August 1993

Realizability, Covers, and Sheaves I. Application to the Simply-Typed Lambda-Calculus

Jean H. Gallier
University of Pennsylvania, jean@cis.upenn.edu

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

Recommended Citation


This paper is posted at ScholarlyCommons. http://repository.upenn.edu/cis_reports/259
For more information, please contact repository@pobox.upenn.edu.
Realizability, Covers, and Sheaves I. Application to the Simply-Typed Lambda-Calculus

Abstract
We present a general method for proving properties of typed λ-terms. This method is obtained by introducing a semantic notion of realizability which uses the notion of a cover algebra (as in abstract sheaf theory, a cover algebra being a Grothendieck topology in the case of a preorder). For this, we introduce a new class of semantic structures equipped with preorders, called pre-applicative structures. These structures need not be extensional. In this framework, a general realizability theorem can be shown. Kleene's recursive realizability and a variant of Kreisel's modified realizability both fit into this framework. Applying this theorem to the special case of the term model, yields a general theorem for proving properties of typed λ-terms, in particular, strong normalization and confluence. This approach clarifies the reducibility method by showing that the closure conditions on candidates of reducibility can be viewed as sheaf conditions. Part I of this paper applies the above approach to the simply-typed λ-calculus (with types →, ×, +, and ⊥). Part II of this paper deals with the second-order (polymorphic) λ-calculus (with types → and ∀).

Comments

This technical report is available at ScholarlyCommons: http://repository.upenn.edu/cis_reports/259
Realizability, Covers, and Sheaves
I. Application to the Simply-typed λ-Calculus

MS-CIS-93-46
LOGIC & COMPUTATION 64

Jean Gallier

University of Pennsylvania
School of Engineering and Applied Science
Computer and Information Science Department
Philadelphia, PA 19104-6389

April 1993
Realizability, Covers, and Sheaves.
I. Applications to the Simply-Typed $\lambda$-Calculus

Preliminary Version

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

August 12, 1993

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

*This research was partially supported by ONR Grant NOOO14-88-K-0593.
1 Introduction

Kleene, Kreisel, and others ([8], [11], [18]), 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 [9], Troelstra [18], Troelstra and van Dalen [19], and Beeson [1]). Tait [16], introduced reducibility (or computability), as a technique for proving strong normalization for the simply-typed \( \lambda \)-calculus. Girard [4], introduced the method of the candidates of reducibility a technique for proving strong normalization for the second-order typed \( \lambda \)-calculus (and \( F_\omega \)). Statman [15] and Mitchell [14], 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 [3, 4], Tait [16, 17], Krivine, [12]), 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. In order to do so, 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 (a Grothendieck topology, in the case of a preorder). 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 \((r\sigma)_{\sigma \in T}\) of sets of realizers associated with the types, is a sheaf. Actually, we consider abstract properties \( P \) of these sets of realizers. The main theorem is the following: provided that the abstract property \( P \) satisfies some fairly simple conditions (P1)-(P5), if a type \( \sigma \) is provable and \( M \) is a proof for \( \sigma \), then the meaning \( A[M]_P \) of \( M \) is a realizer of \( \sigma \) that satisfies the property \( P \). As a corollary, considering the term model for the simply-typed lambda calculus (with types \( \rightarrow \), \( \times \), \( + \), and \( \bot \)), 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 a recent paper, Hyland and Ong [7] show how strong normalization proofs can be obtained from the construction of a modified realizability topos. Very roughly, they show how a suitable
The quotient of the strongly normalizing untyped terms can be made into a categorical (modified realizability) interpretation of system F. There is no doubt that Hyland and Ong's approach and our approach are related, but the technical details are very different, and we are unable to make a precise comparison at this point. 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 sketch our approach for the simply-typed \( \lambda \)-Calculus \( \lambda^{-} \).

Recall that the types and the terms of \( \lambda^{-} \) are given by the following grammar:

\[
\begin{align*}
\sigma & \rightarrow b \mid (\sigma \rightarrow \sigma) \\
M & \rightarrow c \mid x \mid (MM) \mid (\lambda x: \sigma. M).
\end{align*}
\]

The type-checking rules are as usual (see section 2), and we let \( \Lambda_{\sigma} \) denote the set of \( \lambda \)-terms of type \( \sigma \).

