M:\tau]$, we let

$$\mathbf{abst}(\varphi[x:\sigma \triangleright M:\tau]) = (\lambda x:\sigma. M)[\varphi].$$

The structure just defined is denoted as \mathcal{LT}_β .

Clearly, $\mathbf{app}(\mathbf{abst}(\varphi[x:\sigma \triangleright M:\tau]), N) \succeq \varphi[x:\sigma \triangleright M:\tau](N)$, since

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

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 , $(\lambda y:\sigma. M[y/x])[\varphi] = (\lambda y:\sigma. M[y/x][\varphi])$. Then, for this choice of y ,

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

We learned from Gordon Plotkin that models of β -reduction (or $\beta\eta$ -reduction) have been considered before, in particular by Girard [8], Jacobs, Margaria, and Zacchi [12], and Plotkin [22]. In [8], definition 1.12, Girard defines a λ -structure as a triple $D = \langle X, H, K \rangle$ consisting of

- (i) a qualitative domain X ,
- (ii) a stable function H from X to $X \Rightarrow X$, and
- (iii) a stable function K from $X \Rightarrow X$ to X ,

where $X \Rightarrow X$ is the set of all traces of stable functions from X to X . Girard then shows that a λ -structure D models β -reduction if $H \circ K \subset Id_{X \Rightarrow X}$, and that D models η -reduction if $K \circ H \subset Id_X$ (note that the partial order \subset corresponds to the opposite of our ordering \preceq). Girard also states that such structures have nice features, in particular because they can be approximated by finite λ -structures.

The major difference with our approach is that the above models are intended for the untyped λ -calculus, and that we do not have a construct such as $X \Rightarrow X$.

In [22], section 3, Plotkin introduces a notion of model of β -reduction that he calls an *ordered λ -interpretation*. After Mitchell [20], Plotkin defines such a structure as a triple $\mathcal{P} = \langle P, \cdot, \llbracket \cdot \rrbracket(\cdot) \rangle$, where P is a partial order, \cdot is a monotonic application operation $\cdot : P \times P \rightarrow P$, and $\llbracket \cdot \rrbracket(\cdot)$ is a *meaning function*, that maps terms and environments to P , and such that some obvious conditions on $\llbracket \cdot \rrbracket(\cdot)$ hold. If the condition

$$\llbracket \lambda x. M \rrbracket(\rho) \cdot a \preceq \llbracket M \rrbracket(\rho[x := a]),$$

holds, we say that \mathcal{P} is a model of β -reduction. Plotkin then proceeds to show that such models are sound and complete with respect to Curry-style type inference systems (also known as systems for F -deducibility), for various type disciplines. The main difference with our approach is that Plotkin's structures are models of the untyped λ -calculus, and that meaning functions are an intrinsic part of their definition. In our definition, the meaning function is not part of the definition, but it is uniquely defined. For our purposes, this is a much more suitable approach.

Jacobs, Margaria, and Zacchi [12] define models of β -reduction, β -expansion, and β -conversion, quite similarly to Girard, but using cpo's, with $D \Rightarrow D$ the set of all Scott-continuous functions from D to D . They proceed to show how to construct models of filters with polymorphic and intersection types.

Other references to models of reduction can be found in Plotkin [22].

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

In this section, we introduce the bare minimum of concepts needed for understanding the notion of a sheaf on a site. Usually, sites are defined as categories with a notion of a cover, also called a Grothendieck topology (see MacLane and Moerdijk [18]). However, we are only dealing with very special categories, namely preorders, and in such a case, the definition of a Grothendieck topology can be simplified. For example, a sieve, rather than being a set of arrows, is just an ideal. Thus, we will define all the necessary concepts in terms of preorders, referring the interested reader to MacLane and Moerdijk [18] for a general treatment. Originally, the concept of a Grothendieck topology was introduced in order to generalize the notion of an open cover, so that sheaves could be defined on domains that are not necessarily topological spaces. Thus, the terminology "topology"

is not the most appropriate, since what is really been generalized is the notion of a cover, and not the notion of a topology, and following Grayson [9], we prefer to use the term cover algebra. 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 a pre-applicative structure \mathcal{A} , for any $M \in A^\sigma$, a *sieve on M* is any subset $C \subseteq A^\sigma$ 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 \mathbf{Cov} of binary relations \mathbf{Cov}_σ on $2^{A^\sigma} \times A^\sigma$, relating subsets of A^σ called *covers*, to elements of A^σ . Equivalently, \mathbf{Cov} can be defined as a family of functions $\mathbf{Cov}_\sigma: A^\sigma \rightarrow 2^{2^{A^\sigma}}$ assigning to every element $M \in A^\sigma$ a set $\mathbf{Cov}(M)$ of subsets of A^σ (the covers of M). Given any $M \in A^\sigma$, 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 $\mathcal{P} \subseteq \mathcal{A} \times \mathcal{T}$, such that $\mathcal{P}(M, \sigma)$ implies $M \in A^\sigma$, and for every $\sigma \in \mathcal{T}$, there is some $M \in A^\sigma$ s.t. $\mathcal{P}(M, \sigma)$. Equivalently, \mathcal{P} can be viewed as a family $\mathcal{P} = (P_\sigma)_{\sigma \in \mathcal{T}}$, where each P_σ is a nonempty subset of A^σ . The intuition behind \mathcal{P} is that it is a property of realizers. In this section, we will only consider cover conditions for the arrow type.

Definition 4.2 Let \mathcal{A} be a pre-applicative structure and let \mathcal{P} be a family $\mathcal{P} = (P_\sigma)_{\sigma \in \mathcal{T}}$, where each P_σ is a nonempty subset of A^σ . A *\mathcal{P} -cover algebra* (or *\mathcal{P} -Grothendieck topology*) on \mathcal{A} is a family \mathbf{Cov} of binary relations \mathbf{Cov}_σ on $2^{A^\sigma} \times A^\sigma$ satisfying the following properties:

- (0) $\mathbf{Cov}_\sigma(C, M)$ implies $M \in P_\sigma$ (equivalently, $\mathcal{P}(M, \sigma)$).
- (1) If $\mathbf{Cov}(C, M)$, then C is a sieve on M (an ideal below M).
- (2) If $M \in P_\sigma$, then $\mathbf{Cov}(\{N \mid N \preceq M\}, M)$ ($M \in P_\sigma$ is covered by the principal sieve on 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 $Q \preceq M'N$.

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)-(2) are two of 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 (3) and (4) are missing, because they are only needed for the sum type $+$ (or the existential type). They are also conditions on a Grothendieck topology.² Condition (5) is needed to take care of the

²Readers who are anxious to see the full set of conditions should take a look at definition 9.1.

extra structure. Note that it is not necessary to assume that covers are ideals (downwards closed), but this is not harmful.

We need to come up with a semantic characterization of the simple terms, and also of the notion of a stubborn element. This can be done as follows in terms of covers.

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

An example, let us consider the pre-applicative structure \mathcal{LT}_β of definition 3.2. Recall that an *I-term* is a term of the form $\lambda x: \sigma. M$. 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 , or an application MN . 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).

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

- (1) If $M \in P_\sigma$ and M is an I-term, then

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

- (2) If $M \in P_\sigma$ and M is a (simple and) stubborn term, then

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

- (3) If $M \in P_\sigma$ and M is a simple and non-stubborn term, then

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

The conditions of definition 4.2 are easily verified. The above notion of a cover will be used in section 12 to prove a general theorem about the simply typed λ -calculus.

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.

Definition 4.4 Let $\langle \mathcal{A}, \mathcal{P}, \text{Cov} \rangle$ be a \mathcal{P} -site. Properties (P1)-(P3) are defined as follows:

(P1) $\mathcal{P}(M, \sigma)$, for some stubborn element $M \in A^\sigma$.

(P2) If $\mathcal{P}(M, \sigma)$ and $M \succeq N$, then $\mathcal{P}(N, \sigma)$.

(P3) If $\text{Cov}_{\sigma \rightarrow \tau}(C, M)$, $\mathcal{P}(N, \sigma)$, and $\mathcal{P}(M'N, \tau)$ whenever $M' \in C$, then $\mathcal{P}(MN, \tau)$.

From now on, we only consider relations (families) \mathcal{P} satisfying conditions (P1)-(P3) of definition 4.4. Condition (P1) says that each P_σ contains some stubborn element. Finally, we are 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. Recall that \mathcal{T} denotes the set of simple types built up using the type constructor \rightarrow .

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

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

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

- (S1) $S_\sigma \subseteq P_\sigma$.
- (S2) If $M \in S_\sigma$ and $M \succeq N$, then $N \in S_\sigma$.
- (S3) If $\text{Cov}_\sigma(C, M)$, and $C \subseteq S_\sigma$, then $M \in S_\sigma$.

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. It turns out that 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 $\text{Cov}_\sigma(C, M)$ implies that $\mathcal{P}(M, \sigma)$, any \mathcal{P} satisfying conditions (P1)-(P3) trivially satisfies the sheaf property. Finally, note that (S3) and (P1) imply that S_σ is nonempty and contains all stubborn elements in P_σ (because stubborn elements have the empty cover).

By (P3), if $M \in P_{\sigma \rightarrow \tau}$ is stubborn and $N \in P_\sigma$ is any element, then $MN \in P_\tau$. Furthermore, MN is also stubborn. This follows from property (5) of a cover. Thus, if $M \in P_{\sigma \rightarrow \tau}$ is stubborn and $N \in P_\sigma$ is any element, then $MN \in P_\tau$ is stubborn.

We conclude this section by showing explicitly that definition 4.5 is indeed a sheaf condition (for a general and complete treatment, see MacLane and Moerdijk [18]). A pre-applicative structure \mathcal{A} can be viewed as a category whose objects are the elements of \mathcal{A} , and whose arrows are defined such that there is a single arrow denoted $a \rightarrow b$ from a to b iff $a \preceq b$. Then, \mathcal{A}^{op} is the category with the same objects as \mathcal{A} but with the reverse arrows (i.e., there is an arrow from a to b in \mathcal{A}^{op} iff $a \succeq b$).

Let $F: \mathcal{A}^{op} \rightarrow \mathbf{Sets}$ be a functor. Thus, F assigns a set $F(a)$ to every element $a \in \mathcal{A}$, and a function $F(b \rightarrow a): F(b) \rightarrow F(a)$ to every pair $a, b \in \mathcal{A}$ such that $a \preceq b$ (with the usual functorial

conditions). For the sake of brevity, let us denote $F(b \rightarrow a): F(b) \rightarrow F(a)$ as $F_a^b: F(b) \rightarrow F(a)$. Given any $a \in \mathcal{A}$, for any $x \in F(a)$ and any $b \in \mathcal{A}$ such that $b \preceq a$, $F_b^a(x)$ is a member of the set $F(b)$ that we will also denote as $x|b$. We can think of $x|b$ as the *restriction* of $x \in F(a)$ to $F(b)$.

Definition 4.6 Given a site $\langle \mathcal{A}, \mathcal{P}, \text{Cov} \rangle$ and a functor $F: \mathcal{A}^{op} \rightarrow \mathbf{Sets}$, for any $a \in \mathcal{A}$ and any cover C of a (a set C such that $\text{Cov}(C, a)$), a family $\{x_c \mid x_c \in F(c), c \in C\}$ is a *matching family* for C iff for every $c \in C$,

$$x_c|d = x_d \quad \text{for every } d \preceq c.$$

An *amalgamation* of a matching family $\{x_c \mid x_c \in F(c), c \in C\}$ is an element $x \in F(a)$ such that

$$x|c = x_c \quad \text{for every } c \in C.$$

The functor F is a *sheaf* iff for every $a \in \mathcal{A}$, every cover C of a (a set C such that $\text{Cov}(C, a)$), and every family $\{x_c \mid x_c \in F(c), c \in C\}$, if $\{x_c \mid x_c \in F(c), c \in C\}$ is a matching family for C , then it has a unique amalgamation $x \in F(a)$. The functor F is a \mathcal{P} -*sheaf* iff it is a sheaf, and for every $a \in \mathcal{A}$, $F(a) \subseteq \mathcal{T}$ and $\sigma \in F(a)$ implies that $a \in P_\sigma$.

Since a cover is a sieve, $d \preceq c$ for $c \in C$ implies that $d \in C$, and so x_d is a well defined element (of $F(d)$). If in \mathcal{A} , any two elements have a greatest lower bound, it can easily be shown that $\{x_c \mid x_c \in F(c), c \in C\}$ is a matching family for C iff for all $c, d \in C$, then

$$x_c|c \wedge d = x_d|c \wedge d.$$

If the functor F is a sheaf and has the property that the maps $F_a^b: F(b) \rightarrow F(a)$ (with $a \preceq b$) are inclusion maps, then for any matching family $\{x_c \mid x_c \in F(c), c \in C\}$, if x is its amalgamation, $x|c = x_c$ implies that $x = x_c$ for all $c \in C$. Thus, in this case, a matching family consists of a single element x such that $x \in F(c)$ for all $c \in C$. Then, the property of being a sheaf is equivalent to the following condition: For every $a \in \mathcal{A}$, for every cover C of a ,

$$\text{if } x \in F(c) \text{ for every } c \in C, \text{ then } x \in F(a).$$

Now, the functor $\mathcal{S}: \mathcal{A}^{op} \rightarrow \mathbf{Sets}$ defined earlier is such that $M \succeq N$ implies $\mathcal{S}(M) \subseteq \mathcal{S}(N)$. Thus, it is indeed technically true that definition 4.5 means that the functor \mathcal{S} is a \mathcal{P} -sheaf with respect to the Grothendieck topology defined by Cov .

5 \mathcal{P} -Realizability for the Arrow Type

In this section, we define a semantic notion of realizability. This notion is such that realizers are elements of some pre-applicative structure. In the special case when only the arrow type is considered, the definition of realizability does not refer to covers. However, cover conditions are needed for proving lemma 5.2, which basically shows that the notion of a \mathcal{P} -sheaf is an invariant w.r.t. realizability. The notion of \mathcal{P} -*realizability* is defined as follows.

Definition 5.1 Let $\langle \mathcal{A}, \mathcal{P}, \text{Cov} \rangle$ be a \mathcal{P} -site. The sets $\mathbf{r}[\sigma]$ of *realizers* of σ are defined as follows:

$$\begin{aligned} \mathbf{r}[\sigma] &= P_\sigma, & \sigma \text{ a base type,} \\ \mathbf{r}[\sigma \rightarrow \tau] &= \{M \mid M \in P_{\sigma \rightarrow \tau}, \text{ and for all } N, \text{ if } N \in \mathbf{r}[\sigma] \text{ then } MN \in \mathbf{r}[\tau]\}. \end{aligned}$$

Note that instead of defining the family of sets $\mathbf{r}[\sigma]$, we could have defined a binary relation \mathbf{r} such that $M \mathbf{r} \sigma$ iff $M \in \mathbf{r}[\sigma]$. This is the more standard way of defining realizability. Another important point worth noting is that in the definition of $\mathbf{r}[\sigma \rightarrow \tau]$, we are considering only those M such that $M \in P_{\sigma \rightarrow \tau}$. One might be concerned that this will cause difficulties in proving lemma 5.2, but conditions (P1-P3) have been designed to overcome this problem.

Lemma 5.2 *Given a scenic \mathcal{P} -site $\langle \mathcal{A}, \mathcal{P}, \text{Cov} \rangle$, if \mathcal{P} satisfies conditions (P1)-(P3), then $(\mathbf{r}[\sigma])_{\sigma \in \mathcal{T}}$ has the sheaf property, and each $\mathbf{r}[\sigma]$ contains all stubborn elements in P_σ .*

Proof. We proceed by induction on types. If σ is a base type, $\mathbf{r}[\sigma] = P_\sigma$, and obviously, every stubborn element in P_σ is in $\mathbf{r}[\sigma]$. Since $\mathbf{r}[\sigma] = P_\sigma$, (S1) is trivial, (S2) follows from (P2), and (S3) is also trivial.³

We now consider the induction step.

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

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

(S3). Assume that $\text{Cov}_{\sigma \rightarrow \tau}(C, M)$, and that $M' \in \mathbf{r}[\sigma \rightarrow \tau]$ for every $M' \in C$, where M is simple. Recall that by condition (0) of definition 4.2, $\text{Cov}_{\sigma \rightarrow \tau}(C, M)$ implies that $M \in P_{\sigma \rightarrow \tau}$. We prove that for every N , if $N \in \mathbf{r}[\sigma]$, then $MN \in \mathbf{r}[\tau]$. First, we prove that $MN \in P_\tau$, and for this we use (P3).

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

Now, consider $M \in P_{\sigma \rightarrow \tau}$ non stubborn. If $M' \in C$, then by assumption, $M' \in \mathbf{r}[\sigma \rightarrow \tau]$, and for any $N \in \mathbf{r}[\sigma]$, we have $M'N \in \mathbf{r}[\tau]$. Since by (S1), $N \in P_\sigma$ and $M'N \in P_\tau$, by (P3), we have $MN \in P_\tau$. Now, there are two cases.

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

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

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

We now need to relate λ -terms and realizers.

³In fact, if $\mathbf{r}[\sigma] = P_\sigma$, (S3) holds trivially even at nonbase types. This remark is useful if we allow type variables.

6 Interpreting terms in λ^{\rightarrow} in Pre-Applicative Structures

We show how terms in λ^{\rightarrow} are interpreted in pre-applicative structures. For this, we define a *meaning function*.

Definition 6.1 Given a pre-applicative structure \mathcal{A} , a *valuation*, or *environment*, is any function $\rho: \mathcal{X} \rightarrow \mathcal{A}$, such that $\rho(x) \in A^\sigma$ if $x: \sigma$. A *meaning function for \mathcal{A}* is a partial function $\mathcal{A}[\![-]\!](-)$ from pairs of (α -equivalence classes of) terms and valuations to \mathcal{A} , such that $\mathcal{A}[\![-]\!](M)\rho$ is defined whenever $M: \sigma$, in which case $\mathcal{A}[\![-]\!](M)\rho \in A^\sigma$. In addition, a meaning function satisfies the following conditions:

$$\mathcal{A}[\![-]\!](x)\rho = \rho(x)$$

$$\mathcal{A}[\![-]\!](MN)\rho = \mathbf{app}(\mathcal{A}[\![-]\!](M)\rho, \mathcal{A}[\![-]\!](N)\rho)$$

$$\mathcal{A}[\![-]\!](\lambda x: \sigma. M)\rho = \mathbf{abst}(f),$$

where f is the function defined such that, $f(a) = \mathcal{A}[\![-]\!](M)\rho[x := a]$, for every $a \in A^\sigma$.

It is routine to show that the following property holds:

$$\mathcal{A}[\![-]\!](M)\rho_1 = \mathcal{A}[\![-]\!](M)\rho_2, \text{ whenever } \rho_1(x) = \rho_2(x) \text{ for every } x \in FV(M) \text{ (independence)}$$

If we consider the pre-applicative structure $\mathcal{A} = \mathcal{LT}_\beta$ defined just after definition 3.1, then a valuation ρ is a substitution with an infinite domain. Using an induction on the structure of terms, it is easily verified that $\mathcal{LT}_\beta[\![-]\!](M)\rho = M[\varphi]$, where φ is the substitution defined by the restriction of ρ to $FV(M)$.