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 [4]). However, Koletsos [10] realized the following even more crucial fact:

**Crucial Fact:** \( MN \twoheadrightarrow_{\beta} Q \), where \( Q \) is an I-term, only if \( M \) itself reduces to an I-term.

Let \( \mathcal{P} = (P_{\sigma})_{\sigma \in \tau} \) be a family of properties of the simply-typed \( \lambda \)-terms (that type-check). For example, \( P_{\sigma}(M) \) holds iff \( M \) is strongly normalizing (SN), or \( P_{\sigma}(M) \) holds iff confluence holds from \( M \). In Gallier [2], 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 \( \sigma \).

(P2) If \( M \in P_{\sigma} \) and \( M \twoheadrightarrow_{\beta} N \), then \( N \in P_{\sigma} \).

(P3) If \( M \) is simple, \( M \in P_{\sigma_{\rightarrow}}, N \in P_{\tau}, \) and \( (\lambda x: \sigma. M')N \in P_{\tau} \) whenever \( M \twoheadrightarrow_{\tau}^{+} \lambda x: \sigma. M' \), then \( MN \in P_{\tau} \).

(P4) If \( M \in P_{\tau} \), then \( \lambda x: \sigma. M \in P_{\sigma_{\rightarrow}} \).

(P5) If \( N \in P_{\sigma} \) and \( M[N/x] \in P_{\tau} \), then 
\[
(\lambda x: \sigma. M)N \in P_{\tau}.
\]

Then, \( P_{\sigma} \) holds for all terms of type \( \sigma \), i.e. \( P_{\sigma} = \Lambda_{\sigma} \), for every \( \sigma \in \tau \).

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 \( \lambda^{-} \).

3
The proof of Theorem A uses a version of reducibility in which the types are interpreted as follows:

\[ [\sigma] = P_\sigma, \quad \sigma \text{ a base type}, \]

\[ [\sigma \to \tau] = \{ M \mid M \in P_{\sigma \to \tau}, \text{ and for all } N, \]

if \( N \in [\sigma] \text{ then } MN \in [\tau] \}.

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

A family \( S = (S_\sigma)_{\sigma \in \mathcal{T}} \) of nonempty sets of terms is a \( \mathcal{P} \)-candidate if 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 \( M \vdash_\beta \lambda x: \gamma. M' \), 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 \( M \vdash_\beta Q \) 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 [13]). We can think of the set

\[ \{ Q \mid M \vdash_\beta Q, \text{ } Q \text{ an } I \text{-term} \} \]

as a cover of \( M \). 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 \( S = (S_\sigma)_{\sigma \in \mathcal{T}} \) as a functor

\[ S: \mathcal{L}T^{\text{op}} \to \text{Sets}, \]

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

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

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 [13]).

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 (\( \vdash_\beta \)).
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 $\lambda$-calculus. For simplicity, we are restricting our attention to arrow types. Let $T$ be the set of simple types built up from some base types using the constructor $\to$. Given a signature $\Sigma$ of function symbols, where each symbol in $\Sigma$ is assigned some type in $T$, an applicative structure $A$ is defined as a triple

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

where

- $(A^\sigma)_{\sigma \in T}$ is a family of nonempty sets called carriers,
- $(\text{app}^{\sigma, \tau})_{\sigma, \tau \in T}$ is a family of application operators, where each $\text{app}^{\sigma, \tau}$ is a total function $\text{app}^{\sigma, \tau}: A^\sigma \to A^\tau$;
- and $\text{Const}$ is a function assigning a member of $A^\sigma$ to every symbol in $\Sigma$ of type $\sigma$.