7 The Realizability Theorem For λ^{\rightarrow}

In this section, we prove the realizability lemma (lemma 7.6) for λ^{\rightarrow} , 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) of the introduction.

Definition 7.1 We say that a site $\langle \mathcal{A}, \mathcal{P}, \mathbf{Cov} \rangle$ is *well-behaved* iff the following condition holds:

- (1) For any $a \in A^\sigma$, any $\varphi \in [A^\sigma \Rightarrow A^\tau]$, if $\mathbf{abst}(\varphi)$ exists, $\mathbf{Cov}_\tau(C, \mathbf{app}(\mathbf{abst}(\varphi), a))$, and C is a nontrivial cover, then $c \preceq \varphi(a)$ for every $c \in C$.

In view of definition 6.1, definition 7.1 implies the following condition.

Definition 7.2 Given a meaning function $\mathcal{A}[\![-]\!](-)$ on the pre-applicative structure \mathcal{A} , condition (1) is defined as follows:

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

- (P4) For every $a \in A^\sigma$, if $\varphi(a) \in P_\tau$, where $\varphi \in [A^\sigma \Rightarrow A^\tau]$ and $\text{abst}(\varphi)$ exists, then $\text{abst}(\varphi) \in P_{\sigma \rightarrow \tau}$.
- (P5) If $a \in P_\sigma$ and $\varphi(a) \in P_\tau$, where $\varphi \in [A^\sigma \Rightarrow A^\tau]$ and $\text{abst}(\varphi)$ exists, then $\text{app}(\text{abst}(\varphi), a) \in P_\tau$.

In view of definition 6.1, definition 7.3 implies the following conditions.

Definition 7.4 Given a meaning function $\mathcal{A}-$ on the pre-applicative structure \mathcal{A} , conditions (P4) and (P5) are:

- (P4) If $\mathcal{A}[M]\rho \in P_\tau$, then $\mathcal{A}[\lambda x: \sigma. M]\rho \in P_{\sigma \rightarrow \tau}$.
- (P5) If $a \in P_\sigma$ and $\mathcal{A}[M]\rho[x := a] \in P_\tau$, then $\text{app}(\mathcal{A}[\lambda x: \sigma. M]\rho, a) \in P_\tau$.

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 ρ such that $\rho(y) \in \mathbf{r}[\gamma]$ for every $y: \gamma \in FV(M)$, if for every a , ($a \in \mathbf{r}[\sigma]$) implies $\mathcal{A}[M]\rho[x := a] \in \mathbf{r}[\tau]$, then $\mathcal{A}[\lambda x: \sigma. M]\rho \in \mathbf{r}[\sigma \rightarrow \tau]$.*

Proof. We prove that $\mathcal{A}[\lambda x: \sigma. M]\rho \in P_{\sigma \rightarrow \tau}$ and that for every every a , if $a \in \mathbf{r}[\sigma]$, then $\text{app}(\mathcal{A}[\lambda x: \sigma. M]\rho, a) \in \mathbf{r}[\tau]$. We will need the fact that the sets of the form $\mathbf{r}[\sigma]$ have the properties (S1)-(S3), but this follows from lemma 5.2, since (P1)-(P3) hold. First, we prove that $\mathcal{A}[\lambda x: \sigma. M]\rho \in P_{\sigma \rightarrow \tau}$.

Since $\rho(x) \in \mathbf{r}[\gamma]$ for every $x: \gamma \in FV(M)$, letting $a = \rho(x)$, by the assumption of lemma 7.5, $\mathcal{A}[M]\rho \in \mathbf{r}[\tau]$. Then, by (S1), and by (P4), we have $\mathcal{A}[\lambda x: \sigma. M]\rho \in P_{\sigma \rightarrow \tau}$.

Next, we prove that for every every a , if $a \in \mathbf{r}[\sigma]$, then $\text{app}(\mathcal{A}[\lambda x: \sigma. M]\rho, a) \in \mathbf{r}[\tau]$. Let us assume that $a \in \mathbf{r}[\sigma]$. Then, by the assumption of lemma 7.5, $\mathcal{A}[M]\rho[x := a] \in \mathbf{r}[\tau]$. Thus, by (S1), we have $a \in P_\sigma$ and $\mathcal{A}[M]\rho[x := a] \in P_\tau$. By (P5), we have $\text{app}(\mathcal{A}[\lambda x: \sigma. M]\rho, a) \in P_\tau$. Now, there are two cases.

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

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

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

We now prove the main realizability lemma for $\lambda \rightarrow$.

Lemma 7.6 *Given a well-behaved scenic site $\langle \mathcal{A}, \mathcal{P}, \text{Cov} \rangle$, if \mathcal{P} is a family satisfying conditions (P1)-(P5), then for every term M of type σ , for every valuation ρ such that $\rho(y) \in \mathbf{r}[\gamma]$ for every $y: \gamma \in FV(M)$, we have $\mathcal{A}[M]\rho \in \mathbf{r}[\sigma]$.*

Proof. We proceed by induction on the structure of M .

If M is a variable, then $\mathcal{A}[[x]]\rho = \rho(x) \in \mathbf{r}[[\sigma]]$ by the assumption on ρ .

If $M = M_1 N_1$, where M_1 has type $\sigma \rightarrow \tau$ and N_1 has type σ , by the induction hypothesis,

$$\mathcal{A}[[M_1]]\rho \in \mathbf{r}[[\sigma \rightarrow \tau]] \quad \text{and} \quad \mathcal{A}[[N_1]]\rho \in \mathbf{r}[[\sigma]].$$

By the definition of $\mathbf{r}[[\sigma \rightarrow \tau]]$, we get $\mathbf{app}(\mathcal{A}[[M_1]]\rho, \mathcal{A}[[N_1]]\rho) \in \mathbf{r}[[\tau]]$, i.e., $\mathcal{A}[[M_1 N_1]]\rho \in \mathbf{r}[[\tau]]$, by definition 6.1.

If $M = \lambda x:\sigma. M_1$, consider any $a \in \mathbf{r}[[\sigma]]$ and any valuation ρ such that $\rho(y) \in \mathbf{r}[[\gamma]]$ for every $y:\gamma \in FV(M_1) - \{x\}$. Note that by (S3) and (P1), $\mathbf{r}[[\sigma]]$ is indeed nonempty. Thus, the valuation $\rho[x:=a]$ has the property that $\rho(y) \in \mathbf{r}[[\gamma]]$ for every $y:\gamma \in FV(M_1)$. By the induction hypothesis applied to M_1 and $\rho[x:=a]$, we have $\mathcal{A}[[M_1]]\rho[x:=a] \in \mathbf{r}[[\tau]]$. Consequently, by lemma 7.5, $\mathcal{A}[[\lambda x:\sigma. M_1]]\rho \in \mathbf{r}[[\sigma \rightarrow \tau]]$. \square

If M is a closed term of type σ , the independence condition of definition 6.1 implies that $\mathcal{A}[[M]]\rho$ is independent of ρ , and thus we denote it as $\mathcal{A}[[M]]$. We get the following important theorem for λ^{\rightarrow} .

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

Proof. Apply lemma 7.6 to the closed term M of type σ and to any arbitrary valuation ρ . \square

8 Pre-Applicative Structures for $\lambda^{\rightarrow, \times, +, \perp}$

In this section, the pre-applicative structures of section 3 are generalized to the types $\rightarrow, \times, +, \perp$. 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. For simplicity, in this section, we only present pre-applicative structures. The definition of the other structures is given in an appendix (see section 21). We also show that the term model can be viewed as a pre-applicative β -structures, and that the HRO models of Kreisel and Troelstra [16, 26] can be viewed as an applicative β -structure.

Definition 8.1 A *pre-applicative β -structure* is a structure

$$\mathcal{A} = \langle A, \mathbf{fun}, \mathbf{abst}, \Pi, \langle -, - \rangle, \mathbf{inl}, \mathbf{inr}, [-, -], \cdot, \nabla \rangle$$

where

- $A = (A^\sigma)_{\sigma \in \mathcal{T}}$ is a family of (nonempty) sets called *carriers*;
- $(\preceq^\sigma)_{\sigma \in \mathcal{T}}$ is a family of preorders, each \preceq^σ on A^σ ;
- $\mathbf{abst}^{\sigma, \tau}: [A^\sigma \Rightarrow A^\tau] \rightarrow A^{\sigma \rightarrow \tau}$, a family of partial operators;
- $\mathbf{fun}^{\sigma, \tau}: A^{\sigma \rightarrow \tau} \rightarrow [A^\sigma \Rightarrow A^\tau]$, a family of (total) operators;
- $\langle -, - \rangle^{\sigma, \tau}: A^\sigma \times A^\tau \rightarrow A^{\sigma \times \tau}$, a family of partial pairing operators;

$\Pi^{\sigma,\tau}: A^{\sigma \times \tau} \rightarrow A^\sigma \times A^\tau$, a family of (total) projection operators;
 $[-, -]^{\sigma,\tau,\delta}: A^{\sigma \rightarrow \delta} \times A^{\tau \rightarrow \delta} \rightarrow A^{(\sigma+\tau) \rightarrow \delta}$, a family of partial copairing operators;
 $\text{inl}^{\sigma,\tau}: A^\sigma \rightarrow A^{\sigma+\tau}$, a family of (total) operators;
 $\text{inr}^{\sigma,\tau}: A^\tau \rightarrow A^{\sigma+\tau}$, a family of (total) operators;
 $\nabla_\sigma: A^\perp \rightarrow A^\sigma$, is a family of (total) functions.

We define $\text{cinl}: A^{(\sigma+\tau) \rightarrow \delta} \rightarrow [A^\sigma \Rightarrow A^\delta]$, $\text{cinr}: A^{(\sigma+\tau) \rightarrow \delta} \rightarrow [A^\tau \Rightarrow A^\delta]$, and $\text{cinf}: A^{(\sigma+\tau) \rightarrow \delta} \rightarrow [A^\perp \Rightarrow A^\delta]$ as follows: For every $h \in A^{(\sigma+\tau) \rightarrow \delta}$,

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

for every $a \in A^\sigma$,

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

for every $b \in A^\tau$, and

$$\text{cinf}(h)(c) = \text{fun}(h)(\nabla_{\sigma+\tau}(c)),$$

for every $c \in A^\perp$.

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

- (1) $\text{fun}^{\sigma,\tau}(\text{abst}^{\sigma,\tau}(\varphi)) \succeq \varphi$, whenever $\text{abst}^{\sigma,\tau}(\varphi)$ is defined, for $\varphi \in [A^\sigma \Rightarrow A^\tau]$, and $\text{fun}^{\sigma,\tau}(\nabla_{\sigma \rightarrow \tau}(c)) \succeq \lambda a \in A^\sigma. \nabla_\tau(c)$, for $c \in A^\perp$;
- (2) $\Pi^{\sigma,\tau}(\langle a, b \rangle) \succeq (a, b)$, for all $a \in A^\sigma, b \in A^\tau$, whenever $\langle a, b \rangle$ is defined, and $\Pi^{\sigma,\tau}(\nabla_{\sigma \times \tau}(c)) \succeq (\nabla_\sigma(c), \nabla_\tau(c))$, for every $c \in A^\perp$;
- (3) $\text{cinl}([f, g]) \succeq \text{fun}(f)$, $\text{cinr}([f, g]) \succeq \text{fun}(g)$, and $\text{cinf}([f, g]) \succeq \nabla_\delta$, whenever $[f, g]$ is defined,

The operators fun induce (total) operators

$\text{fun}^{\sigma,\tau}: A^{\sigma \rightarrow \tau} \rightarrow [A^\sigma \Rightarrow A^\tau]$, such that, for every $f \in A^{\sigma \rightarrow \tau}$ and every $a \in A^\sigma$,

$$\text{app}^{\sigma,\tau}(f, a) = \text{fun}^{\sigma,\tau}(f)(a).$$

Then, condition (1) can be written as

- (1') $\text{app}^{\sigma,\tau}(\text{abst}^{\sigma,\tau}(\varphi), a) \succeq \varphi(a)$, for all $a \in A^\sigma$, and $\text{app}^{\sigma,\tau}(\nabla_{\sigma \rightarrow \tau}(c), a) \succeq \nabla_\tau(c)$, for every $a \in A^\sigma$ and every $c \in A^\perp$, and condition (3) can be rewritten as
- (3') $\text{cinl}([f, g])(a) \succeq \text{app}(f, a)$, for all $a \in A^\sigma$, $\text{cinr}([f, g])(b) \succeq \text{app}(g, b)$, for all $b \in A^\tau$, and $\text{cinf}([f, g])(c) \succeq \nabla_\delta(c)$, for all $c \in A^\perp$, whenever $[f, g]$ is defined, for $f \in A^{\sigma \rightarrow \delta}$ and $g \in A^{\tau \rightarrow \delta}$.

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

We say that a pre-applicative β -structure is an *applicative β -structure* iff in conditions (1)-(3), \succeq is replaced by the identity relation $=$.

We will omit superscripts whenever possible. We can think of the elements of A^\perp as error elements, and copies of these error elements exist at all types (given by the functions ∇_σ).

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

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

When \mathcal{A} is an applicative β -structure, then, in definition 8.1, conditions (1)-(3) amounts to

(1) $\mathbf{fun}^{\sigma,\tau} \circ \mathbf{abst}^{\sigma,\tau} = \mathbf{id}$ on the domain of \mathbf{abst} , and $\mathbf{fun}^{\sigma,\tau} \circ \nabla_{\sigma \rightarrow \tau} = \lambda a \in A^\sigma. \nabla_\tau$;

(2) $\Pi^{\sigma,\tau} \circ \langle -, - \rangle^{\sigma,\tau} = \mathbf{id}$ on the domain of $\langle -, - \rangle$, and $\Pi^{\sigma,\tau} \circ \nabla_{\sigma \times \tau} = \langle \nabla_\sigma, \nabla_\tau \rangle$;

(3) $\langle \mathbf{cInl}, \mathbf{cInr} \rangle \circ [-, -] = \mathbf{fun}^{\sigma,\delta} \times \mathbf{fun}^{\tau,\delta}$ on the domain of definition of $[-, -]$, and $\mathbf{cInf} \circ [-, -] = \lambda f \in A^{\sigma \rightarrow \delta}. \lambda g \in A^{\tau \rightarrow \delta}. \nabla_\delta$, where $\lambda f \in A^{\sigma \rightarrow \delta}. \lambda g \in A^{\tau \rightarrow \delta}. \nabla_\delta$ denotes the constant function from $A^{\sigma \rightarrow \delta} \times A^{\tau \rightarrow \delta}$ to $[A^\perp \Rightarrow A^\delta]$, whose value is ∇_δ for all $f \in A^{\sigma \rightarrow \delta}$ and $g \in A^{\tau \rightarrow \delta}$.

In view of (1), from (3), we get

$\langle \mathbf{cInl}, \mathbf{cInr} \rangle \circ ([-, -] \circ (\mathbf{abst}^{\sigma,\delta} \times \mathbf{abst}^{\tau,\delta})) = \mathbf{id}$ on the domain of definition of $[-, -] \circ (\mathbf{abst}^{\sigma,\delta} \times \mathbf{abst}^{\tau,\delta})$.

However, we have no left inverse to ∇_δ , and we don't have an analogous identity for \mathbf{cInf} .

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

Let us give an (important) example of a pre-applicative β -structure.

Definition 8.2 Let $A^\sigma = \Lambda_\sigma$ be the set of all typed λ -terms of type σ . We let \mathbf{app} , π_1 , π_2 , $\langle -, - \rangle$, \mathbf{Inl} , \mathbf{Inr} , $[-, -]$, ∇ , be the obvious constructs (for example, $\mathbf{app}(M, N) = MN$). Define $N \preceq M$ iff $M \xrightarrow{*}_\beta N$. The operator \mathbf{abst} is defined as in definition 3.2. The structure just defined is denoted as \mathcal{LT}_β .

Another interesting example is provided by an adaptation of the so-called HRO-models (hereditarily recursive operations), due to Kreisel and Troelstra [16, 26]. These models are based on the Kleene partial applicative structure provided by acceptable Gödel numberings of the partial recursive functions. Assume that we have such a Gödel numbering, and denote the partial recursive function of index e as φ_e . Recall that such a numbering induces a partial operation $\cdot: \mathbf{N} \times \mathbf{N} \rightarrow \mathbf{N}$ (where \mathbf{N} denotes the set of natural numbers) defined as follows: $m \cdot n = \varphi_m(n)$, whenever it is defined. A partial recursive function φ_e is recursive iff $\varphi_e(n)$ is defined for all $n \in \mathbf{N}$. We also assume that we have a given pairing function $p: \mathbf{N} \times \mathbf{N} \rightarrow \mathbf{N}$, with projection functions $j_1: \mathbf{N} \rightarrow \mathbf{N}$ and $j_2: \mathbf{N} \rightarrow \mathbf{N}$, such that $p(j_1(m), j_2(m)) = m$ for all $m \in \mathbf{N}$, $j_1(p(m, n)) = m$, and $j_2(p(m, n)) = n$, for all $m, n \in \mathbf{N}$. In the rest of this section, we ignore the type \perp .

Definition 8.3 We define an applicative structure as follows. Each A^σ is a set of pairs of the form $\langle n, \sigma \rangle$, where $n \in \mathbf{N}$, and we denote the subset $\{n \mid \langle n, \sigma \rangle \in A^\sigma\}$ of \mathbf{N} as $\mathit{dom}(A^\sigma)$.

Let $A^\sigma = \{\langle n, \sigma \rangle \mid n \in \mathbf{N}\}$, for every base type σ ,

$$A^{\sigma \rightarrow \tau} = \{\langle e, \sigma \rightarrow \tau \rangle \mid \varphi_e \text{ is total on } \mathit{dom}(A^\sigma)\},$$

$$A^{\sigma \times \tau} = \{ \langle n, \sigma \times \tau \rangle \mid \langle j_1(n), \sigma \rangle \in A^\sigma \quad \text{and} \quad \langle j_2(n), \tau \rangle \in A^\tau \},$$

and

$$A^{\sigma + \tau} = \{ \langle p(0, n), \sigma + \tau \rangle \mid \langle n, \sigma \rangle \in A^\sigma \} \cup \{ \langle p(1, n), \sigma + \tau \rangle \mid \langle n, \tau \rangle \in A^\tau \}.$$

The preorder on each A^σ is the identity relation.