The meaning of simply-typed $\lambda$-terms is usually defined using the notion of an environment, or valuation. A valuation is a function $p: X \to \bigcup (A^\sigma)_{\sigma \in T}$, where $X$ 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 $\lambda$-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 $\Gamma \vdash M: \sigma$, where $\Gamma$, called a context, is a set of variable declarations of the form $x_1: \sigma_1, \ldots, x_n: \sigma_n$, where the $x_i$ are pairwise distinct and the $\sigma_i$ are types, $M$ is a simply-typed $\lambda$-term, and $\sigma$ is a type. There is a standard proof system that allows to type-check terms. A term $M$ type-checks with type $\sigma$ in the context $\Gamma$ (where $\Gamma$ contains an assignment of types to all the variables in $M$) iff the judgement $\Gamma \vdash M: \sigma$ is derivable in this proof system. Given a context $\Gamma$, we say that a valuation $\rho$ satisfies $\Gamma$ iff $\rho(x) \in A^\sigma$ for every $x: \sigma \in \Gamma$ (in other words, $\rho$ respects the typing of the variables declared in $\Gamma$). Then given a context $\Gamma$ and a valuation $\rho$ satisfying $\Gamma$, the meaning $[\Gamma \vdash M: \sigma]_{\rho}$ of a judgement $\Gamma \vdash M: \sigma$ is defined by induction on the derivation of $\Gamma \vdash M: \sigma$, according to the following clauses:

- $[\Gamma \vdash x: \sigma]_{\rho} = \rho(x)$, if $x$ is a variable;
- $[\Gamma \vdash c: \sigma]_{\rho} = \text{Const}(c)$, if $c$ is a constant;
- $[\Gamma \vdash M N: \tau]_{\rho} = \text{app}^{\sigma, \tau}([\Gamma \vdash M: (\sigma \to \tau)]_{\rho}, [\Gamma \vdash N: \sigma]_{\rho})$,
- $[\Gamma \vdash \lambda x: \sigma. M : (\sigma \to \tau)]_{\rho} = f$, where $f$ is the unique element of $A^{\sigma \to \tau}$ such that $\text{app}^{\sigma, \tau}(f, a) = [\Gamma, x: \sigma \vdash M : \tau]_{\rho}[x := a]$, for every $a \in A^\sigma$.

Note that in order for the element $f \in A^{\sigma \to \tau}$ 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 $f, g \in A^{\sigma \to \tau}$, if $\text{app}(f, a) = \text{app}(g, a)$ 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 $A^\sigma$ contains enough elements so that there is an element $f \in A^{\sigma \to \tau}$ such that $\text{app}^{\sigma, \tau}(f, a) = [\Gamma, x: \sigma \vdash M : \tau]_{\rho}[x := a]$, for every $a \in A^\sigma$.

Note that each operator $\text{app}^{\sigma, \tau}: A^{\sigma \to \tau} \times A^\sigma \to A^\tau$ induces a function $\text{fun}^{\sigma, \tau}: A^{\sigma \to \tau} \to [A^\sigma \to A^\tau]$, where $[A^\sigma \to A^\tau]$ denotes the set of functions from $A^\sigma$ to $A^\tau$, defined such that $\text{fun}^{\sigma, \tau}(f)(a) = \text{app}^{\sigma, \tau}(f, a)$. 

5
for all \( f \in A^\sigma \to \tau \), and all \( a \in A^\sigma \). Then, extensionality is equivalent to the fact that each \( \text{fun}^\sigma \tau \) is injective. Note that \( \text{fun}^\sigma \tau : A^\sigma \to [A^\sigma \Rightarrow A^\tau] \) is the “curried” version of \( \text{app}^\sigma \tau : A^\sigma \times A^\sigma \to A^\tau \), and it exists because the category of sets is Cartesian-closed.

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

\[
\text{fun}^\sigma \tau (\text{abst}^\sigma \tau (\varphi)) = \varphi,
\]

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

Thus, by defining a more general kind of applicative structure using the operators \( \text{fun}^\sigma \tau \) and \( \text{abst}^\sigma \tau \), we can still give meanings to \( \lambda \)-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 Gunter [6]. 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 \( A^\sigma \) is equipped with a preorder \( \preceq^\sigma \), and rather than considering the equality

\[
\text{fun}^\sigma \tau (\text{abst}^\sigma \tau (\varphi)) = \varphi,
\]

we can consider inequalities

\[
\text{fun}^\sigma \tau (\text{abst}^\sigma \tau (\varphi)) \succeq \varphi.
\]

This way, we can deal with intentional (nonapplicative) structures that model reduction rather than conversion.