We let $\mathbf{app}(\langle m, \sigma \rightarrow \tau \rangle, \langle n, \sigma \rangle) = \langle \varphi_m(n), \tau \rangle$, which is well-defined, by definition of $A^{\sigma \rightarrow \tau}$. Π and $\langle -, - \rangle$ have an obvious definition in terms of p, j_1 , and j_2 . We let $\mathbf{inl}(\langle n, \sigma \rangle) = \langle p(0, n), \sigma + \tau \rangle$, $\mathbf{inr}(\langle n, \tau \rangle) = \langle p(1, n), \sigma + \tau \rangle$, and $[\langle m, \sigma \rightarrow \delta \rangle, \langle n, \tau \rightarrow \delta \rangle]$ is defined as follows. Let ψ be the function defined such that $\psi(p(0, s)) = \varphi_m(s)$ for all $s \in \mathbf{N}$, and $\psi(p(1, t)) = \varphi_n(t)$ for all $t \in \mathbf{N}$. Since φ_m and φ_n are partial recursive functions, ψ is a partial recursive function, and we let

$$[\langle m, \sigma \rightarrow \delta \rangle, \langle n, \tau \rightarrow \delta \rangle] = \langle e, (\sigma + \tau) \rightarrow \delta \rangle,$$

where e is some designated index for ψ (some index e such that $\varphi_e = \psi$).

Note that $\mathbf{fun}: A^{\sigma \rightarrow \tau} \rightarrow [A^\sigma \Rightarrow A^\tau]$ is the function defined such that $\mathbf{fun}(\langle e, \sigma \rightarrow \tau \rangle)(\langle n, \sigma \rangle) = \langle \varphi_e(n), \tau \rangle$. We still need to define \mathbf{abst} .

For every $m \in \mathbf{N}$, for every $e \in \mathbf{N}$, index of a total recursive function of $m + 1$ arguments, for every finite sequence $\rho = \langle \rho_1, \dots, \rho_m \rangle$ of natural numbers, let $e[\rho]$ denote the function in $[A^\sigma \Rightarrow A^\tau]$ defined such that

$$e[\rho](\langle n, \sigma \rangle) = \langle \varphi_e(\rho_1, \dots, \rho_m, n), \tau \rangle,$$

provided that $\varphi_e(\rho_1, \dots, \rho_m, n) \in \text{dom}(A^\tau)$, for all $n \in \text{dom}(A^\sigma)$. Then, by the s - m - n -theorem,

$$\varphi_e(\rho_1, \dots, \rho_m, n) = \varphi_{s(e, m, \rho_1, \dots, \rho_m)}(n),$$

for all $n \in \mathbf{N}$, and we let $\mathbf{abst}(e[\rho]) = \langle s(e, m, \rho_1, \dots, \rho_m), \sigma \rightarrow \tau \rangle$. The above applicative structure is denoted as \mathcal{HRO} .

By an easy induction on types, we can show that A^σ is nonempty for every type σ . Indeed, each $A^{\sigma \rightarrow \tau}$ is nonempty, since constant functions are total recursive, and the other cases are trivial. In the definition of $[\langle m, \sigma \rightarrow \delta \rangle, \langle n, \tau \rightarrow \delta \rangle]$, since φ_m is total on $\text{dom}(A^{\sigma \rightarrow \delta})$ and φ_n is total on $\text{dom}(A^{\tau \rightarrow \delta})$, the function ψ is total on $\text{dom}(A^{(\sigma + \tau) \rightarrow \delta})$, and thus, $[\langle m, \sigma \rightarrow \delta \rangle, \langle n, \tau \rightarrow \delta \rangle]$ is well defined. We still need to check that $\mathbf{fun}(\mathbf{abst}(e[\rho])) = e[\rho]$ for every $e[\rho] \in [A^\sigma \Rightarrow A^\tau]$. For such a function $e[\rho]$,

$$\mathbf{fun}(\mathbf{abst}(e[\rho]))(\langle n, \sigma \rangle) = \langle \varphi_{s(e, m, \rho_1, \dots, \rho_m)}(n), \tau \rangle = \langle \varphi_e(\rho_1, \dots, \rho_m, n), \tau \rangle,$$

by the s - m - n -theorem, and thus, $\mathbf{fun}(\mathbf{abst}(e[\rho])) = e[\rho]$. The other conditions of definition 8.1 are easily verified. These structures are not extensional.

9 \mathcal{P} -Realizability for the Arrow, Product, Sum, and \perp Types

In this section, we extend the semantic notion of realizability defined in section 5 to the calculus $\lambda^{\rightarrow, \times, +, \perp}$. This time, the definition of realizability for the sum type requires the notion of a cover. First, it is necessary to extend definition 4.2 to take care of product and sum types.

Definition 9.1 Let \mathcal{A} be a pre-applicative structure and let \mathcal{P} be a family $\mathcal{P} = (P_\sigma)_{\sigma \in \mathcal{T}}$, where each P_σ is a nonempty subset of A^σ . A \mathcal{P} -cover algebra (or \mathcal{P} -Grothendieck topology) on \mathcal{A} is a family \mathbf{Cov} of binary relations \mathbf{Cov}_σ on $2^{A^\sigma} \times A^\sigma$ satisfying the following properties:

- (0) $\mathbf{Cov}_\sigma(C, M)$ implies $M \in P_\sigma$ (equivalently, $\mathcal{P}(M, \sigma)$).
- (1) If $\mathbf{Cov}(C, M)$, then C is a sieve on M (an ideal below M).
- (2) If $M \in P_\sigma$, then $\mathbf{Cov}(\{N \mid N \preceq M\}, M)$ ($M \in P_\sigma$ 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 $\mathbf{Cov}(\pi_1(M)) = \mathbf{triv}(\pi_1(M))$, $\mathbf{Cov}(\pi_2(M)) = \mathbf{triv}(\pi_2(M))$, and if $\mathbf{Cov}(C, M)$ and $\mathbf{Cov}(D, \pi_1(M))$ (resp. $\mathbf{Cov}(D, \pi_2(M))$) with C and D nontrivial, then for every $Q \in D$, there is some $M' \in C$ such that $Q \preceq \pi_1(M')$ (resp. $Q \preceq \pi_2(M')$).

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.

It is also necessary to extend definition 4.3 to take care of product types.

Definition 9.2 We say that $M \in A^\sigma$ is *simple* iff $\mathbf{Cov}(C, M)$ for at least two distinct covers C . We say that $M \in A^\sigma$ is *stubborn* iff $\mathbf{Cov}(M) = \mathbf{triv}(M)$ (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), $\pi_1(M)$, and $\pi_2(M)$ are simple.

Definition 4.4 is extended as follows.

Definition 9.3 Let $\langle \mathcal{A}, \mathcal{P}, \mathbf{Cov} \rangle$ be a \mathcal{P} -site. Properties (P1)-(P3) are defined as follows:

- (P1) $\mathcal{P}(M, \sigma)$, for some stubborn element $M \in A^\sigma$.
- (P2) If $\mathcal{P}(M, \sigma)$ and $M \succeq N$, then $\mathcal{P}(N, \sigma)$.
- (P3)
 - (1) If $\mathbf{Cov}_{\sigma \rightarrow \tau}(C, M)$, $\mathcal{P}(N, \sigma)$, and $\mathcal{P}(M'N, \tau)$ whenever $M' \in C$, then $\mathcal{P}(MN, \tau)$.
 - (2) If $\mathbf{Cov}_{\sigma \times \tau}(C, M)$, and $\mathcal{P}(\pi_1(M'), \sigma)$ and $\mathcal{P}(\pi_2(M'), \tau)$ whenever $M' \in C$, then $\mathcal{P}(\pi_1(M), \sigma)$ and $\mathcal{P}(\pi_2(M), \tau)$.

From now on, we only consider relations (families) \mathcal{P} satisfying the conditions of definition 9.3.

Note that (P3) still implies that if $M \in P_{\sigma \rightarrow \tau}$ is stubborn and $N \in P_\sigma$ is any element, then $MN \in P_\tau$ is stubborn. It also implies that if $M \in P_{\sigma \times \tau}$ is stubborn, then $\pi_1(M) \in P_\sigma$ is stubborn and $\pi_2(M) \in P_\tau$ is stubborn. This is a consequence of property (6) of definition 9.1.

Definition 4.5 remains unchanged. However, for the reader's convenience, it is repeated. Recall that \mathcal{T} denotes the set of simple types built up from the type constructors \rightarrow , \times , and $+$.

Definition 9.4 Let $\langle \mathcal{A}, \mathcal{P}, \text{Cov} \rangle$ be a \mathcal{P} -site. A function $\mathcal{S}: \mathcal{A} \rightarrow 2^{\mathcal{T}}$ has the *sheaf property* (or is a *\mathcal{P} -sheaf*) iff it satisfies the following conditions:

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

A function $\mathcal{S}: \mathcal{A} \rightarrow 2^{\mathcal{T}}$ as in definition 9.4 can also be viewed as a family $\mathcal{S} = (S_\sigma)_{\sigma \in \mathcal{T}}$, where $S_\sigma = \{M \in \mathcal{A} \mid \sigma \in \mathcal{S}(M)\}$. Then, the sets S_σ are called *\mathcal{P} -candidates*. The conditions of definition 9.4 are then stated as follows:

- (S1) $S_\sigma \subseteq P_\sigma$.
- (S2) If $M \in S_\sigma$ and $M \succeq N$, then $N \in S_\sigma$.
- (S3) If $\text{Cov}_\sigma(C, M)$, and $C \subseteq S_\sigma$, then $M \in S_\sigma$.

We now generalize the definition of realizers to take into accounts the types \times , $+$, and \perp . We define *\mathcal{P} -realizability* as follows.

Definition 9.5 Let $\langle \mathcal{A}, \mathcal{P}, \text{Cov} \rangle$ be a \mathcal{P} -site. The sets $\mathbf{r}[\sigma]$ of *realizers of σ* are defined as follows:

$$\begin{aligned} \mathbf{r}[\sigma] &= P_\sigma, & \sigma \text{ a base type,} \\ \mathbf{r}[\sigma \rightarrow \tau] &= \{M \mid M \in P_{\sigma \rightarrow \tau}, \text{ and for all } N, \text{ if } N \in \mathbf{r}[\sigma] \text{ then } MN \in \mathbf{r}[\tau]\}, \\ \mathbf{r}[\sigma \times \tau] &= \{M \mid M \in P_{\sigma \times \tau}, \pi_1(M) \in \mathbf{r}[\sigma], \text{ and } \pi_2(M) \in \mathbf{r}[\tau]\}, \\ \mathbf{r}[\sigma + \tau] &= \{M \mid \text{Cov}_{\sigma + \tau}(\{\text{inl}(M_1) \mid M_1 \in \mathbf{r}[\sigma] \text{ and } M \succeq \text{inl}(M_1)\} \cup \\ &\quad \{\text{inr}(M_2) \mid M_2 \in \mathbf{r}[\tau] \text{ and } M \succeq \text{inr}(M_2)\} \cup \\ &\quad \{\nabla_{\sigma + \tau}(M_3) \mid M_3 \in P_\perp \text{ and } M \succeq \nabla_{\sigma + \tau}(M_3)\}, M)\}. \end{aligned}$$

We now prove a generalization of lemma 5.2.

Lemma 9.6 *Given a scenic \mathcal{P} -site $\langle \mathcal{A}, \mathcal{P}, \text{Cov} \rangle$, if \mathcal{P} satisfies conditions (P1)-(P3), then the family $(\mathbf{r}[\sigma])_{\sigma \in \mathcal{T}}$ has the sheaf property, and each $\mathbf{r}[\sigma]$ contains all stubborn elements in P_σ .*

Proof. We proceed by induction on types. The base case is as in lemma 5.2. The induction step has more cases since we also need to deal with \times , $+$, and \perp .

- (S1). This is trivial by the definitions of $\mathbf{r}[\sigma \rightarrow \tau]$, $\mathbf{r}[\sigma \times \tau]$, and $\mathbf{r}[\sigma + \tau]$,
- (S2). There are three cases depending on the type.

1. Arrow type $\sigma \rightarrow \tau$. The proof is as in lemma 5.2.

2. Product type $\sigma \times \tau$. Assume that $M \succeq M'$ for $M \in \mathbf{r}[\sigma \times \tau]$. We need to prove that $M' \in P_{\sigma \times \tau}$, $\pi_1(M') \in \mathbf{r}[\sigma]$, and $\pi_2(M') \in \mathbf{r}[\tau]$. Since $M \in \mathbf{r}[\sigma \times \tau]$, by (S1), $M \in P_{\sigma \times \tau}$, and by (P2) $M' \in P_{\sigma \times \tau}$. Since $M \in \mathbf{r}[\sigma \times \tau]$, we have $\pi_1(M) \in \mathbf{r}[\sigma]$ and $\pi_2(M) \in \mathbf{r}[\tau]$. But by monotonicity, $\pi_1(M) \succeq \pi_1(M')$ and $\pi_2(M) \succeq \pi_2(M')$, and by the induction hypothesis, by (S2), we get $\pi_1(M') \in \mathbf{r}[\sigma]$ and $\pi_2(M') \in \mathbf{r}[\tau]$.

3. Sum type $\sigma + \tau$. Assume that $M \succeq M'$ for $M \in \mathbf{r}[\sigma + \tau]$. Since $M \in \mathbf{r}[\sigma + \tau]$, we have

$$\begin{aligned} & \text{Cov}_{\sigma+\tau}(\{\mathbf{inl}(M_1) \mid M_1 \in \mathbf{r}[\sigma] \text{ and } M \succeq \mathbf{inl}(M_1)\} \cup \\ & \quad \{\mathbf{inr}(M_2) \mid M_2 \in \mathbf{r}[\tau] \text{ and } M \succeq \mathbf{inr}(M_2)\} \cup \\ & \quad \{\nabla_{\sigma+\tau}(M_3) \mid M_3 \in P_{\perp} \text{ and } M \succeq \nabla_{\sigma+\tau}(M_3)\}, M). \end{aligned}$$

Consider the cover D of M :

$$\begin{aligned} D = & \{\mathbf{inl}(M_1) \mid M_1 \in \mathbf{r}[\sigma] \text{ and } M \succeq \mathbf{inl}(M_1)\} \cup \\ & \{\mathbf{inr}(M_2) \mid M_2 \in \mathbf{r}[\tau] \text{ and } M \succeq \mathbf{inr}(M_2)\} \cup \\ & \{\nabla_{\sigma+\tau}(M_3) \mid M_3 \in P_{\perp} \text{ and } M \succeq \nabla_{\sigma+\tau}(M_3)\}. \end{aligned}$$

By property (3) of definition 9.1, for any $M' \in D$, the set $\{Q \mid Q \in D, Q \preceq M'\}$ is a cover of M' . Now, if $M' \preceq M$, by property (1) of definition 9.1, $M' \in D$, and it is clear that

$$\begin{aligned} \{Q \mid Q \in D, Q \preceq M'\} = & \{\mathbf{inl}(M_1) \mid M_1 \in \mathbf{r}[\sigma] \text{ and } M' \succeq \mathbf{inl}(M_1)\} \cup \\ & \{\mathbf{inr}(M_2) \mid M_2 \in \mathbf{r}[\tau] \text{ and } M' \succeq \mathbf{inr}(M_2)\} \cup \\ & \{\nabla_{\sigma+\tau}(M_3) \mid M_3 \in P_{\perp} \text{ and } M' \succeq \nabla_{\sigma+\tau}(M_3)\}. \end{aligned}$$

Then, we have

$$\begin{aligned} & \text{Cov}_{\sigma+\tau}(\{\mathbf{inl}(M_1) \mid M_1 \in \mathbf{r}[\sigma] \text{ and } M' \succeq \mathbf{inl}(M_1)\} \cup \\ & \quad \{\mathbf{inr}(M_2) \mid M_2 \in \mathbf{r}[\tau] \text{ and } M' \succeq \mathbf{inr}(M_2)\} \cup \\ & \quad \{\nabla_{\sigma+\tau}(M_3) \mid M_3 \in P_{\perp} \text{ and } M' \succeq \nabla_{\sigma+\tau}(M_3)\}, M'). \end{aligned}$$

showing that $M' \in \mathbf{r}[\sigma + \tau]$.

(S3). Let M be simple. There are three cases depending on the type of M .

1. Arrow type $\sigma \rightarrow \tau$. The proof is as in lemma 5.2.

2. Product type $\sigma \times \tau$. Assume that $\text{Cov}_{\sigma \times \tau}(C, M)$ and that $M' \in \mathbf{r}[\sigma \times \tau]$ whenever $M' \in C$, where M is simple. By property (0) of definition 9.1, we have $M \in P_{\sigma \times \tau}$. We need to show that $\pi_1(M) \in \mathbf{r}[\sigma]$ and $\pi_2(M) \in \mathbf{r}[\tau]$.

If $M \in P_{\sigma \times \tau}$ is stubborn, we have shown that $\pi_1(M) \in P_{\sigma}$ is stubborn and that $\pi_2(M) \in P_{\tau}$ is stubborn. By the induction hypothesis, all stubborn elements in P_{σ} are in $\mathbf{r}[\sigma]$ and all stubborn elements in P_{τ} are in $\mathbf{r}[\tau]$. Thus, when M is stubborn, $\pi_1(M) \in \mathbf{r}[\sigma]$ and $\pi_2(M) \in \mathbf{r}[\tau]$.

Next, assume that M is not stubborn. Since $M' \in \mathbf{r}[\sigma \times \tau]$ whenever $M' \in C$, we have $\pi_1(M') \in \mathbf{r}[\sigma]$ and $\pi_2(M') \in \mathbf{r}[\tau]$. By (S1), we have $\pi_1(M') \in P_{\sigma}$, $\pi_2(M') \in P_{\tau}$, and by (P3)(2), we get $\pi_1(M) \in P_{\sigma}$ and $\pi_2(M) \in P_{\tau}$. If σ is a base type, then $\mathbf{r}[\sigma] = P_{\sigma}$ and $\pi_1(M) \in \mathbf{r}[\sigma]$. Similarly, if τ is a base type, then $\mathbf{r}[\tau] = P_{\tau}$ and $\pi_2(M) \in \mathbf{r}[\tau]$.

Let us now consider the case where σ is not a base type, the case where τ is not a base type being similar. Then, $\pi_1(M) \in P_{\sigma}$ and $\pi_1(M)$ is simple (since the site is scenic). We use (S3) to prove that $\pi_1(M) \in \mathbf{r}[\sigma]$. Assume that $\text{Cov}_{\sigma}(D, \pi_1(M))$ for any cover D of $\pi_1(M)$. The case where $\pi_1(M)$ is stubborn follows from the induction hypothesis. Otherwise, since $\text{Cov}_{\sigma \times \tau}(C, M)$ and C and D are nontrivial, by property (6) of definition 9.1, for every $Q \in D$, there is some $M' \in C$ such

that $Q \preceq \pi_1(M')$. By the assumption, $M' \in \mathbf{r}[\sigma \times \tau]$. This implies that $\pi_1(M') \in \mathbf{r}[\sigma]$, and by the induction hypothesis and (S2), we have $Q \in \mathbf{r}[\sigma]$. By (S3), we conclude that $\pi_1(M) \in \mathbf{r}[\sigma]$.

3. Sum type $\sigma + \tau$. Assume that $\mathbf{Cov}_{\sigma+\tau}(C, M)$ and that $N \in \mathbf{r}[\sigma + \tau]$ for every $N \in C$. Let

$$\begin{aligned} D = & \{\mathbf{inl}(M_1) \mid M_1 \in \mathbf{r}[\sigma] \text{ and } M \succeq \mathbf{inl}(M_1)\} \cup \\ & \{\mathbf{inr}(M_2) \mid M_2 \in \mathbf{r}[\tau] \text{ and } M \succeq \mathbf{inr}(M_2)\} \cup \\ & \{\nabla_{\sigma+\tau}(M_3) \mid M_3 \in P_{\perp} \text{ and } M \succeq \nabla_{\sigma+\tau}(M_3)\}. \end{aligned}$$

Using the properties of \preceq , it is clear that D is a sieve on M . We need to prove that $\mathbf{Cov}_{\sigma+\tau}(D, M)$, since this is equivalent to $M \in \mathbf{r}[\sigma + \tau]$. Let $N \in C$, and consider the set $\{Q \mid Q \in D, Q \preceq N\}$. We prove that $\mathbf{Cov}(\{Q \mid Q \in D, Q \preceq N\}, N)$. However, since $N \in C$ and by assumption, $N \in \mathbf{r}[\sigma + \tau]$ for every $N \in C$, we have

$$\begin{aligned} \mathbf{Cov}_{\sigma+\tau}(\{\mathbf{inl}(M_1) \mid M_1 \in \mathbf{r}[\sigma] \text{ and } N \succeq \mathbf{inl}(M_1)\} \cup \\ \{\mathbf{inr}(M_2) \mid M_2 \in \mathbf{r}[\tau] \text{ and } N \succeq \mathbf{inr}(M_2)\} \cup \\ \{\nabla_{\sigma+\tau}(M_3) \mid M_3 \in P_{\perp} \text{ and } N \succeq \nabla_{\sigma+\tau}(M_3)\}, N). \end{aligned}$$

Since $N \preceq M$, it is clear that

$$\begin{aligned} \{Q \mid Q \in D, Q \preceq N\} = & \{\mathbf{inl}(M_1) \mid M_1 \in \mathbf{r}[\sigma] \text{ and } N \succeq \mathbf{inl}(M_1)\} \cup \\ & \{\mathbf{inr}(M_2) \mid M_2 \in \mathbf{r}[\tau] \text{ and } N \succeq \mathbf{inr}(M_2)\} \cup \\ & \{\nabla_{\sigma+\tau}(M_3) \mid M_3 \in P_{\perp} \text{ and } N \succeq \nabla_{\sigma+\tau}(M_3)\}. \end{aligned}$$

Then, by property (4) of definition 9.1, we have $\mathbf{Cov}_{\sigma+\tau}(D, M)$, that is, $M \in \mathbf{r}[\sigma + \tau]$. \square

We also need to extend definition 6.1 to give an interpretation to the new terms.

10 Interpreting λ -Terms in $\lambda^{\rightarrow, \times, +, \perp}$

We extend definition 6.1 to take care of \times , $+$, and \perp .

Definition 10.1 Given a pre-applicative structure \mathcal{A} , a *valuation*, or *environment*, is any function $\rho: \mathcal{X} \rightarrow \mathcal{A}$, such that $\rho(x) \in A^\sigma$ if $x: \sigma$. A *meaning function* for \mathcal{A} is a partial function $\mathcal{A}[\![-]\!-](-)$ from pairs of (α -equivalence classes of) terms and valuations to \mathcal{A} , such that $\mathcal{A}[\![-]\!-](\rho)$ is defined whenever $M: \sigma$, in which case $\mathcal{A}[\![-]\!-](\rho) \in A^\sigma$. In addition, a meaning function satisfies the following conditions:

$$\mathcal{A}[\![-]\!-](\rho) = \rho(x)$$

$$\mathcal{A}[\![-]\!-](\rho) = \mathbf{app}(\mathcal{A}[\![-]\!-](\rho), \mathcal{A}[\![-]\!-](\rho))$$

$$\mathcal{A}[\![-]\!-](\rho) = \mathbf{abst}(f),$$

where f is the function defined such that,

$$f(a) = \mathcal{A}[\![-]\!-](\rho[x := a]), \text{ for every } a \in A^\sigma$$

$$\mathcal{A}[\![-]\!-](\rho) = \pi_1(\mathcal{A}[\![-]\!-](\rho))$$

$$\mathcal{A}[\![-]\!-](\rho) = \pi_2(\mathcal{A}[\![-]\!-](\rho))$$

$$\begin{aligned}
\mathcal{A}[\langle M_1, M_2 \rangle] \rho &= \langle \mathcal{A}[M_1] \rho, \mathcal{A}[M_2] \rho \rangle \\
\mathcal{A}[\text{inl}(M)] \rho &= \text{inl}(\mathcal{A}[M] \rho) \\
\mathcal{A}[\text{inr}(M)] \rho &= \text{inr}(\mathcal{A}[M] \rho) \\
\mathcal{A}[[M, N]] \rho &= [\mathcal{A}[M] \rho, \mathcal{A}[N] \rho] \\
\mathcal{A}[\nabla_\sigma(M)] \rho &= \nabla_\sigma(\mathcal{A}[M] \rho).
\end{aligned}$$

It is routine to show that the following property holds:

$$\mathcal{A}[M] \rho_1 = \mathcal{A}[M] \rho_2, \text{ whenever } \rho_1(x) = \rho_2(x) \text{ for every } x \in FV(M) \text{ (independence)}$$

If we consider the pre-applicative structure $\mathcal{A} = \mathcal{L}\mathcal{T}_\beta$, then a valuation ρ is a substitution with an infinite domain. Using an induction on the structure of terms, it is easily verified that $\mathcal{L}\mathcal{T}_\beta[M] \rho = M[\varphi]$, where φ is the substitution defined by the restriction of ρ to $FV(M)$.

As far as realizability is concerned, if $M:\sigma$, then $\mathcal{L}\mathcal{T}_\beta[M] \rho$ is a typed λ -term realizing σ . Definition 9.5 is then a variant of Kreisel's modified realizability.

It is also interesting to see what happens if we try to interpret terms in the applicative structure $\mathcal{H}\mathcal{R}\mathcal{O}$ of definition 8.3. A valuation is a function ρ such that $\rho(x) = \langle k, \sigma \rangle$ for every $x:\sigma$, where $k \in \mathbf{N}$. Thus, given a term M such that $FV(M) = \{x_1:\sigma_1, \dots, x_m:\sigma_m\}$, a valuation ρ defines a finite sequence $\langle \rho_1, \dots, \rho_m \rangle$ of natural numbers, where $\rho_i = \rho(x_i)$. It is easily shown by induction on the structure of $M:\sigma$ that $\mathcal{H}\mathcal{R}\mathcal{O}[M] \rho = \langle \varphi_e(\rho_1, \dots, \rho_m), \sigma \rangle$, where e is the index a total recursive function φ_e in the arguments $\langle \rho_1, \dots, \rho_m \rangle$. Thus, every typed λ -terms can be interpreted in $\mathcal{H}\mathcal{R}\mathcal{O}$, and $\mathcal{H}\mathcal{R}\mathcal{O}[M] \rho$ is given by a function recursive in the restriction of ρ to $FV(M)$. As far as realizability is concerned, if $M:\sigma$, then $\mathcal{H}\mathcal{R}\mathcal{O}[M] \rho \in \mathbf{r}[\sigma]$ yields a realizer for σ which is given by a recursive function of ρ . In this case, definition 9.5 is equivalent to Kleene's recursive realizability (for \rightarrow , \times , and $+$).

11 The Realizability Theorem For $\lambda^{\rightarrow, \times, +, \perp}$

In this section, we generalize the realizability lemma (lemma 7.6) and its main corollary (theorem 7.7) to the calculus $\lambda^{\rightarrow, \times, +, \perp}$. In order to do so, we need to add conditions to definition 7.1 to take care of \times , $+$, and \perp .

Definition 11.1 We say that a site $\langle \mathcal{A}, \mathcal{P}, \text{Cov} \rangle$ is *well-behaved* iff the following conditions hold:

- (1) For any $a \in A^\sigma$, any $\varphi \in [A^\sigma \Rightarrow A^\tau]$, if $\mathbf{abst}(\varphi)$ exists, $\text{Cov}_\tau(C, \mathbf{app}(\mathbf{abst}(\varphi), a))$, and C is a nontrivial cover, then $c \preceq \varphi(a)$ for every $c \in C$;
For any $a \in A^\perp$, any $b \in A^\sigma$, if $\text{Cov}_\tau(C, \mathbf{app}(\nabla_{\sigma \rightarrow \tau}(a), b))$ and C is a nontrivial cover, then $c \preceq \nabla_\tau(a)$ for every $c \in C$;
- (2) If $\text{Cov}_\sigma(C, \pi_1(\langle a_1, a_2 \rangle))$ and C is a nontrivial cover, then $c \preceq a_1$ for every $c \in C$.
If $\text{Cov}_\tau(C, \pi_2(\langle a_1, a_2 \rangle))$ and C is a nontrivial cover, then $c \preceq a_2$ for every $c \in C$.
If $\text{Cov}_\sigma(C, \pi_1(\nabla_{\sigma \times \tau}(a)))$ and C is a nontrivial cover, then $c \preceq \nabla_\sigma(a)$ for every $c \in C$.
If $\text{Cov}_\tau(C, \pi_2(\nabla_{\sigma \times \tau}(a)))$ and C is a nontrivial cover, then $c \preceq \nabla_\tau(a)$ for every $c \in C$.

- (3) If $\text{Cov}(p) = \text{triv}(p)$, then $\text{Cov}(\text{app}([f, g], p)) = \text{triv}(\text{app}([f, g], p))$, and if $\text{Cov}_{\sigma+\tau}(C, p)$, $\text{Cov}_\delta(D, \text{app}([f, g], p))$, and C and D are nontrivial, then for every $d \in D$, either there is some $\text{inl}(p_1) \in C$ such that $d \preceq \text{app}(f, p_1)$, or there is some $\text{inr}(p_2) \in C$ such that $d \preceq \text{app}(g, p_2)$, or there is some $\nabla_{\sigma+\tau}(p_3) \in C$ such that $d \preceq \nabla_\delta(p_3)$, where $f \in A^{\sigma \rightarrow \delta}$ and $g \in A^{\tau \rightarrow \delta}$.

In view of definition 10.1, definition 11.1 implies the following conditions.

Definition 11.2 Given a meaning function $\mathcal{A}-$ on the pre-applicative structure \mathcal{A} , condition (1)-(3) are defined as follows:

- (1) For any $a \in A^\sigma$, if $\text{Cov}_\tau(C, \text{app}(\mathcal{A}[\lambda x: \sigma. M]\rho, a))$ and C is a nontrivial cover, then $c \preceq \mathcal{A}[M]\rho[x := a]$ for every $c \in C$.
 For any $b \in A^\sigma$, if $\text{Cov}_\tau(C, \text{app}(\mathcal{A}[\nabla_{\sigma \rightarrow \tau}(M)]\rho, b))$ and C is a nontrivial cover, then $c \preceq \mathcal{A}[\nabla_\tau(M)]\rho$ for every $c \in C$;
- (2) If $\text{Cov}_\sigma(C, \pi_1(\mathcal{A}[\langle M_1, M_2 \rangle]\rho))$ and C is a nontrivial cover, then $c \preceq \mathcal{A}[M_1]\rho$ for every $c \in C$.
 If $\text{Cov}_\tau(C, \pi_2(\mathcal{A}[\langle M_1, M_2 \rangle]\rho))$ and C is a nontrivial cover, then $c \preceq \mathcal{A}[M_2]\rho$ for every $c \in C$.
 If $\text{Cov}_\sigma(C, \pi_1(\mathcal{A}[\nabla_{\sigma \times \tau}(M)]\rho))$ and C is a nontrivial cover, then $c \preceq \mathcal{A}[\nabla_\sigma(M)]\rho$ for every $c \in C$.
 If $\text{Cov}_\tau(C, \pi_2(\mathcal{A}[\nabla_{\sigma \times \tau}(M)]\rho))$ and C is a nontrivial cover, then $c \preceq \mathcal{A}[\nabla_\tau(M)]\rho$ for every $c \in C$.
- (3) If $\text{Cov}(p) = \text{triv}(p)$, then $\text{Cov}(\text{app}(\mathcal{A}[[M, N]]\rho, p)) = \text{triv}(\text{app}(\mathcal{A}[[M, N]]\rho, p))$, and if $\text{Cov}_{\sigma+\tau}(C, p)$, $\text{Cov}_\delta(D, \text{app}(\mathcal{A}[[M, N]]\rho, p))$, and C and D are nontrivial, then for every $d \in D$, either there is some $\text{inl}(p_1) \in C$ such that $d \preceq \text{app}(\mathcal{A}[M]\rho, p_1)$, or there is some $\text{inr}(p_2) \in C$ such that $d \preceq \text{app}(\mathcal{A}[N]\rho, p_2)$, or there is some $\nabla_{\sigma+\tau}(p_3) \in C$ such that $d \preceq \nabla_\delta(p_3)$.

We also need to add conditions to definition 7.4 to take care of \times , $+$, and \perp .

Definition 11.3 Given a well-behaved site $\langle \mathcal{A}, \mathcal{P}, \text{Cov} \rangle$, properties (P4) and (P5) are defined as follows:

(P4)

- (1) For every $a \in A^\sigma$, if $\varphi(a) \in P_\tau$, where $\varphi \in [A^\sigma \Rightarrow A^\tau]$ and $\text{abst}(\varphi)$ exists, then $\text{abst}(\varphi) \in P_{\sigma \rightarrow \tau}$.
- (2) If $a_1 \in P_\sigma$ and $a_2 \in P_\tau$, then $\langle a_1, a_2 \rangle \in P_{\sigma \times \tau}$.
- (3) If $a \in P_\sigma$, then $\text{inl}(a) \in P_{\sigma+\tau}$, and if $a \in P_\tau$, then $\text{inr}(a) \in P_{\sigma+\tau}$.
- (4) If $a_1 \in P_{\sigma \rightarrow \delta}$ and $a_2 \in P_{\sigma \rightarrow \tau}$, then $[a_1, a_2] \in P_{(\sigma+\tau) \rightarrow \delta}$.
- (5) If $a \in P_\perp$, then $\nabla_\sigma(a) \in P_\sigma$.

(P5)

- (1) If $a \in P_\sigma$ and $\varphi(a) \in P_\tau$, where $\varphi \in [A^\sigma \Rightarrow A^\tau]$ and $\text{abst}(\varphi)$ exists, then $\text{app}(\text{abst}(\varphi), a) \in P_\tau$.
- (2) If $a_1 \in P_\sigma$ and $a_2 \in P_\tau$, then $\pi_1(\langle a_1, a_2 \rangle) \in P_\sigma$ and $\pi_2(\langle a_1, a_2 \rangle) \in P_\tau$.

- (3) If $\text{Cov}_{\sigma+\tau}(C, p)$, $f \in P_{\sigma \rightarrow \delta}$, $g \in P_{\tau \rightarrow \delta}$, $\text{app}(f, p_1) \in P_\delta$ whenever $\text{inl}(p_1) \in C$, $\text{app}(g, p_2) \in P_\delta$ whenever $\text{inr}(p_2) \in C$, and $p_3 \in P_\perp$ whenever $\nabla_{\sigma+\tau}(p_3) \in C$, then $\text{app}([f, g], p) \in P_\delta$.
- (4) If $a \in P_\perp$ and $b \in P_\sigma$, then $\text{app}(\nabla_{\sigma \rightarrow \tau}(a), b) \in P_\tau$.
If $a \in P_\perp$, then $\pi_1(\nabla_{\sigma \times \tau}(a)) \in P_\sigma$ and $\pi_2(\nabla_{\sigma \times \tau}(a)) \in P_\tau$.

It is easy to verify that $\text{app}([f, g], p) \in P_\delta$ is stubborn if $p \in P_{\sigma+\tau}$ is stubborn, $f \in P_{\sigma \rightarrow \delta}$, and $g \in P_{\tau \rightarrow \delta}$. This follows from condition (3) of definition 11.1.

In view of definition 10.1, definition 11.3 implies the following conditions.

Definition 11.4 Given a meaning function $\mathcal{A}[\![-]\!-](-)$ on the pre-applicative structure \mathcal{A} , conditions (P4)-(P5) are defined as follows:

(P4)

- (1) If $\mathcal{A}[M]\rho \in P_\tau$, then $\mathcal{A}[\lambda x: \sigma. M]\rho \in P_{\sigma \rightarrow \tau}$.
- (2) If $\mathcal{A}[M]\rho \in P_\sigma$ and $\mathcal{A}[N]\rho \in P_\tau$, then $\mathcal{A}[\langle M, N \rangle]\rho \in P_{\sigma \times \tau}$.
- (3) If $\mathcal{A}[M]\rho \in P_\sigma$, then $\text{inl}(\mathcal{A}[M]\rho) \in P_{\sigma+\tau}$, and if $\mathcal{A}[M]\rho \in P_\tau$, then $\text{inr}(\mathcal{A}[M]\rho) \in P_{\sigma+\tau}$.
- (4) If $\mathcal{A}[M]\rho \in P_{\sigma \rightarrow \delta}$ and $\mathcal{A}[N]\rho \in P_{\tau \rightarrow \delta}$, then $\mathcal{A}[\langle M, N \rangle]\rho \in P_{(\sigma+\tau) \rightarrow \delta}$.
- (5) If $\mathcal{A}[M]\rho \in P_\perp$, then $\mathcal{A}[\nabla_\sigma(M)]\rho \in P_\sigma$.

(P5)

- (1) If $a \in P_\sigma$ and $\mathcal{A}[M]\rho[x := a] \in P_\tau$, then $\text{app}(\mathcal{A}[\lambda x: \sigma. M]\rho, a) \in P_\tau$.
- (2) If $\mathcal{A}[M]\rho \in P_\sigma$ and $\mathcal{A}[N]\rho \in P_\tau$, then $\pi_1(\mathcal{A}[\langle M, N \rangle]\rho) \in P_\sigma$ and $\pi_2(\mathcal{A}[\langle M, N \rangle]\rho) \in P_\tau$.
- (3) If $\text{Cov}_{\sigma+\tau}(C, p)$, $\mathcal{A}[M]\rho \in P_{\sigma \rightarrow \delta}$, $\mathcal{A}[N]\rho \in P_{\tau \rightarrow \delta}$, $\text{app}(\mathcal{A}[M]\rho, p_1) \in P_\delta$ whenever $\text{inl}(p_1) \in C$, and $\text{app}(\mathcal{A}[N]\rho, p_2) \in P_\delta$ whenever $\text{inr}(p_2) \in C$, and $p_3 \in P_\perp$ whenever $\nabla_{\sigma+\tau}(p_3) \in C$, then $\text{app}(\mathcal{A}[\langle M, N \rangle]\rho, p) \in P_\delta$.
- (4) If $\mathcal{A}[M]\rho \in P_\perp$ and $b \in P_\sigma$, then $\text{app}(\mathcal{A}[\nabla_{\sigma \rightarrow \tau}(M)]\rho, b) \in P_\tau$.
If $\mathcal{A}[M]\rho \in P_\perp$, then $\pi_1(\mathcal{A}[\nabla_{\sigma \times \tau}(M)]\rho) \in P_\sigma$ and $\pi_2(\mathcal{A}[\nabla_{\sigma \times \tau}(M)]\rho) \in P_\tau$.

We have the following generalization of lemma 7.5.