This paper is organized as follows. The syntax of the simply-typed \( \lambda \)-calculus \( \lambda \to \times \to + \) is reviewed in section 2. Pre-applicative structures 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 in section 4. The notion of \( \mathcal{P} \)-realizability is defined in section 5, for the arrow type. In section 6, it is shown how to interpret terms in \( \lambda \to \) in pre-applicative structures. The realizability theorem for the typed \( \lambda \)-calculus \( \lambda \to \) is shown in section 7. The notion of \( \mathcal{P} \)-realizability is extended to \( \lambda \to \times \to + \) in section 8. In section 9, it is shown how to interpret terms in \( \lambda \to \times \to + \) in pre-applicative structures. The realizability theorem for the typed \( \lambda \)-calculus \( \lambda \to \times \to + \) is shown in section 10. The syntax of the simply-typed \( \lambda \)-calculus \( \lambda \to \times \to + , \bot \) is reviewed in section 11. Pre-applicative structures for the typed \( \lambda \)-calculus \( \lambda \to \times \to + , \bot \) are defined in section 12. The notion of \( \mathcal{P} \)-realizability is extended to \( \lambda \to \times \to + , \bot \) in section 13. In section 14, it is shown how to interpret terms in \( \lambda \to \times \to + , \bot \) in pre-applicative structures. The realizability theorem for the typed \( \lambda \)-calculus \( \lambda \to \times \to + , \bot \) is shown in section 15. Section 16 contains an application of the main theorem of section 15 to prove a general theorem about terms of the system \( \lambda \to \times \to + , \bot \).
2 Syntax of the Typed λ-Calculus $\lambda^{\to, \times, +}$

Let $T$ denote the set of (simple) types, including base types $b$, and compound types $(\sigma \to \tau)$, $(\sigma \times \tau)$, 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 $\mathcal{X} = (X_\sigma)_{\sigma \in T}$ of variables, where each $X_\sigma$ is a countably infinite set of variables of type $\sigma$, 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 $\text{inl}(x: \sigma) \Rightarrow M \mid \text{inr}(y: \tau) \Rightarrow N$, it will be more convenient and simpler to use a slightly more general construct $[M, N]$, where $M$ is of type $\sigma \to \delta$ and $N$ is of type $\tau \to \delta$, 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 $\text{inl}(x: \sigma) \Rightarrow M \mid \text{inr}(y: \tau) \Rightarrow N$, where $P$ is of type $\sigma + \tau$, as $[\lambda x: \sigma. M, \lambda y: \tau. N]P$. The type-checking rules of the system are summarized in the following definition.

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

\[
x: \sigma, \text{ when } x \in X_\sigma,
\]

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

\[
\frac{M: \tau}{(\lambda x: \sigma. M): \sigma \to \tau} \quad \text{(abstraction)}
\]

where $x \in X_\sigma$;

\[
\frac{M: \sigma \to \tau \quad N: \sigma}{(MN): \tau} \quad \text{(application)}
\]

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

\[
\frac{M: \sigma \times \tau}{\pi_1(M): \sigma} \quad \text{(projection)}
\]

\[
\frac{M: \sigma \times \tau}{\pi_2(M): \tau} \quad \text{(projection)}
\]

\[
\frac{M: \sigma}{\text{inl}(M): \sigma + \tau} \quad \text{(injection)}
\]

\[
\frac{M: \tau}{\text{inr}(M): \sigma + \tau} \quad \text{(injection)}
\]

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

The standard elimination rule for $+$ is:

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

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 case \(P\) of \(\text{inl}(x: \sigma) \Rightarrow M | \text{inr}(y: \tau) \Rightarrow N\). For this, we design more atomic reduction rules for \([M, N]\). These rules do not incorporate the \(\beta\)-reduction step implicit in the traditional reduction rules.

**Definition 2.2** The reduction rules of the system \(\lambda^{\rightarrow,\times,\rightarrow}\) are listed below:

\[
\begin{align*}
(\lambda x: \sigma. M)N & \rightarrow M[N/x], \\
\pi_1((M, N)) & \rightarrow M, \\
\pi_2((M, N)) & \rightarrow N, \\
[M, N]\text{inl}(P) & \rightarrow MP, \\
[M, N]\text{inr}(P) & \rightarrow NP.
\end{align*}
\]

The traditional rules for the case construct are

\[
\begin{align*}
\text{case inl}(P)\text{ of } \text{inl}(x: \sigma) & \Rightarrow M | \text{inr}(y: \tau) \Rightarrow N \rightarrow M[P/x], \\
\text{case inr}(P)\text{ of } \text{inl}(x: \sigma) & \Rightarrow M | \text{inr}(y: \tau) \Rightarrow N \rightarrow N[P/y].
\end{align*}
\]

The above reduction rules can be simulated by the \([-, -]\)-rules of definition 2.2 and \(\beta\)-reduction as follows:

\[
\begin{align*}
[\lambda x: \sigma. M, \ \lambda y: \tau. N]\text{inl}(P) & \rightarrow (\lambda x: \sigma. M)P \rightarrow_\beta M[P/x], \\
[\lambda x: \sigma. M, \ \lambda y: \tau. N]\text{inr}(P) & \rightarrow (\lambda y: \tau. N)P \rightarrow_\beta N[P/y].
\end{align*}
\]

The reduction relation defined by the rules of definition 2.2 is denoted as \(\rightarrow_\beta\) (even though there are reductions other than \(\beta\)-reduction). From now on, when we refer to a \(\lambda\)-term, we mean a \(\lambda\)-term that type-checks. We let \(\Lambda_\sigma\) denote the set of \(\lambda\)-terms of type \(\sigma\).

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. \(\preceq^\sigma\) and \(\preceq^\tau\), under the pointwise preorder induced by \(\preceq^\tau\) defined such that, \(f \preceq g\) iff \(f(a) \preceq^\tau g(a)\) for all \(a \in A^\sigma\).

### 3 Pre-Applicative Structures

In this section, some new semantic structures called pre-applicative structures are defined. There are various kinds of pre-applicative structures: pre-applicative \(\beta\)-structures, pre-applicative \(\beta\eta\)-structures, extensional pre-applicative \(\beta\)-structures, and the corresponding so-called applicative versions. We also show that the term model can be viewed as a pre-applicative \(\beta\)-structures, and that the HRO models of Kreisel and Troelstra [11, 18] can be viewed as an applicative \(\beta\)-structure.

**Definition 3.1** A pre-applicative \(\beta\)-structure is a structure

\[A = \langle A, \preceq, \text{fun}, \text{abst}, \text{II}, \langle-, -, \rangle, \text{inl}, \text{inr}, \langle-, -, \rangle\rangle,\]

where

\[A = (A^\sigma)_{\sigma \in T}\] is a family of (nonempty) sets called carriers;
(≤σ)σ∈τ is a family of preorders, each ≤σ on Aσ;
abstσ,τ: [Aσ → Aτ] → Aσ→τ, a family of partial operators;
funσ,τ: Aσ→τ → [Aσ → Aτ], a family of (total) operators;
(-,-)σ,τ: Aσ × Aτ → Aσ×τ, a family of partial pairing operators;
Πσ,τ: Aσ×τ → Aσ × Aτ, a family of (total) projection operators;
[-,-]σ,τ,δ: Aσ→δ × Aτ→δ → Aσ+τ→δ, a family of partial copairing operators;
inlσ,τ: Aσ → Aσ+τ, a family of (total) operators;
inrσ,τ: Aτ → Aσ+τ, a family of (total) operators.

We define cinl: A(σ+τ)→δ → [Aσ ⇒ Aδ] and cinr: A(σ+τ)→δ → [Aτ ⇒ Aδ] as follows: For every h ∈ A(σ+τ)→δ, cinl(h)(a) = fun(h)(inl(a)),
for every a ∈ Aσ, and cinr(h)(b) = fun(h)(inr(b)),
for every b ∈ Aτ.