Lemma 11.5 *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 ρ , the following properties hold: (1) If $\rho(y) \in \mathbf{r}[\gamma]$ for every $y: \gamma \in \text{FV}(M)$, and for every a , ($a \in \mathbf{r}[\sigma]$ implies $\mathcal{A}[M]\rho[x := a] \in \mathbf{r}[\tau]$), then $\mathcal{A}[\lambda x: \sigma. M]\rho \in \mathbf{r}[\sigma \rightarrow \tau]$. (2) If $\mathcal{A}[M]\rho \in \mathbf{r}[\sigma]$ and $\mathcal{A}[N]\rho \in \mathbf{r}[\tau]$, then $\mathcal{A}[\langle M, N \rangle]\rho \in \mathbf{r}[\sigma \times \tau]$; (3) If $\mathcal{A}[M]\rho \in \mathbf{r}[\sigma \rightarrow \delta]$, and $\mathcal{A}[N]\rho \in \mathbf{r}[\tau \rightarrow \delta]$, then $\mathcal{A}[\langle M, N \rangle]\rho \in \mathbf{r}[(\sigma + \tau) \rightarrow \delta]$. (4) If $a \in P_\perp$, then $\nabla_\sigma(a) \in \mathbf{r}[\sigma]$ for every σ .*

Proof. It is similar to the proof of lemma 7.5, except that we need to prove more clauses. By lemma 9.6, we know that the sets of the form $\mathbf{r}[\sigma]$ have the properties (S1)-(S3).

(1) This has already been proved in lemma 7.5.

(2) We need to show that $\mathcal{A}[\langle M, N \rangle]\rho \in P_{\sigma \times \tau}$, $\pi_1(\mathcal{A}[\langle M, N \rangle]\rho) \in \mathbf{r}[\sigma]$, and $\pi_2(\mathcal{A}[\langle M, N \rangle]\rho) \in \mathbf{r}[\tau]$. Since $\mathcal{A}[M]\rho \in \mathbf{r}[\sigma]$ and $\mathcal{A}[N]\rho \in \mathbf{r}[\tau]$, by (S1), $\mathcal{A}[M]\rho \in P_\sigma$ and $\mathcal{A}[N]\rho \in P_\tau$. By (P4)(2), we get $\mathcal{A}[\langle M, N \rangle]\rho \in P_{\sigma \times \tau}$. By (P5)(2), we also have $\pi_1(\mathcal{A}[\langle M, N \rangle]\rho) \in P_\sigma$ and

$\pi_2(\mathcal{A}[\langle M, N \rangle]\rho) \in P_\tau$. If σ is a base type then $\mathbf{r}[\sigma] = P_\sigma$ and $\pi_1(\mathcal{A}[\langle M, N \rangle]\rho) \in \mathbf{r}[\sigma]$. Similarly, if τ is a base type then $\mathbf{r}[\tau] = P_\tau$ and $\pi_2(\mathcal{A}[\langle M, N \rangle]\rho) \in \mathbf{r}[\tau]$.

If both σ and τ are nonbase types, $\pi_1(\mathcal{A}[\langle M, N \rangle]\rho) \in P_\sigma$ and $\pi_2(\mathcal{A}[\langle M, N \rangle]\rho) \in P_\tau$ are simple (since the site is scenic). We prove that $\pi_1(\mathcal{A}[\langle M, N \rangle]\rho) \in \mathbf{r}[\sigma]$ and $\pi_2(\mathcal{A}[\langle M, N \rangle]\rho) \in \mathbf{r}[\tau]$ using (S3). We consider the case of $\pi_1(\mathcal{A}[\langle M, N \rangle]\rho)$, the case of $\pi_2(\mathcal{A}[\langle M, N \rangle]\rho)$ being similar. The case where $\pi_1(\mathcal{A}[\langle M, N \rangle]\rho)$ is stubborn is trivial. Otherwise, assume that $\text{Cov}_\sigma(C, \pi_1(\mathcal{A}[\langle M, N \rangle]\rho))$, where C is a nontrivial cover. We need to prove that $c \in \mathbf{r}[\sigma]$ whenever $c \in C$. By condition (2) of definition 11.2, $c \preceq \mathcal{A}[[M]\rho]$ for every $c \in C$. Since $\mathcal{A}[[M]\rho] \in \mathbf{r}[\sigma]$ and $c \preceq \mathcal{A}[[M]\rho$, by (S2), we have $c \in \mathbf{r}[\sigma]$.

(3) We need to prove that $\mathcal{A}[[M, N]]\rho \in P_{(\sigma+\tau) \rightarrow \delta}$, and that $\mathbf{app}(\mathcal{A}[[M, N]]\rho, p) \in \mathbf{r}[\delta]$, for every $p \in \mathbf{r}[\sigma + \tau]$. Since $\mathcal{A}[[M]\rho] \in \mathbf{r}[\sigma \rightarrow \delta]$ and $\mathcal{A}[[N]\rho] \in \mathbf{r}[\tau \rightarrow \delta]$, by (S2), we have $\mathcal{A}[[M]\rho] \in P_{\sigma \rightarrow \delta}$ and $\mathcal{A}[[N]\rho] \in P_{\tau \rightarrow \delta}$, and by (P4)(4), we get $\mathcal{A}[[M, N]]\rho \in P_{(\sigma+\tau) \rightarrow \delta}$.

Next, we prove that $\mathbf{app}(\mathcal{A}[[M, N]]\rho, p) \in P_\delta$. Assume that the hypothesis of (3) holds. By assumption, $p \in \mathbf{r}[\sigma + \tau]$, $\mathcal{A}[[M]\rho] \in \mathbf{r}[\sigma \rightarrow \delta]$, and $\mathcal{A}[[N]\rho] \in \mathbf{r}[\tau \rightarrow \delta]$. By (S1), we have $p \in P_{\sigma+\tau}$, $\mathcal{A}[[M]\rho] \in P_{\sigma \rightarrow \delta}$, and $\mathcal{A}[[N]\rho] \in P_{\tau \rightarrow \delta}$. If p is stubborn, we have shown that $\mathbf{app}(\mathcal{A}[[M, N]]\rho, p) \in P_\delta$ is stubborn, and thus $\mathbf{app}(\mathcal{A}[[M, N]]\rho, p) \in \mathbf{r}[\delta]$ by (S3).

Otherwise, since $p \in \mathbf{r}[\sigma + \tau]$, the cover C given by

$$\begin{aligned} C = & \{\mathbf{inl}(p_1) \mid p_1 \in \mathbf{r}[\sigma] \text{ and } p \succeq \mathbf{inl}(p_1)\} \cup \\ & \{\mathbf{inr}(p_2) \mid p_2 \in \mathbf{r}[\tau] \text{ and } p \succeq \mathbf{inr}(p_2)\} \cup \\ & \{\nabla_{\sigma+\tau}(p_3) \mid p_3 \in P_\perp \text{ and } p \succeq \nabla_{\sigma+\tau}(p_3)\} \end{aligned}$$

is a nontrivial cover, and $\text{Cov}_{\sigma+\tau}(C, p)$. Then, since by the assumptions of the lemma, $\mathcal{A}[[M]\rho] \in \mathbf{r}[\sigma \rightarrow \delta]$ and $\mathcal{A}[[N]\rho] \in \mathbf{r}[\tau \rightarrow \delta]$, we have $\mathbf{app}(\mathcal{A}[[M]\rho, p_1) \in \mathbf{r}[\delta]$ whenever $\mathbf{inl}(p_1) \in C$, $\mathbf{app}(\mathcal{A}[[N]\rho, p_2) \in \mathbf{r}[\delta]$ whenever $\mathbf{inr}(p_2) \in C$, and $p_3 \in P_\perp$ whenever $\nabla_{\sigma+\tau}(p_3) \in C$, since $p_1 \in \mathbf{r}[\sigma]$, $p_2 \in \mathbf{r}[\tau]$, and $p_3 \in P_\perp$, by definition of C . Now (using S1), the conditions of (P5)(3) are met for C , and we have $\mathbf{app}(\mathcal{A}[[M, N]]\rho, p) \in P_\delta$. If δ is a base type, then $\mathbf{r}[\delta] = P_\delta$, and $\mathbf{app}(\mathcal{A}[[M, N]]\rho, p) \in \mathbf{r}[\delta]$.

If δ is not a base type, then $\mathbf{app}(\mathcal{A}[[M, N]]\rho, p)$ is simple (since the site is scenic). We use (S3) to prove that $\mathbf{app}(\mathcal{A}[[M, N]]\rho, p) \in \mathbf{r}[\delta]$. The case where $\mathbf{app}(\mathcal{A}[[M, N]]\rho, p)$ is stubborn is trivial.

Otherwise, assume that $\text{Cov}_\delta(D, \mathbf{app}(\mathcal{A}[[M, N]]\rho, p))$, where D is a nontrivial cover. Since $p \in \mathbf{r}[\sigma + \tau]$, the cover C given by

$$\begin{aligned} C = & \{\mathbf{inl}(p_1) \mid p_1 \in \mathbf{r}[\sigma] \text{ and } p \succeq \mathbf{inl}(p_1)\} \cup \\ & \{\mathbf{inr}(p_2) \mid p_2 \in \mathbf{r}[\tau] \text{ and } p \succeq \mathbf{inr}(p_2)\} \cup \\ & \{\nabla_{\sigma+\tau}(p_3) \mid p_3 \in P_\perp \text{ and } p \succeq \nabla_{\sigma+\tau}(p_3)\} \end{aligned}$$

is a nontrivial cover, and $\text{Cov}_{\sigma+\tau}(C, p)$. Since C and D are nontrivial, by condition (3) of definition 11.2, for every $d \in D$, either there is some $\mathbf{inl}(p_1) \in C$ such that $d \preceq \mathbf{app}(\mathcal{A}[[M]\rho, p_1)$, or there is some $\mathbf{inr}(p_2) \in C$ such that $d \preceq \mathbf{app}(\mathcal{A}[[N]\rho, p_2)$, or there is some $\nabla_{\sigma+\tau}(p_3) \in C$ such that $d \preceq \nabla_\delta(p_3)$.

In the first two cases, since by definition of C , $p_1 \in \mathbf{r}[\sigma]$ and $p_2 \in \mathbf{r}[\tau]$, and by assumption, $\mathcal{A}[M]\rho \in \mathbf{r}[\sigma \rightarrow \delta]$ and $\mathcal{A}[N]\rho \in \mathbf{r}[\tau \rightarrow \delta]$, we have $\mathbf{app}(\mathcal{A}[M]\rho, p_1) \in \mathbf{r}[\delta]$ and $\mathbf{app}(\mathcal{A}[N]\rho, p_2) \in \mathbf{r}[\delta]$, and by (S2), we get $d \in \mathbf{r}[\delta]$. In the third case, by definition of C , we have $p_3 \in P_\perp$, and by (4) (of this lemma, to be proved next), we have $\nabla_\delta(p_3) \in \mathbf{r}[\delta]$. Then, by (S2), in all cases we get $d \in \mathbf{r}[\delta]$. Finally, by (S3), we have $\mathbf{app}(\mathcal{A}[[M, N]]\rho, p) \in \mathbf{r}[\delta]$.

(4) We proceed by induction on σ . When σ is a base type, since $\nabla_\sigma(M) \in P_\sigma$ by (P4)(5) and since $\mathbf{r}[\sigma] = P_\sigma$, we have $\nabla_\sigma(M) \in \mathbf{r}[\sigma]$.

1. Arrow type $\sigma \rightarrow \tau$. We prove that $\mathbf{app}(\nabla_{\sigma \rightarrow \tau}(a), b) \in \mathbf{r}[\tau]$ for every $b \in \mathbf{r}[\sigma]$. Since $a \in P_\perp$ and by (S1) $b \in P_\sigma$, by (P5)(4), we have $\mathbf{app}(\nabla_{\sigma \rightarrow \tau}(a), b) \in P_\tau$. If τ is a base type, $\mathbf{r}[\tau] = P_\tau$ and $\mathbf{app}(\nabla_{\sigma \rightarrow \tau}(a), b) \in \mathbf{r}[\tau]$. Otherwise, $\mathbf{app}(\nabla_{\sigma \rightarrow \tau}(a), b) \in P_\tau$ is a simple term and we use (S3). The case where $\mathbf{app}(\nabla_{\sigma \rightarrow \tau}(a), b)$ is stubborn is trivial. Otherwise, assume that $\mathbf{Cov}_\tau(C, \mathbf{app}(\nabla_{\sigma \rightarrow \tau}(a), b))$ for some nontrivial cover C . Then, by condition (1) of definition 11.1, $c \preceq \nabla_\tau(a)$ for every $c \in C$; By the induction hypothesis, $\nabla_\tau(a) \in \mathbf{r}[\tau]$, and by (S2), we have $c \in \mathbf{r}[\tau]$. Thus, by (S3), we have $\mathbf{app}(\nabla_{\sigma \rightarrow \tau}(a), b) \in \mathbf{r}[\tau]$.

2. Product type $\sigma \times \tau$. We prove that $\pi_1(\nabla_{\sigma \times \tau}(a)) \in \mathbf{r}[\sigma]$ and $\pi_2(\nabla_{\sigma \times \tau}(a)) \in \mathbf{r}[\tau]$. Since $a \in P_\perp$, by (P5)(4), we have $\pi_1(\nabla_{\sigma \times \tau}(a)) \in P_\sigma$ and $\pi_2(\nabla_{\sigma \times \tau}(a)) \in P_\tau$. If σ is a base type, then $\mathbf{r}[\sigma] = P_\sigma$ and $\pi_1(\nabla_{\sigma \times \tau}(a)) \in \mathbf{r}[\sigma]$. Similarly, if τ is a base type, then $\mathbf{r}[\tau] = P_\tau$ and $\pi_2(\nabla_{\sigma \times \tau}(a)) \in \mathbf{r}[\tau]$.

If σ is not a base type, then $\pi_1(\nabla_{\sigma \times \tau}(a)) \in P_\sigma$ is a simple term and we use (S3). The case where $\pi_1(\nabla_{\sigma \times \tau}(a))$ is stubborn is trivial. Otherwise, assume that $\mathbf{Cov}_\sigma(C, \pi_1(\nabla_{\sigma \times \tau}(a)))$ where C is a nontrivial cover. Then, by condition (2) of definition 11.1, $c \preceq \nabla_\sigma(a)$ for every $c \in C$. Since by the induction hypothesis, $\nabla_\sigma(a) \in \mathbf{r}[\sigma]$, by (S2), we have $c \in \mathbf{r}[\sigma]$. By (S3), we have $\pi_1(\nabla_{\sigma \times \tau}(a)) \in \mathbf{r}[\sigma]$. A similar argument applies to $\pi_2(\nabla_{\sigma \times \tau}(a))$.

3. Sum type $\sigma + \tau$. By (P4)(5), since $a \in P_\perp$, we have $\nabla_{\sigma + \tau}(a) \in P_{\sigma + \tau}$. Let D be the following set:

$$\begin{aligned} D = & \{ \mathbf{inl}(p_1) \mid p_1 \in \mathbf{r}[\sigma] \text{ and } \nabla_{\sigma + \tau}(a) \succeq \mathbf{inl}(p_1) \} \cup \\ & \{ \mathbf{inr}(p_2) \mid p_2 \in \mathbf{r}[\tau] \text{ and } \nabla_{\sigma + \tau}(a) \succeq \mathbf{inr}(p_2) \} \cup \\ & \{ \nabla_{\sigma + \tau}(p_3) \mid p_3 \in P_\perp \text{ and } \nabla_{\sigma + \tau}(a) \succeq \nabla_{\sigma + \tau}(p_3) \}. \end{aligned}$$

By the properties of \preceq , it is easy to verify that D is indeed a sieve. We need to prove that $\mathbf{Cov}_{\sigma + \tau}(D, \nabla_{\sigma + \tau}(a))$, since this is equivalent to $\nabla_{\sigma + \tau}(a) \in \mathbf{r}[\sigma + \tau]$. Now, since $q \preceq \nabla_{\sigma + \tau}(a)$ implies that $q = \nabla_{\sigma + \tau}(a_1)$ for some $a_1 \preceq a$, and since $a \in P_\perp$, by (P2) we have $a_1 \in P_\perp$. Thus, it is clear that $D = \{ q \mid q \preceq \nabla_{\sigma + \tau}(a) \}$, which is a principal sieve. However, since $\nabla_{\sigma + \tau}(a) \in P_{\sigma + \tau}$, by property (2) of definition 9.1, $\nabla_{\sigma + \tau}(a) \in P_{\sigma + \tau}$ is covered by the principal sieve D , and thus $\mathbf{Cov}_{\sigma + \tau}(D, \nabla_{\sigma + \tau}(a))$. Therefore, we have $\nabla_{\sigma + \tau}(a) \in \mathbf{r}[\sigma + \tau]$. \square

Finally, we now prove the main realizability lemma for $\lambda^{\rightarrow, \times, +, \perp}$.

Lemma 11.6 *Given a well-behaved scenic site $\langle \mathcal{A}, \mathcal{P}, \mathbf{Cov} \rangle$, if \mathcal{P} is a family satisfying conditions (P1)-(P5), then for every term M of type σ , for every valuation ρ such that $\rho(y) \in \mathbf{r}[\gamma]$ for every $y: \gamma \in FV(M)$, we have $\mathcal{A}[M]\rho \in \mathbf{r}[\sigma]$.*

Proof. We proceed by induction on the structure of M . Some of the cases have already been covered in the proof of lemma 7.6, but we also need to handle the new terms.

If $M = \langle M_1, N_1 \rangle$, where M_1 has type σ and N_1 has type τ , then by the induction hypothesis, $\mathcal{A}[[M_1]]\rho \in \mathbf{r}[\sigma]$ and $\mathcal{A}[[N_1]]\rho \in \mathbf{r}[\tau]$. By lemma 11.5, we have $\mathcal{A}[\langle M_1, N_1 \rangle]\rho \in \mathbf{r}[\sigma \times \tau]$.

If $M = \pi_1(M_1)$ where M_1 has type $\sigma \times \tau$, then by the induction hypothesis, $\mathcal{A}[[M_1]]\rho \in \mathbf{r}[\sigma \times \tau]$. By the definition of $\mathbf{r}[\sigma \times \tau]$, this implies that $\pi_1(\mathcal{A}[[M_1]]\rho) \in \mathbf{r}[\sigma]$, that is, $\mathcal{A}[\pi_1(M_1)]\rho \in \mathbf{r}[\sigma]$, by definition 10.1. Similarly, we get $\mathcal{A}[\pi_2(M_1)]\rho \in \mathbf{r}[\tau]$.