It is assumed that fun, abst, Π, (−, −), inl, inr, and [−, −], are monotonic. Furthermore, the following conditions are satisfied
(1) funσ,τ(abstσ,τ(φ)) ≥ φ, whenever abstσ,τ(φ) is defined for φ ∈ [Aσ ⇒ Aτ];
(2) Πσ,τ((a, b)) ≥ (a, b), for all a ∈ Aσ, b ∈ Aτ, whenever (a, b) is defined;
(3) cinl([f, g]) ≥ fun(f), and cinr([f, g]) ≥ fun(g), whenever [f, g] is defined, for f ∈ Aσ→δ and g ∈ Aτ→δ.

The operators fun induce (total) operators
appσ,τ: Aσ→τ × Aσ → Aτ, such that, for every f ∈ Aσ→τ and every a ∈ Aσ,
appσ,τ(f, a) = funσ,τ(f)(a).

Then, condition (1) can be written as
(1') appσ,τ(abstσ,τ(φ), a) ≥ φ(a), for all a ∈ Aσ, for φ ∈ [Aσ ⇒ Aτ], whenever abstσ,τ(φ) is defined, and condition (3) can be rewritten as
(3') cinl([f, g])(a) ≥ app(f, a), for all a ∈ Aσ, and cinr([f, g])(b) ≥ app(g, b), for all b ∈ Aτ, whenever [f, g] is defined, for f ∈ Aσ→δ and g ∈ Aτ→δ.

Finally, N ≤ inl(M₁) implies that N = inl(N₁) for some N₁ ≤ M₁, and N ≤ inr(M₁) implies that N = inr(N₁) for some N₁ ≤ M₁.

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

Intuitively, A is a set of realizers. We will omit superscripts whenever possible.

The projection operators Π induce projections π₁σ,τ: Aσ×τ → Aσ and π₂σ,τ: Aσ×τ → Aτ, such that for every a ∈ Aσ×τ, if Πσ,τ(a) = (a₁, a₂), then π₁σ,τ(a) = a₁ and π₂σ,τ(a) = a₂.
When \( A \) is an applicative \( \beta \)-structure, then, in definition 3.1, conditions (1)-(3) amounts to

1. \( \text{fun}^{\sigma, \tau} \circ \text{abst}^{\sigma, \tau} = \text{id} \) on the domain of definition of \( \text{abst} \);
2. \( \Pi^{\sigma, \tau} \circ (-, -)^{\tau, \sigma} = \text{id} \) on the domain of definition of \((-,-); \) and
3. \( \langle \text{cinl}, \text{cinr} \rangle \circ [-,-] = \text{fun}^{\sigma, \delta} \times \text{fun}^{\tau, \delta} \) on the domain of definition of \([-,-].

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

\[ \langle \text{cinl}, \text{cinr} \rangle \circ (\text{abst}^{\sigma, \delta} \times \text{abst}^{\tau, \delta}) = \text{id} \] on the domain of definition of \([-,-] \circ (\text{abst}^{\sigma, \delta} \times \text{abst}^{\tau, \delta}).

In this case, \( \text{abst} \) is injective and \( \text{fun} \) is surjective on the domain of definition of \( \text{abst} \) (and left inverse to \( \text{abst} \)), \((-,-) \) is injective and \( \Pi \) is surjective on the domain of definition of \((-,-) \) (and left inverse to \((-,-) \)), \([-,-] \circ (\text{abst}^{\sigma, \delta} \times \text{abst}^{\tau, \delta}) \) is injective on its domain of definition, and \( \langle \text{cinl}, \text{cinr} \rangle \) is surjective on this domain (and left inverse to \([-,-] \circ (\text{abst}^{\sigma, \delta} \times \text{abst}^{\tau, \delta}).