If $M = \text{inl}(M_1)$ where M has type $\sigma + \tau$, then by the induction hypothesis, $\mathcal{A}[[M_1]]\rho \in \mathbf{r}[\sigma]$. By (P4)(3), we have $\text{inl}(\mathcal{A}[[M_1]]\rho) \in P_{\sigma+\tau}$. Consider the cover D of $\text{inl}(\mathcal{A}[[M_1]]\rho)$:

$$\begin{aligned} D = & \{ \text{inl}(p_1) \mid p_1 \in \mathbf{r}[\sigma] \text{ and } \text{inl}(\mathcal{A}[[M_1]]\rho) \succeq \text{inl}(p_1) \} \cup \\ & \{ \text{inr}(p_2) \mid p_2 \in \mathbf{r}[\tau] \text{ and } \text{inl}(\mathcal{A}[[M_1]]\rho) \succeq \text{inr}(p_2) \} \cup \\ & \{ \nabla_{\sigma+\tau}(p_3) \mid p_3 \in P_{\perp} \text{ and } \text{inl}(\mathcal{A}[[M_1]]\rho) \succeq \nabla_{\sigma+\tau}(p_3) \}. \end{aligned}$$

We need to show that $\text{Cov}_{\sigma+\tau}(D, \text{inl}(\mathcal{A}[[M_1]]\rho))$. We claim that

$$D = \{ p \mid \text{inl}(\mathcal{A}[[M_1]]\rho) \succeq p \}.$$

By the properties of \preceq , $p \preceq \text{inl}(\mathcal{A}[[M_1]]\rho)$ implies that $p = \text{inl}(p_1)$ and $p_1 \preceq \mathcal{A}[[M_1]]\rho$. Since $\mathcal{A}[[M_1]]\rho \in \mathbf{r}[\sigma]$, and by (S2), $p_1 \in \mathbf{r}[\sigma]$ whenever $p_1 \preceq \mathcal{A}[[M_1]]\rho$, we do have

$$D = \{ p \mid \text{inl}(\mathcal{A}[[M_1]]\rho) \succeq p \}.$$

However, by property (2) of definition 9.1, since $\text{inl}(\mathcal{A}[[M_1]]\rho) \in P_{\sigma+\tau}$ and D is a principal cover, $\text{Cov}_{\sigma+\tau}(D, \text{inl}(\mathcal{A}[[M_1]]\rho))$ holds. Since by definition 10.1, $\mathcal{A}[\text{inl}(M_1)]\rho = \text{inl}(\mathcal{A}[[M_1]]\rho)$, we have $\mathcal{A}[\text{inl}(M_1)]\rho \in \mathbf{r}[\sigma + \tau]$. The case where $M = \text{inr}(M_1)$ is similar.

If $M = [M_1, N_1]$ is of type $(\sigma + \tau) \rightarrow \delta$, by the induction hypothesis applied to M_1, N_1 , we have $\mathcal{A}[[M_1]]\rho \in \mathbf{r}[\sigma + \tau]$, and $\mathcal{A}[[N_1]]\rho \in \mathbf{r}[\delta]$. Thus, by lemma 11.5, we have $\mathcal{A}[[M_1, N_1]]\rho \in \mathbf{r}[(\sigma + \tau) \rightarrow \delta]$.

If $M = \nabla_{\sigma}(M_1)$, then by the induction hypothesis, $\mathcal{A}[[M_1]]\rho \in \mathbf{r}[\perp] = P_{\perp}$. By lemma 11.5 (4), we have $\nabla_{\sigma}(\mathcal{A}[[M_1]]\rho) \in \mathbf{r}[\sigma]$. Since by definition 10.1, $\mathcal{A}[\nabla_{\sigma}(M_1)]\rho = \nabla_{\sigma}(\mathcal{A}[[M_1]]\rho)$, we have $\mathcal{A}[\nabla_{\sigma}(M_1)]\rho \in \mathbf{r}[\sigma]$. \square

Theorem 7.7 is generalized to the calculus $\lambda^{\rightarrow, \times, +, \perp}$ as follows.

Theorem 11.7 *Given a well-behaved scenic site $\langle \mathcal{A}, \mathcal{P}, \text{Cov} \rangle$, if \mathcal{P} is a family satisfying conditions (P1)-(P5), then for every closed term M of type σ , we have $\mathcal{A}[[M]] \in P_{\sigma}$. (in other words, the realizer $\mathcal{A}[[M]]$ satisfies the unary predicate defined by \mathcal{P} , i.e., every provable type is realizable).*

Proof. Apply lemma 11.6 to the closed term M of type σ and to any arbitrary valuation ρ . \square

12 Applications to the System $\lambda^{\rightarrow, \times, +, \perp}$

This section shows that theorem 11.7 can be used to prove a general theorem about terms of the system $\lambda^{\rightarrow, \times, +, \perp}$. As a corollary, it can be shown that all terms of $\lambda^{\rightarrow, \times, +, \perp}$ are strongly normalizing and confluent.

In order to apply theorem 11.7, we define a notion of cover for the site \mathcal{A} whose underlying pre-applicative structure is the structure \mathcal{LT}_{β} of definition 8.2.

Definition 12.1 An *I-term* is a term of the form either $\lambda x:\sigma. M$, $\langle M, N \rangle$, $\text{inl}(M)$, $\text{inr}(M)$, $[M, N]$, or $\nabla_\sigma(M)$. 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 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 λ -terms.

Definition 12.2 The cover algebra Cov is defined as follows:

(1) If $M \in P_\sigma$ and M is an I-term, then

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

(2) If $M \in P_\sigma$ and M is a (simple and) stubborn term, then

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

(3) If $M \in P_\sigma$ and M is a simple and non-stubborn term, then

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

Recall from definition 9.2 that M is simple iff it has at least two distinct covers. Thus, definition 12.2 implies that a term is simple in the sense of definition 12.1 iff it is simple in the sense of definition 9.2. Similarly a term is stubborn in the sense of definition 12.1 iff it is stubborn in the sense of definition 9.2. Also, definition 12.1 implies that \mathcal{LT}_β is scenic.

Properties (P1-P3) are listed below.

Definition 12.3 Properties (P1)-(P3) are defined as follows:

(P1) $x \in P_\sigma$, $c \in P_\sigma$, for every variable x and constant c of type σ .

(P2) If $M \in P_\sigma$ and $M \rightarrow_\beta N$, then $N \in P_\sigma$.

(P3) If M is simple, then:

(1) If $M \in P_{\sigma \rightarrow \tau}$, $N \in P_\sigma$, $(\lambda x:\sigma. M')N \in P_\tau$ whenever $M \xrightarrow{+}_\beta \lambda x:\sigma. M'$, and $\nabla_{\sigma \rightarrow \tau}(M')N \in P_\tau$ whenever $M \xrightarrow{+}_\beta \nabla_{\sigma \rightarrow \tau}(M')$, then $MN \in P_\tau$.

(2) If $M \in P_{\sigma \times \tau}$, $\pi_1(\langle M', N' \rangle) \in P_\sigma$ and $\pi_2(\langle M', N' \rangle) \in P_\tau$ whenever $M \xrightarrow{+}_\beta \langle M', N' \rangle$, and $\pi_1(\nabla_{\sigma \times \tau}(M')) \in P_\sigma$ and $\pi_2(\nabla_{\sigma \times \tau}(M')) \in P_\tau$ whenever $M \xrightarrow{+}_\beta \nabla_{\sigma \times \tau}(M')$, then $\pi_1(M) \in P_\sigma$ and $\pi_2(M) \in P_\tau$.

A careful reader will notice that conditions (P3) of definition 12.3 are not simply a reformulation of condition (P3) of definition 9.3. This is because according to definition 12.2, a non-stubborn term M is covered by the nontrivial cover $\{N \mid M \xrightarrow{+}_\beta Q \xrightarrow{*}_\beta N\}$, where Q is some I-term, but the conditions of definition 12.3 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 $M \in P_{\sigma \rightarrow \tau}$ is a stubborn term and $N \in P_\sigma$ is any term, then $MN \in P_\tau$ by (P3). 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 $M \in P_{\sigma \rightarrow \tau}$ is a stubborn term and $N \in P_\sigma$ is any term, then MN is a stubborn term in P_τ . We can show in a similar fashion that (P3) implies that if $M \in P_{\sigma \times \tau}$ is a stubborn term, then $\pi_1(M)$ is a stubborn term in P_σ and $\pi_2(M)$ is a stubborn term in P_τ .

Properties (P4-P5) are listed below.

Definition 12.4 Properties (P4) and (P5) are defined as follows:

(P4)

- (1) If $M \in P_\tau$, then $\lambda x: \sigma. M \in P_{\sigma \rightarrow \tau}$.
- (2) If $M \in P_\sigma$ and $N \in P_\tau$, then $\langle M, N \rangle \in P_{\sigma \times \tau}$.
- (3) If $M \in P_\sigma$, then $\text{inl}(M) \in P_{\sigma+\tau}$, and if $M \in P_\tau$, then $\text{inr}(M) \in P_{\sigma+\tau}$.
- (4) If $M \in P_{\sigma \rightarrow \delta}$ and $N \in P_{\tau \rightarrow \delta}$, then $[M, N] \in P_{(\sigma+\tau) \rightarrow \delta}$.
- (5) If $M \in P_\perp$, then $\nabla_\sigma(M) \in P_\sigma$.

(P5)

- (1) If $N \in P_\sigma$ and $M[N/x] \in P_\tau$, then $(\lambda x: \sigma. M)N \in P_\tau$.
- (2) If $M \in P_\sigma$ and $N \in P_\tau$, then $\pi_1(\langle M, N \rangle) \in P_\sigma$ and $\pi_2(\langle M, N \rangle) \in P_\tau$.
- (3) If $P \in P_{\sigma+\tau}$, $M \in P_{\sigma \rightarrow \delta}$, $N \in P_{\tau \rightarrow \delta}$, $MP_1 \in P_\delta$ whenever $P \xrightarrow{*}_\beta \text{inl}(P_1)$, $NP_2 \in P_\delta$ whenever $P \xrightarrow{*}_\beta \text{inr}(P_2)$, and $P_1 \in P_\perp$ whenever $P \xrightarrow{*}_\beta \nabla_{\sigma+\tau}(P_1)$, then $[M, N]P \in P_\delta$.
- (4) If $M_1 \in P_\perp$ and $N \in P_\sigma$, then $\nabla_{\sigma \rightarrow \tau}(M_1)N \in P_\tau$. If $M_1 \in P_\perp$, then $\pi_1(\nabla_{\sigma \times \tau}(M_1)) \in P_\sigma$ and $\pi_2(\nabla_{\sigma \times \tau}(M_1)) \in P_\tau$.

Again, a careful reader will notice that conditions (P5) of definition 12.4 are not simply a reformulation of conditions (P5) of definition 11.4. 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.

It is easy to verify that $[M, N]P \in P_\delta$ is a stubborn term in P_δ , if $P \in P_{\sigma+\tau}$ is stubborn, $M \in P_{\sigma \rightarrow \delta}$, and $N \in P_{\tau \rightarrow \delta}$. Indeed, $[M, N]P \in P_\delta$ can only reduce to an I-term if P does. We now show that the conditions of definition 9.1 and the conditions of definition 11.2 hold.

Lemma 12.5 *Definition 12.2 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 9.1 are easily verified. Let us verify conditions (5) and (6).

- (5) If $\text{Cov}(M) = \text{triv}(M)$, then $\text{Cov}(MN) = \text{triv}(MN)$, 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 12.1, 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 12.1 implies that $M' \in C$.

- (6) If $\mathbf{Cov}(M) = \mathbf{triv}(M)$, then $\mathbf{Cov}(\pi_1(M)) = \mathbf{triv}(\pi_1(M))$, $\mathbf{Cov}(\pi_2(M)) = \mathbf{triv}(\pi_2(M))$, and if $\mathbf{Cov}(C, M)$ and $\mathbf{Cov}(D, \pi_1(M))$ (resp. $\mathbf{Cov}(D, \pi_2(M))$) with C and D nontrivial, then for every $Q \in D$, there is some $M' \in C$ such that $Q \preceq \pi_1(M')$ (resp. $Q \preceq \pi_2(M')$).

The first part says that if M is stubborn, then $\pi_1(M)$ and $\pi_2(M)$ are stubborn, which has already been verified. If the covers C and D are nontrivial, then by definition 12.1, M , $\pi_1(M)$, and $\pi_2(M)$, must be simple and non-stubborn terms. In this case, $Q \in D$ means that

$$\pi_1(M) \xrightarrow{+}_{\beta} P \xrightarrow{*}_{\beta} Q,$$

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

$$\pi_1(M) \xrightarrow{+}_{\beta} \pi_1(M') \xrightarrow{*}_{\beta} P \xrightarrow{*}_{\beta} Q,$$

where M' is an I-term. Since M is simple and non-stubborn, definition 12.1 implies that $M' \in C$. The same argument applies to $\pi_2(M)$.

Let us now verify the conditions of definition 11.2. First, recall that for the structure \mathcal{LT}_{β} , for every valuation ρ (an infinite substitution) $\mathcal{LT}_{\beta}[[M]]\rho = M[\varphi]$, where φ is the substitution defined by the restriction of ρ to $FV(M)$. Also $\mathbf{app}(M, N) = MN$, and recall that A^{σ} is the set of terms of type σ .

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

We have $\mathbf{app}(\mathcal{LT}_{\beta}[[\lambda x: \sigma. M]]\rho, a) = ((\lambda x: \sigma. M)[\varphi])a$, where φ is the substitution defined by the restriction of ρ to $FV(M) - \{x\}$. By definition 12.1, 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 $\mathcal{LT}_{\beta}[[M]]\rho$, we have $\mathcal{LT}_{\beta}[[M]]\rho[x := a] = M[\varphi[x := a]]$, and this part of the proof is complete. The proof for $\nabla_{\sigma \rightarrow \tau}(M)$ is completely analogous.

- (2) If $\text{Cov}_\sigma(C, \pi_1(\mathcal{LT}_\beta[\langle M_1, M_2 \rangle]\rho))$ and C is a nontrivial cover, then $c \preceq \mathcal{LT}_\beta[M_1]\rho$ for every $c \in C$.
 If $\text{Cov}_\tau(C, \pi_2(\mathcal{LT}_\beta[\langle M_1, M_2 \rangle]\rho))$ and C is a nontrivial cover, then $c \preceq \mathcal{LT}_\beta[M_2]\rho$ for every $c \in C$.
 If $\text{Cov}_\sigma(C, \pi_1(\mathcal{LT}_\beta[\nabla_{\sigma \times \tau}(M)]\rho))$ and C is a nontrivial cover, then $c \preceq \mathcal{LT}_\beta[\nabla_\sigma(M)]\rho$ for every $c \in C$.
 If $\text{Cov}_\tau(C, \pi_2(\mathcal{LT}_\beta[\nabla_{\sigma \times \tau}(M)]\rho))$ and C is a nontrivial cover, then $c \preceq \mathcal{LT}_\beta[\nabla_\tau(M)]\rho$ for every $c \in C$.

We have $\mathcal{LT}_\beta[\langle M_1, M_2 \rangle]\rho = \langle M_1, M_2 \rangle[\varphi]$, where φ is the substitution defined by the restriction of ρ to $FV(M_1) \cup FV(M_2)$. By definition 12.1, since C is nontrivial, $c \in C$ means that

$$\pi_1(\langle M_1, M_2 \rangle[\varphi]) \xrightarrow{+}_{\beta} Q \xrightarrow{*}_{\beta} c,$$

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

$$\pi_1(\langle M_1, M_2 \rangle[\varphi]) \longrightarrow_{\beta} M_1[\varphi] \xrightarrow{*}_{\beta} c.$$

Since $\mathcal{LT}_\beta[M_1]\rho = M_1[\varphi]$, this part of the proof is complete. The other cases are entirely analogous.

- (3) If $\text{Cov}(P) = \text{triv}(P)$, then $\text{Cov}(\text{app}(\mathcal{LT}_\beta[[M, N]]\rho, P)) = \text{triv}(\text{app}(\mathcal{LT}_\beta[[M, N]]\rho, P))$, and if $\text{Cov}_{\sigma+\tau}(C, P)$, $\text{Cov}_\delta(D, \text{app}(\mathcal{LT}_\beta[[M, N]]\rho, P))$, and C and D are nontrivial, then for every $d \in D$, either there is some $\text{inl}(P_1) \in C$ such that $d \preceq \text{app}(\mathcal{LT}_\beta[M]\rho, P_1)$, or there is some $\text{inr}(P_2) \in C$ such that $d \preceq \text{app}(\mathcal{LT}_\beta[N]\rho, P_2)$, or there is some $\nabla_{\sigma+\tau}(P_3) \in C$ such that $d \preceq \nabla_\delta(P_3)$.

The first part says that $[M[\varphi], N[\varphi]]P$ is stubborn if P is stubborn, which has already been shown (where φ is the substitution defined by the restriction of ρ to $FV(M) \cup FV(N)$). By definition 12.1, since D is nontrivial, $d \in D$ means that

$$[M[\varphi], N[\varphi]]P \xrightarrow{*}_{\beta} Q \xrightarrow{*}_{\beta} d,$$

where Q is an I-term. This can happen only if either

$$P \xrightarrow{*}_{\beta} \text{inl}(P_1), \text{ and}$$

$$[M[\varphi], N[\varphi]]\text{inl}(P_1) \longrightarrow_{\beta} M[\varphi]P_1 \xrightarrow{*}_{\beta} d,$$

or $P \xrightarrow{*}_{\beta} \text{inr}(P_2)$, and

$$[M[\varphi], N[\varphi]]\text{inr}(P_2) \longrightarrow_{\beta} N[\varphi]P_2 \xrightarrow{*}_{\beta} d,$$

or $P \xrightarrow{*}_{\beta} \nabla_{\sigma+\tau}(P_3)$, and

$$[M[\varphi], N[\varphi]]\nabla_{\sigma+\tau}(P_3) \longrightarrow_{\beta} \nabla_\delta(P_3) \xrightarrow{*}_{\beta} d.$$

In each case, since C is nontrivial, by definition 12.1, we have $\text{inl}(P_1) \in C$, $\text{inl}(P_2) \in C$, and $\nabla_{\sigma+\tau}(P_3) \in C$. \square

Thus, the site $\langle \mathcal{LT}_\beta, \mathcal{P}, \text{Cov} \rangle$, is scenic and well-behaved. Consequently, we can apply theorem 11.7, and get a general theorem for proving properties of terms of the system $\lambda^{\rightarrow, \times, +, \perp}$. In fact, for the structure \mathcal{LT}_β , for a property \mathcal{P} satisfying conditions (P1)-(P5), by (P1) and (P3), every variable x of type σ is stubborn (for every σ). Thus, we can apply lemma 11.6 with the valuation ρ such that $\rho(x) = x$ for every variable x , since by lemma 9.6, $\mathbf{r}[\sigma]$ contains every stubborn term. Consequently, we have the following theorem (compare with theorem A of the introduction).

Theorem 12.6 *If \mathcal{P} is a family of λ -terms satisfying conditions (P1)-(P5), then $P_\sigma = \Lambda_\sigma$ for every type σ (in other words, every term satisfies the unary predicate defined by \mathcal{P}).*

Proof. By lemma 12.5, the site $\langle \mathcal{LT}_\beta, \mathcal{P}, \text{Cov} \rangle$ is scenic and well-behaved. By the discussion just before stating theorem 12.6, the identity valuation ρ such that $\rho(x) = x$ for every variable x , is such that $\rho(x) \in \mathbf{r}[\sigma]$ for every $x:\sigma$. Thus, we can apply lemma 11.6 to any term M of type σ and to ρ , and we have $\mathcal{LT}_\beta[M]\rho \in \mathbf{r}[\sigma]$. However, in the present case, $\mathcal{LT}_\beta[M]\rho = M$. Thus, $M \in \mathbf{r}[\sigma]$, and since $\mathbf{r}[\sigma] \subseteq P_\sigma$, we have $M \in P_\sigma$, as claimed. \square

As a corollary, strong normalization and confluence can be shown, see Gallier [5] for such a treatment.