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

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

**Definition 3.2** A pre-applicative \( \beta \)-structure \( A \) is extensional iff \( \text{fun} \), \( \Pi \), and \( \langle \text{cinl}, \text{cinr} \rangle \), are isotone, and the following conditions hold:

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

When \( A \) is an applicative \( \beta \)-structure, conditions (1)-(3) hold, and the functions \( \text{fun}, \Pi, \) and \( \langle \text{cinl}, \text{cinr} \rangle \), are injective, we say that we have an extensional applicative \( \beta \)-structure.

When \( A \) is an extensional pre-applicative \( \beta \)-structure, in view of condition (1), \( \text{abst} (\text{fun}(f)) \) is defined for any \( f \in A^{\sigma^{\rightarrow \tau}}. \) Observe that by condition (1) of definition 3.1, we have \( \text{fun}(f) \preceq \text{fun}(\text{abst}(\text{fun}(f))) \), and since \( \text{fun} \) is isotone, this implies that

1. \( \text{abst}(\text{fun}(f)) \succeq f \), 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 \( a \in A^{\sigma 	imes \tau}; \) and
3. \( [\text{abst}(\text{cinl}(h)), \text{abst}(\text{cinr}(h))] \succeq h \), for all \( h \in A^{(\sigma+\tau)^{\rightarrow \delta}}. \)

We will call the above inequalities the \( \eta \)-like rules.

In many cases, a pre-applicative \( \beta \)-structure that satisfies the \( \eta \)-like rules is not extensional. This motivates the next definition.
Definition 3.3 A pre-applicative $\beta$-structure $A$ is a $\beta\eta$-structure if the following conditions hold:

1. $\text{ran}(\text{fun}) \subseteq \text{dom}(\text{abst})$, and $\text{abst}(\text{fun}(f)) \supseteq f$, for all $f \in A^{\sigma \rightarrow \tau}$;
2. $\text{ran}(\Pi) \subseteq \text{dom}((-,-))$, and $(\pi_1(a), \pi_2(a)) \supseteq a$, for all $a \in A^{\sigma \times \tau}$; and
3. $\text{ran}((\text{cinl}^{\sigma,\tau,\delta}, \text{cinr}^{\sigma,\tau,\delta})) \subseteq \text{dom}([-,-] \circ (\text{abst}^{\sigma,\delta} \times \text{abst}^{\tau,\delta}))$, and $\text{abst}(\text{cinl}(h)), \text{abst}(\text{cinr}(h)) \supseteq h$, for all $h \in A^{(\sigma+\tau)\rightarrow \delta}$.

When $A$ is an applicative $\beta$-structure and in conditions (1)-(3), $\supseteq$ is replaced by $=$, we say that we have an applicative $\beta\eta$-structure.

From the remark before definition 3.3, an extensional pre-applicative $\beta$-structure is an $\beta\eta$-structure. When $A$ is an applicative $\beta\eta$-structure, conditions (1)-(3) of definition 3.3 amount to:

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

This implies that $\text{fun}$, $\Pi$, and $(\text{cinl}, \text{cinr})$, are injective. Thus, an applicative $\beta\eta$-structure is extensional. In this case, (together with conditions (1)-(3) of definition 3.1 in the case of an applicative $\beta$-structure), we have $\text{dom}(\text{abst}) = \text{fun}(A^{\sigma \rightarrow \tau})$, $\text{fun}$ is a bijection between $A^{\sigma \rightarrow \tau}$ and a subset of $A^{\sigma \rightarrow \tau}$ (with inverse $\text{abst}$), $\Pi$ is a bijection between $A^{\sigma \times \tau}$ and a subset of $A^{\sigma \times \tau}$ (with inverse $(-,-)$), and $(\text{cinl}^{\sigma,\tau,\delta}, \text{cinr}^{\sigma,\tau,\delta})$ is a bijection between $A^{(\sigma+\tau)\rightarrow \delta}$ and a subset of $A^{\sigma \rightarrow \tau}$ (with inverse $[-,-] \circ (\text{abst}^{\sigma,\delta} \times \text{abst}^{\tau,\delta})$).

Let us give an (important) example of a pre-applicative $\beta$-structure.