We now consider the generalization of the previous treatment to the second-order typed λ -calculus $\lambda^{\rightarrow, \forall^2}$

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

In this section, we review quickly the syntax of the second-order typed λ -calculus $\lambda^{\rightarrow, \forall^2}$. 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, the reader should consult Breazu-Tannen and Coquand [2]. For simplicity, we only consider the types \rightarrow and \forall^2 , but the types \times , $+$, and \perp , can also be handled, as in section 2.

Let \mathcal{T} denote the set of second-order types. This set comprises type variables X , type constants k , and compound types $(\sigma \rightarrow \tau)$, 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 $\text{Type}: \text{Const} \rightarrow \mathcal{T}$, such that every constant c is assigned a *closed type* $\text{Type}(c)$ 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 13.1 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) , $(M\tau)$, $\lambda x: \sigma. M$, and $\lambda X. M$, are raw terms.

We let $\text{FV}(M)$ 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 13.2 The judgements of the polymorphic typed λ -calculus $\lambda^{\rightarrow, \forall^2}$ are defined by the following rules.

$$\begin{aligned} & \Gamma \triangleright x: \sigma, \quad \text{when } x: \sigma \in \Gamma, \\ & \Gamma \triangleright c: \text{Type}(c), \quad \text{when } c \text{ is a constant,} \\ & \frac{\Gamma, x: \sigma \triangleright M: \tau}{\Gamma \triangleright (\lambda x: \sigma. M): (\sigma \rightarrow \tau)} \quad (\text{abstraction}) \\ & \frac{\Gamma \triangleright M: (\sigma \rightarrow \tau) \quad \Gamma \triangleright N: \sigma}{\Gamma \triangleright (MN): \tau} \quad (\text{application}) \\ & \frac{\Gamma, X: \star \triangleright M: \sigma}{\Gamma \triangleright (\lambda X. M): \forall X. \sigma} \quad (\forall\text{-intro}) \end{aligned}$$

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

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

The reason why we do not officially consider that a context contains type variables, is that in the rule $(\forall\text{-elim})$, 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.

Definition 13.3 The reduction rules of the system $\lambda^{\rightarrow, \forall^2}$ are listed below:

$$\begin{aligned} & (\lambda x: \sigma. M)N \longrightarrow M[N/x], \\ & (\lambda X. M)\tau \longrightarrow M[\tau/X]. \end{aligned}$$

The reduction relation defined by the rules of definition 13.3 is denoted as \longrightarrow_β . From now on, when we refer to a λ -term, we mean a λ -term that type-checks. We let $\Lambda_{\langle\sigma, \Gamma\rangle}$ denote the set of judgements of the form $\Gamma \triangleright M: \sigma$.

14 Pre-Applicative Structures for $\lambda_{\rightarrow, \forall^2}$

In this section, the definition of a pre-applicative structure (given in section 3) is generalized to $\lambda_{\rightarrow, \forall^2}$. For simplicity, only pre-applicative β -structures are defined. Pre-applicative $\beta\eta$ -structures and extensional pre-applicative β -structures are defined in an appendix (see section 21). The types \times , $+$, and \perp , can easily be handled as in section 8, but for simplicity, we only deal with the types \rightarrow and \forall^2 . 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 14.2) in order to “curry” the map $\mathbf{tapp}^\Phi: A^{\forall(\Phi)} \times T \rightarrow \coprod (A^{\Phi(s)})_{s \in T}$.

Definition 14.1 An algebra of (polymorphic) types is a tuple

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

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

We hope that readers will forgive us for denoting an algebra of types $\langle T, \rightarrow, [T \Rightarrow T], \forall \rangle$ with the same symbol T . Intuitively, given a valuation $\theta: \mathcal{V} \rightarrow T$, a type $\sigma \in \mathcal{T}$ 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 $\bigcup \{ \langle a, i \rangle \mid a \in A_i \}_{i \in I}$. 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 $a \preceq_i b$.

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

Definition 14.2 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 $\prod (A^{\Phi(t)})_{t \in T}$, which is also described explicitly as the set of functions in $(\prod (A^{\Phi(s)})_{s \in T})^T$ defined as follows:

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

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

$$\mathcal{A} = \langle A, \preceq, \text{fun}, \text{abst}, \text{tfun}, \text{tabst} \rangle,$$

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 $\Phi \in [T \Rightarrow T]$.

It is assumed that fun , abst , tfun , and tabst , 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 $\varphi \in \prod_{\Phi}(A^s)_{s \in T}$.

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.

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 $\varphi \in \prod_{\Phi}(A^s)_{s \in T}$.

We say that a pre-applicative β -structure is an *applicative β -structure* iff in conditions (1)-(2), \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 17 how the term model can be viewed as a pre-applicative β -structure (see definition 17.5).

When \mathcal{A} is an applicative β -structure, then, in definition 14.3, conditions (1)-(2) amounts to

- (1) $\text{fun}^{s,t} \circ \text{abst}^{s,t} = \text{id}$ on the domain of definition of abst ;

(2) $\mathbf{tfun}^\Phi \circ \mathbf{tabst}^\Phi = \mathbf{id}$ on the domain of definition of \mathbf{tabst} .

In this case, \mathbf{abst} is injective and \mathbf{fun} is surjective on the domain of definition of \mathbf{abst} (and left inverse to \mathbf{abst}), \mathbf{tabst} is injective and \mathbf{tfun} is surjective on the domain of definition of \mathbf{tabst} (and left inverse to \mathbf{tabst}).

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

15 \mathcal{P} -Cover Algebras and \mathcal{P} -Sheaves for $\lambda^{\rightarrow, \forall^2}$

In this section, we basically repeat the definitions for covers and sheaves given in section 9, 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 section 9, we define all the necessary concepts in terms of preorders, referring the interested reader to MacLane and Moerdijk [18] 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 15.1 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 \mathbf{Cov} of binary relations \mathbf{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 sections 9, 10, and 11).

Definition 15.2 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).
- (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)-(2) 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.

Conditions (3) and (4) have been omitted, since they are 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 15.3 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)-(P3).

Definition 15.4 Given an algebra of types T , let $\langle \mathcal{A}, \mathcal{P}, \mathbf{Cov} \rangle$ 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 $\mathbf{Cov}_{s \rightarrow t}(C, M)$, $\mathcal{P}(N, s)$, and $\mathcal{P}(M'N, t)$ whenever $M' \in C$, then $\mathcal{P}(MN, t)$.
- (P3b) If $\mathbf{Cov}_{\vee(\Phi)}(C, M)$, $s \in T$, 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 15.4. The sheaf property is defined as in section 9, except that a more general notion of pre-applicative structure is involved.

Definition 15.5 Given an algebra of types T , let $\langle \mathcal{A}, \mathcal{P}, \mathbf{Cov} \rangle$ 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 $\mathbf{Cov}_s(C, M)$ 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 15.5 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 15.5 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 (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 15.6 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

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

Since \mathcal{P} itself is a \mathcal{P} -sheaf, the set $\text{Sheaf}(\mathcal{A}, \mathcal{P})$ is nonempty. The fact that definition 15.5 is indeed a sheaf condition is shown exactly as in section 4 (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 $a \in P_s$).

16 \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 16.4).

Definition 16.1 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 $\sigma \in \mathcal{T}$ is interpreted as an element $\llbracket \sigma \rrbracket \theta$ of T as follows:

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

In the above definition, $\Lambda t \in T. \llbracket \sigma \rrbracket \theta[X := t]$ denotes the function Φ from T to T such that $\Phi(t) = \llbracket \sigma \rrbracket \theta[X := t]$ 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 $\llbracket \sigma \rrbracket \theta$ is well-defined for every valuation θ . The following lemmas will be needed later.

Lemma 16.2 *For every type $\sigma \in \mathcal{T}$, 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 $\llbracket \sigma \rrbracket \theta_1 = \llbracket \sigma \rrbracket \theta_2$.*

Proof. A straightforward induction on σ . \square

Lemma 16.3 *Given a type interpretation T , for all $\sigma, \tau \in \mathcal{T}$, for every type valuation θ , we have*

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

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

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

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

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

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

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

Then, since

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

we have

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

\square

The next definition can be viewed as a semantic version of Girard's "candidats de réductibilité" (see Girard [7], Gallier [4]).

Definition 16.4 *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 $\eta: \mathcal{V} \rightarrow \bigcup \mathbf{Sheaf}(\mathcal{A}, \mathcal{P})$ is a function called a *candidate assignment*, such that:*

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

Given $\mu = \langle \theta, \eta \rangle$, for any $s \in T$ and any $S \in \mathbf{Sheaf}(\mathcal{A}, \mathcal{P})_s$, for some s -component $S = S_s$ of some \mathcal{P} -sheaf $\mathcal{S} = (S_s)_{s \in T} \in \mathbf{Sheaf}(\mathcal{A}, \mathcal{P})$, 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 16.5 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 $(\mathbf{r}[\sigma]\mu)_{\sigma \in \mathcal{T}}$, where for every $\sigma \in \mathcal{T}$, $\mathbf{r}[\sigma]\mu$ is the set of *realizers* of σ , is defined as follows:

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

The following lemmas will be needed later.

Lemma 16.6 For every type $\sigma \in \mathcal{T}$, 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 16.2). \square

Lemma 16.7 Given a type interpretation T and a \mathcal{P} -site $\langle \mathcal{A}, \mathcal{P}, \text{Cov} \rangle$, for all $\sigma, \tau \in \mathcal{T}$, for every sheaf-valuation $\mu = \langle \theta, \eta \rangle$, we have

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

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 16.5, we have

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

By lemma 16.3, we have

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

and by the induction hypothesis, we have

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

However, by definition,

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

and so, we have

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

\square

The following lemma shows that the notion of a \mathcal{P} -sheaf is an inductive invariant. In Gallier [4], this is the lemma we call ‘‘ Girard’s trick’’.⁴

⁴Of course, this is unfair. Girard has many tricks!

Lemma 16.8 *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 \mathbf{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 \mathbf{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 show 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 15.2, $\text{Cov}_{[\sigma \rightarrow \tau]\theta}(C, M)$ 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 $N \in P_{[\tau]\theta}$, 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 15.2, 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 15.2, $\mathbf{Cov}_{[\forall X. \sigma]\theta}(C, M)$ 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 15.2, 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.

17 Interpreting $\lambda^{\rightarrow, \forall^2}$ in Pre-Applicative Structures

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

Definition 17.1 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

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

Note that if ρ satisfies a context Γ , this implies that $A^{\llbracket \sigma \rrbracket \theta} \neq \emptyset$ for every $x: \sigma \in \Gamma$. Also, conditions (1)-(2) of definition 14.3 imply that the following conditions hold:

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

We are now ready to interpret λ -terms.

Definition 17.2 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:

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

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

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

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

where φ is the function defined such that,

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

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

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

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

where φ is the function defined such that,

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

We are assuming that the domains of \mathbf{abst} and \mathbf{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 17.3 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 16.2). \square

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

Definition 17.4 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 17.5 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 operation \rightarrow is defined as follows:

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

We let $A^{\mathbf{error}} = \emptyset$, and $A^{\langle \sigma, \Gamma \rangle}$ be the set of all provable typing judgements of the form $\Gamma \triangleright M : \sigma$. We denote $A^{\langle \sigma, \Gamma \rangle}$ 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, $\forall(\Phi) = \langle \forall X. \sigma, \Gamma \rangle$.⁵

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 operation \rightarrow , either $[\sigma]\theta = \mathbf{error}$, 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 $\epsilon : \mathcal{X} \rightarrow \bigcup (A^s)_{s \in T}$. As noted just after definition 17.1, 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 .⁶

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).

⁵The choice of X is irrelevant as long as X does not occur in Γ , since X is bound in $\forall X. \sigma$.

⁶ $A_\Delta^\sigma = \emptyset$ when there is no provable judgement $\Delta \triangleright M : \sigma$ for any M .

Given a context Γ , according to definition 17.1, 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 $[\epsilon]: \mathcal{X} \rightarrow \text{Terms}$, 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 $x \in \text{dom}(\Gamma)$, and $\varphi(X) = [\theta](X)$ for every $X \in \bigcup_{\sigma \in \Gamma} FTV(\sigma)$. Also, $\rho \models \Gamma$ is just the condition $\Delta \Vdash \Gamma[\varphi]$ of definition 17.4, where Δ is the context uniquely determined by θ .

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 **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 **tfun**($\Gamma \triangleright M: \forall X. \sigma$) as the function $[\Gamma \triangleright M: \forall X. \sigma]$ from T to $\coprod (A_{\Gamma}^{\sigma})_{\sigma \in T}$, 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** $^{\Phi}$ 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

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

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 $\coprod (A_{\Delta}^{\sigma})_{\sigma \in T}$, defined such that,

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

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 $\prod_{\Phi(A_{\Delta}^s)}_{s \in T}$. 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 14.3 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 17.5.

18 The Realizability Theorem for $\lambda^{\rightarrow, \forall^2}$

In this section, we prove the realizability lemma (lemma 18.6) for $\lambda^{\rightarrow, \forall^2}$, and its main corollary, theorem 18.7. 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).

Definition 18.1 We say that a site $\langle \mathcal{A}, \mathcal{P}, \mathbf{Cov} \rangle$ is *well-behaved* iff the following conditions hold:

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

In view of definition 17.2, definition 18.1 implies the following condition.

- Definition 18.2**
- (1) For any $a \in A^{\llbracket \sigma \rrbracket \tau}$, if $\mathbf{Cov}_{\llbracket \tau \rrbracket \theta}(C, \mathbf{app}(\mathcal{A}[\llbracket \Gamma \triangleright \lambda x: \sigma. M: (\sigma \rightarrow \tau) \rrbracket \rho], a))$ and C is a nontrivial cover, then $c \preceq \mathcal{A}[\llbracket \Gamma, x: \sigma \triangleright M: \tau \rrbracket \rho[x: = a]]$ for every $c \in C$.
 - (2) For any $s \in T$, if $\mathbf{Cov}_{\llbracket \sigma \rrbracket \theta[X: = s]}(C, \mathbf{tapp}(\mathcal{A}[\llbracket \Gamma \triangleright \lambda X. M: \forall X. \sigma \rrbracket \rho], s))$ and C is a nontrivial cover, then $c \preceq \mathcal{A}[\llbracket \Gamma, X: \star \triangleright M: \sigma \rrbracket \rho[X: = s]]$ 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 18.3 Given a well-behaved site $\langle \mathcal{A}, \mathcal{P}, \mathbf{Cov} \rangle$, 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 $\mathbf{abst}(\varphi)$ exists, then $\mathbf{abst}(\varphi) \in P_{s \rightarrow t}$.
- (P4b) For every $s \in T$, if $\varphi(s) \in P_{\Phi(s)}$, where $\varphi \in \prod_{\Phi}(A^s)_{s \in T}$ and $\mathbf{tabst}(\varphi)$ exists, then $\mathbf{tabst}(\varphi) \in P_{\forall(\Phi)}$.
- (P5a) If $a \in P_s$ and $\varphi(a) \in P_t$, where $\varphi \in [A^s \Rightarrow A^t]$ and $\mathbf{abst}(\varphi)$ exists, then $\mathbf{app}(\mathbf{abst}(\varphi), a) \in P_t$.
- (P5b) If $s \in T$ and $\varphi(s) \in P_{\Phi(s)}$, where $\varphi \in \prod_{\Phi}(A^s)_{s \in T}$ and $\mathbf{tabst}(\varphi)$ exists, then $\mathbf{tapp}(\mathbf{tabst}(\varphi), s) \in P_{\Phi(s)}$.

In view of definition 17.2, definition 18.3 implies the following conditions.

Definition 18.4

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

Lemma 18.5 *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 16.8, 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 18.5, $\mathcal{A}[\Gamma, x: \sigma \triangleright M: \tau]\rho \in \mathbf{r}[\tau]\mu$. 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 18.5, $\mathcal{A}[\Gamma, x: \sigma \triangleright M: \tau]\rho[x: = a] \in \mathbf{r}[\tau]\mu$. 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 16.8, 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 18.2, $c \leq \mathcal{A}[\Gamma, x: \sigma \triangleright M: \tau]\rho[x: = a]$ 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 16.8, 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 18.5, $\mathcal{A}[\Gamma, X: \star \triangleright M: \sigma]\rho \in \mathbf{r}[\sigma]\mu[X: = \langle s, S \rangle]$ 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 18.5, $\mathcal{A}[\Gamma, X: \star \triangleright M: \sigma]\rho[X: = s] \in \mathbf{r}[\sigma]\mu[X: = \langle s, S \rangle]$. 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 18.2, $c \preceq \mathcal{A}[\Gamma, X: \star \triangleright M: \sigma]\rho[X := s]$ 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 18.6 *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 17.2.

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 18.5 (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 $s = \llbracket \tau \rrbracket \theta$ and $S = \mathbf{r}[\tau]\mu$, we get

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

However, by lemma 16.7, we have

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

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 17.2, we have $\mathcal{A} \llbracket \Gamma \triangleright (M_1 \tau) : \sigma[\tau/X] \rrbracket \rho \in \mathbf{r} \llbracket \sigma[\tau/X] \rrbracket \mu$.

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 $S \in \mathbf{Sheaf}(\mathcal{A}, \mathcal{P})_s$. Since $X \notin \text{dom}(\Gamma)$, by lemma 16.6, we have $\mathbf{r} \llbracket \delta \rrbracket \mu = \mathbf{r} \llbracket \delta \rrbracket \mu[X := \langle s, S \rangle]$ 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

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

Since the above holds for every $s \in T$ and every $S \in \mathbf{Sheaf}(\mathcal{A}, \mathcal{P})_s$, by lemma 18.5 (2), we have $\mathcal{A} \llbracket \Gamma \triangleright \lambda X. M_1 : \forall X. \sigma \rrbracket \rho \in \mathbf{r} \llbracket \forall X. \sigma \rrbracket \mu$. \square

If M is a closed term of type σ , lemma 17.3 implies that $\mathcal{A} \llbracket \triangleright M : \sigma \rrbracket \rho$ is independent of ρ , and thus we denote it as $\mathcal{A} \llbracket M : \sigma \rrbracket$. We obtain the following important theorem for $\lambda^{\rightarrow, \forall^2}$.

Theorem 18.7 *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 $\mathcal{A} \llbracket M : \sigma \rrbracket \in P_{\llbracket \sigma \rrbracket \theta}$. (in other words, the realizer $\mathcal{A} \llbracket M : \sigma \rrbracket$ satisfies the unary predicate defined by \mathcal{P} , i.e., every provable type is realizable).*

Proof. Apply lemma 18.6 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

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

This section shows that theorem 18.7 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 18.7, we define a notion of cover for the site \mathcal{A} whose underlying pre-applicative structure is the structure \mathcal{LT}_β of definition 17.5.

Definition 19.1 An *I-term* is a term of the form either $\lambda x : \sigma. M$ or $\lambda X. M$. 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 , or a type application $M\tau$. 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 19.2 The cover algebra Cov is defined as follows:

- (1) If $\Gamma \triangleright M : \sigma \in P_{\langle \sigma, \Gamma \rangle}$ 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 15.3 that M is simple iff it has at least two distinct covers. Thus, definition 19.2 implies that a term is simple in the sense of definition 19.1 iff it is simple in the sense of definition 15.3. Similarly a term is stubborn in the sense of definition 19.1 iff it is stubborn in the sense of definition 15.3. Also, definition 19.1 implies that \mathcal{LT}_β is scenic.

Properties (P1-P3) are listed below.

Definition 19.3 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 19.3 are not simply a reformulation of conditions (P3) of definition 15.4. This is because according to definition 19.2, $\Gamma \triangleright M: \sigma$, 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 19.3 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 19.4 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 $\tau \in \mathcal{T}$ 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 19.4 are not simply a reformulation of conditions (P5) of definition 18.4. 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 15.2 and the conditions of definition 18.2 hold.

Lemma 19.5 *Definition 19.2 defines a cover algebra, and the site $\langle \mathcal{L}\mathcal{T}_\beta, \mathcal{P}, \text{Cov} \rangle$ is scenic and well-behaved.*

Proof. The verification is straightforward. As an illustration, let us verify the conditions of definition 18.2. First, recall that for the structure $\mathcal{L}\mathcal{T}_\beta$, 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

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

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

- (1) For any $a \in A^{\llbracket \sigma \rrbracket \tau}$, if $\text{Cov}_{\llbracket \tau \rrbracket \theta}(C, \text{app}(\mathcal{A}[\Gamma \triangleright \lambda x : \sigma. M : (\sigma \rightarrow \tau)]\rho, a))$ 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 19.1, 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 $\mathcal{L}\mathcal{T}_\beta[\Gamma, x : \sigma \triangleright M : \tau]\rho$, we have $\mathcal{L}\mathcal{T}_\beta[\Gamma, x : \sigma \triangleright M : \tau]\rho[x := a] = \Delta \triangleright M[\varphi[x := a]] : \tau[\varphi]$, and this part of the proof is complete.

- (2) For any $s \in \mathcal{T}$, if $\text{Cov}_{\llbracket \sigma \rrbracket \theta[X := s]}(C, \text{tapp}(\mathcal{A}[\Gamma \triangleright \lambda X. M : \forall X. \sigma]\rho, s))$ 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 19.1, 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 $\mathcal{L}\mathcal{T}_\beta[\Gamma, X : \star \triangleright M : \sigma]\rho$, we have

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

and the proof is complete. \square

Thus, the site $\langle \mathcal{L}\mathcal{T}_\beta, \mathcal{P}, \mathbf{Cov} \rangle$, is scenic and well-behaved. Consequently, we can apply theorem 18.7, and get a general new theorem for proving properties of terms of the system $\lambda^{\rightarrow, \forall^2}$. In fact, for the structure $\mathcal{L}\mathcal{T}_\beta$, 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 18.6 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 16.8, $\mathbf{r}[\delta]\mu$ contains every stubborn term, for every $x: \delta \in \Gamma$. Consequently, we have the following new theorem.

Theorem 19.6 *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 19.5, the site $\langle \mathcal{L}\mathcal{T}_\beta, \mathcal{P}, \mathbf{Cov} \rangle$ is scenic and well-behaved. By the discussion just before stating theorem 19.6, 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 18.6 to any judgement $\Gamma \triangleright M: \sigma$ and to μ and ρ just defined, and we have

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

However, in the present case, $\mathcal{L}\mathcal{T}_\beta[\Gamma \triangleright M: \sigma]\rho = \Gamma \triangleright M: \sigma$. 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 19.7 *The reduction relation $\xrightarrow{*}_\beta$ of the system $\lambda^{\rightarrow, \forall^2}$ 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 19.6, 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 tree⁷ 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 depth⁸ 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 $MN \rightarrow_\beta P$. Since M is simple, MN itself is not a redex, and so $P = M_1N_1$ where either $N = N_1$ and $M \rightarrow_\beta M_1$, or $M = M_1$ and $N \rightarrow_\beta N_1$.

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.

⁷the tree of reduction sequences from M

⁸the length of a longest path in the tree, counting the number of edges

(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 $M\tau \rightarrow_{\beta} P$. Since M is simple, $M\tau$ itself is not a redex, and so $P = M_1\tau$ where $M \rightarrow_{\beta} M_1$.

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.

(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 [5].

20 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, including a new theorem for proving properties of terms in $\lambda^{\rightarrow, \forall^2}$ (theorem 19.6). As corollaries, we obtain alternate proofs of 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. It should also be noted that our pre-applicative structures are models of the reduction relation, and not of the convertibility relation. This seems inevitable, since we are interested in proving properties of the reduction relation, but this seems to have been missed until now. We also managed to formulate conditions on the property \mathcal{P} to be proved, independently of the conditions on the candidates. Strong normalization and confluence happen to satisfy these conditions, but more progress in this direction would be interesting.

Extending the results of this paper to pre-applicative $\beta\eta$ -structures and to typed λ -calculi with η -like reductions should pose no problems for the types \rightarrow , \times , and \forall^2 . However, in view of results of Dougherty [3], there may be some difficulties in dealing with the sum type, since confluence fails (with the traditional orientation of η -like rules).

As we mentioned in the introduction, Hyland and Ong [11] 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 somewhat 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 [19], and a recent paper by Michel Parigot [21], the conditions on sets of realizers are sheaf conditions.⁹ One simply needs to change slightly the definition of Cov . 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 [19], 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. It seems that their approach consists in modifying the definition of reducibility itself. However, only strong normalization is considered, and it seems that they squeezed some of the conditions from one place to another in the proof. Their presentation may be more attractive to the community at large, which is not a negligible point.

We believe that nonextensional structures modelling reduction 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

⁹We need to examine more closely these approaches to determine whether they fit into our framework.

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. Another challenging question is to figure out whether it is possible to adapt the framework of this paper to deal with other calculi, for example, the pure λ -calculus, or calculi for various systems of linear logic.

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, Scott Weinstein, and two anonymous referees, for some very helpful comments.

21 Appendix: Extensional and $\beta\eta$ Pre-Applicative Structures

We begin with extensional pre-applicative structures for $\lambda^{\rightarrow, \times, +}$. 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 21.1 A pre-applicative β -structure \mathcal{A} is *extensional* iff \mathbf{fun} , Π , and $\langle \mathbf{cinl}, \mathbf{cinr} \rangle$, are isotone, and the following conditions hold:

- (1) $\mathit{ran}(\mathbf{fun}) \subseteq \mathit{dom}(\mathbf{abst})$;
- (2) $\mathit{ran}(\Pi) \subseteq \mathit{dom}(\langle -, - \rangle)$;
- (3) $\mathit{ran}(\langle \mathbf{cinl}^{\sigma, \tau, \delta}, \mathbf{cinr}^{\sigma, \tau, \delta} \rangle) \subseteq \mathit{dom}([_, _] \circ (\mathbf{abst}^{\sigma, \delta} \times \mathbf{abst}^{\tau, \delta}))$.

When \mathcal{A} is an applicative β -structure, conditions (1)-(3) hold, and the functions \mathbf{fun} , Π , and $\langle \mathbf{cinl}, \mathbf{cinr} \rangle$, 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), $\mathbf{abst}(\mathbf{fun}(f))$ is defined for any $f \in A^{\sigma \rightarrow \tau}$. Observe that by condition (1) of definition 8.1, we have $\mathbf{fun}(f) \preceq \mathbf{fun}(\mathbf{abst}(\mathbf{fun}(f)))$, and since \mathbf{fun} is isotone, this implies that

- (1) $\mathbf{abst}(\mathbf{fun}(f)) \succeq f$, for all $f \in A^{\sigma \rightarrow \tau}$.

Similarly, we can prove that

- (2) $\langle \pi_1(a), \pi_2(a) \rangle \succeq a$, for all $a \in A^{\sigma \times \tau}$; and
- (3) $[\mathbf{abst}(\mathbf{cinl}(h)), \mathbf{abst}(\mathbf{cinr}(h))] \succeq h$, for all $h \in A^{(\sigma + \tau) \rightarrow \delta}$.

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

- (1) $\mathit{ran}(\mathbf{fun}) \subseteq \mathit{dom}(\mathbf{abst})$, and $\mathbf{abst}(\mathbf{fun}(f)) \succeq f$, for all $f \in A^{\sigma \rightarrow \tau}$;
- (2) $\mathit{ran}(\Pi) \subseteq \mathit{dom}(\langle -, - \rangle)$, and $\langle \pi_1(a), \pi_2(a) \rangle \succeq a$, for all $a \in A^{\sigma \times \tau}$; and
- (3) $\mathit{ran}(\langle \mathbf{cinl}^{\sigma, \tau, \delta}, \mathbf{cinr}^{\sigma, \tau, \delta} \rangle) \subseteq \mathit{dom}([_, _] \circ (\mathbf{abst}^{\sigma, \delta} \times \mathbf{abst}^{\tau, \delta}))$, and

$$[\mathbf{abst}(\mathbf{cinl}(h)), \mathbf{abst}(\mathbf{cinr}(h))] \succeq h, \text{ for all } h \in A^{(\sigma+\tau) \rightarrow \delta}.$$

When \mathcal{A} is an applicative β -structure and in conditions (1)-(3), \succeq is replaced by $=$, we say that we have an *applicative $\beta\eta$ -structure*.

From the remark before definition 21.2, an extensional pre-applicative β -structure is a $\beta\eta$ -structure. When \mathcal{A} is an applicative $\beta\eta$ -structure, conditions (1)-(3) of definition 21.2 amount to:

- (1) $\mathbf{abst}^{\sigma,\tau} \circ \mathbf{fun}^{\sigma,\tau} = \mathbf{id}$;
- (2) $\langle -, - \rangle^{\sigma,\tau} \circ \Pi^{\sigma,\tau} = \mathbf{id}$; and
- (3) $([-, -] \circ (\mathbf{abst}^{\sigma,\delta} \times \mathbf{abst}^{\tau,\delta})) \circ \langle \mathbf{cinl}^{\sigma,\tau,\delta}, \mathbf{cinr}^{\sigma,\tau,\delta} \rangle = \mathbf{id}$.

This implies that \mathbf{fun} , Π , and $\langle \mathbf{cinl}, \mathbf{cinr} \rangle$, are injective. Thus, an applicative $\beta\eta$ -structure is extensional. In this case, (together with conditions (1)-(3) of definition 8.1 in the case of an applicative β -structure), we have $\mathit{dom}(\mathbf{abst}) = \mathbf{fun}(A^{\sigma \rightarrow \tau})$, \mathbf{fun} is a bijection between $A^{\sigma \rightarrow \tau}$ and a subset of $[A^\sigma \Rightarrow A^\tau]$ (with inverse \mathbf{abst}), Π is a bijection between $A^{\sigma \times \tau}$ and a subset of $A^\sigma \times A^\tau$ (with inverse $\langle -, - \rangle$), and $\langle \mathbf{cinl}^{\sigma,\tau,\delta}, \mathbf{cinr}^{\sigma,\tau,\delta} \rangle$ is a bijection between $A^{(\sigma+\tau) \rightarrow \delta}$ and a subset of $[A^\sigma \Rightarrow A^\delta] \times [A^\tau \Rightarrow A^\delta]$ (with inverse $[-, -] \circ (\mathbf{abst}^{\sigma,\delta} \times \mathbf{abst}^{\tau,\delta})$).

Extensional pre-applicative structures and $\beta\eta$ -structures for $\lambda^{\rightarrow, \times, +, \perp}$ are defined just as in definition 21.1 and definition 21.2, and the same remarks apply. However, these remarks only apply for types different from \perp .

We now define extensional pre-applicative structures for $\lambda^{\rightarrow, \forall^2}$.

Definition 21.3 A pre-applicative β -structure \mathcal{A} is *extensional* iff \mathbf{fun} and \mathbf{tfun} are isotone, and the following conditions hold:

- (1) $\mathit{ran}(\mathbf{fun}) \subseteq \mathit{dom}(\mathbf{abst})$;
- (2) $\mathit{ran}(\mathbf{tfun}) \subseteq \mathit{dom}(\mathbf{tabst})$.

When \mathcal{A} is an applicative β -structure, conditions (1)-(2) hold, and the functions \mathbf{fun} and \mathbf{tfun} 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), $\mathbf{abst}(\mathbf{fun}(f))$ is defined for any $f \in A^{s \rightarrow t}$. Observe that by condition (1) of definition 14.3, we have $\mathbf{fun}(f) \preceq \mathbf{fun}(\mathbf{abst}(\mathbf{fun}(f)))$, and since \mathbf{fun} is isotone, this implies that

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

Similarly, we can prove that

- (2) $\mathbf{tabst}(\mathbf{tfun}(f)) \succeq f$, for all $f \in A^{\forall(\Phi)}$.

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

- (1) $\text{ran}(\mathbf{fun}) \subseteq \text{dom}(\mathbf{abst})$, and $\mathbf{abst}(\mathbf{fun}(f)) \succeq f$, for all $f \in A^{s \rightarrow t}$;
- (2) $\text{ran}(\mathbf{tfun}) \subseteq \text{dom}(\mathbf{tabst})$, and $\mathbf{tabst}(\mathbf{tfun}(f)) \succeq f$, for all $f \in A^{\vee(\Phi)}$.

When \mathcal{A} is an applicative β -structure and in conditions (1)-(2), \succeq is replaced by $=$, we say that we have an *applicative $\beta\eta$ -structure*.

The term model can easily be made into a pre-applicative $\beta\eta$ -structure (by adapting definition 17.5). From the remark before definition 21.4, an extensional pre-applicative β -structure is a $\beta\eta$ -structure. When \mathcal{A} is an applicative $\beta\eta$ -structure, conditions (1)-(2) of definition 21.4 amount to:

- (1) $\mathbf{abst}^{s,t} \circ \mathbf{fun}^{s,t} = \text{id}$;
- (2) $\mathbf{tabst}^{\Phi} \circ \mathbf{tfun}^{\Phi} = \text{id}$.

This implies that \mathbf{fun} and \mathbf{tfun} , are injective. Thus, an applicative $\beta\eta$ -structure is extensional. In this case, (together with conditions (1)-(4) of definition 14.3 in the case of an applicative β -structure), we have $\text{dom}(\mathbf{abst}) = \text{fun}(A^{s \rightarrow t})$, \mathbf{fun} is a bijection between $A^{s \rightarrow t}$ and a subset of $[A^s \Rightarrow A^t]$ (with inverse \mathbf{abst}), $\text{dom}(\mathbf{tabst}) = \mathbf{tfun}(A^{\vee(\Phi)})$, and \mathbf{tfun} is a bijection between $A^{\vee(\Phi)}$ and a subset of $\prod_{\Phi}(A^s)_{s \in T}$ (with inverse \mathbf{tabst}).

References

- [1] M. J. Beeson. *Foundations of Constructive Mathematics*. Springer-Verlag, Berlin, 1985.
- [2] V. Breazu-Tannen and T. Coquand. Extensional models for polymorphism. *Theoretical Computer Science*, 59:85–114, 1988.
- [3] D.J. Dougherty. Some lambda calculi with categorical sums and products. In C. Kirchner, editor, *Rewriting Techniques and Applications*, LNCS, Vol. 690, pages 137–151. Springer-Verlag, 1993.
- [4] 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.
- [5] Jean H. Gallier. On the correspondence between proofs and λ -terms. In P. DeGroot, editor, *The Curry-Howard Isomorphism*, Cahiers du Centre de Logique, No. 8, pages 55–138. Université Catholique de Louvain, 1995.
- [6] Jean-Yves Girard. Une extension de l’interprétation de Gödel à l’analyse, et son application à l’élimination des coupures dans l’analyse et la théorie des types. In J.E. Fenstad, editor, *Proc. 2nd Scand. Log. Symp.*, pages 63–92. North-Holland, 1971.
- [7] 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.
- [8] Jean-Yves Girard. The system F of variable types, fifteen years later. *Theoretical Computer Science*, 45(2):159–192, 1986.

- [9] R. J. Grayson. Forcing in intuitionistic systems without power set. *J. Symbolic Logic*, 48(3):670–682, 1983.
- [10] C. A. Gunter. *Semantics of Programming Languages*. Foundations of Computing. MIT Press, 1992.
- [11] 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.
- [12] B. Jacobs, I. Margaria, and M. Zacchi. Filter models with polymorphic types. *Theoretical Computer Science*, 95(1):143–158, 1992.
- [13] S. C. Kleene. On the interpretation of intuitionistic number theory. *J. Symbolic Logic*, 10:109–124, 1945.
- [14] S. C. Kleene. *Introduction to Metamathematics*. North-Holland, seventh edition, 1952.
- [15] G. Koletsos. Church-Rosser theorem for typed functional systems. *J. Symbolic Logic*, 50(3):782–790, 1985.
- [16] G. Kreisel. Interpretation of analysis by means of functionals of finite type. In A. Heyting, editor, *Constructivity in Mathematics*, pages 101–128. North-Holland, 1959.
- [17] J.L. Krivine. *Lambda-Calcul, types et modèles*. Etudes et recherches en informatique. Masson, 1990.
- [18] S. MacLane and I. Moerdijk. *Sheaves in Geometry and Logic*. Springer Verlag, New York, 1992.
- [19] 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.
- [20] J. C. Mitchell. A type-inference approach to reduction properties and semantics of polymorphic expressions. In *ACM Conference on LISP and Functional Programming*, pages 308–319. ACM, 1986. Reprinted in *Logical Foundations of Functional Programming*, G. Huet, Ed., Addison Wesley, 1990, 195-212.
- [21] M. Parigot. Strong normalization for second-order classical natural deduction. In *Eighth Annual IEEE Symposium on Logic In Computer Science*, pages 39–46. IEEE, 1993.
- [22] G.D. Plotkin. A semantics for static type inference. *Theoretical Computer Science*, 1993. To appear.
- [23] R. Statman. Logical Relations and the Typed λ -Calculus. *Information and Control*, 65(2/3):85–97, 1985.
- [24] W.W. Tait. Intensional interpretation of functionals of finite type I. *J. Symbolic Logic*, 32:198–212, 1967.

- [25] W.W. Tait. A realizability interpretation of the theory of species. In R. Parikh, editor, *Logic Colloquium*, volume 453 of *Lecture Notes in Math.*, pages 240–251. Springer Verlag, 1975.
- [26] A. S. Troelstra. *Metamathematical Investigation of intuitionistic arithmetic and analysis*. Lecture Notes in Mathematics No 344. Springer-Verlag, 1973.
- [27] A.S. Troelstra and D. van Dalen. *Constructivism in Mathematics: An Introduction, Vol. I and II*, volume 123 of *Studies in Logic*. North-Holland, 1988.