Definition 3.4 Let $A^r = \Lambda_\sigma$ be the set of all typed $\lambda$-terms of type $\sigma$. We let $\text{app}$, $\pi_1$, $\pi_2$, $(-,-)$, $\text{inl}$, $\text{inr}$, $([-,-]$ be the obvious constructs (for example, $\text{app}(M,N) = MN$). Define $N \preceq M$ iff $M \rightarrow_\beta N$. Finally, we need to define $\text{abst}$. For every (type-preserving) substitution $\varphi$, for every term $M : \tau$ and for every variable $x$ of type $\sigma$, consider the function $\varphi[x : \sigma \triangleright M : \tau]$ from $A^\sigma$ to $A^\tau$, 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

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

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

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

Indeed, $(\lambda x : \sigma. M)[\varphi]$ is $\alpha$-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 $y$,

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

The other conditions of definition 3.1 are easily verified.

In order to get a $\beta\eta$-version of $\Lambda T_\beta$, we add the $\eta$-reduction rule to our typed $\lambda$-calculus, and we extend the definition of $\text{abst}$ as follows.
Definition 3.5 The definition of abst given in definition 3.4 is extended as follows. For every term \( M \) of type \( \sigma \rightarrow \tau \), we have the function \( \text{fun}(M) \) from \( A^\sigma \) to \( A^\tau \) defined such that \( \text{fun}(M)(N) = MN \) for every \( N \) of type \( \sigma \), and we let \( \text{abst}(\text{fun}(M)) = \lambda x: \sigma. (Mx) \) for any variable \( x \notin FV(M) \). The resulting applicative structure is denoted as \( LT_{\beta\eta} \).

Since \( \lambda x: \sigma. (Mx) \rightarrow_\eta M \), we have \( \text{abst}(\text{fun}(M)) \supseteq M \). We also have \( \text{fun}(\text{abst}(\text{fun}(M))) \supseteq \text{fun}(M) \), since \( \text{abst}(\text{fun}(M)) = \lambda x: \sigma. (Mx) \), \( \text{fun}(\lambda x: \sigma. (Mx))(N) = (\lambda x: \sigma. (Mx))N \) for every \( N \) of type \( \sigma \), and since \( (\lambda x: \sigma. (Mx))N \rightarrow_\beta MN \), since \( x \notin FV(M) \).

Another interesting example is provided by an adaptation of the so-called HRO-models (hereditarily recursive operations), due to Kreisel and Troelstra [11, 18]. 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 \( \varphi_e \). Recall that such a numbering induces a partial operation \( \cdot: N \times N \rightarrow N \) (where \( N \) denotes the set of natural numbers) defined as follows: \( m \cdot n = \varphi_{m}(n) \), whenever it is defined. A partial recursive function \( \varphi_e \) is recursive iff \( \varphi_e(n) \) is defined for all \( n \in N \). We also assume that we have a given pairing function \( p: N \times N \rightarrow N \), with projection functions \( j_1: N \times N \rightarrow N \) and \( j_2: N \times N \rightarrow N \), such that \( p(j_1(m), j_2(m)) = m \) for all \( m \in N \), \( j_1(p(m, n)) = m \), and \( j_2(p(m, n)) = n \), for all \( m, n \in N \).

Definition 3.6 We define an applicative structure as follows. Each \( A^\sigma \) is a set of pairs of the form \( \langle n, \sigma \rangle \), where \( n \in N \), and we denote the subset \( \{ n | \langle n, \sigma \rangle \in A^\sigma \} \) of \( N \) as \( \text{dom}(A^\sigma) \).

Let \( A^\sigma = \{ \langle n, \sigma \rangle | n \in N \} \), for every base type \( \sigma \),

\[
A^{\sigma \rightarrow \tau} = \{ \langle e, \sigma \rightarrow \tau \rangle | \varphi_e \text{ is total on } \text{dom}(A^\sigma) \},
\]

\[
A^{\sigma \times \tau} = \{ \langle n, \sigma \times \tau \rangle | \langle j_1(n), \sigma \rangle \in A^\sigma \text{ and } \langle j_2(n), \tau \rangle \in A^\tau \},
\]

and

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

The preorder on each \( A^\sigma \) is the identity relation.

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

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

where \( e \) is some designated index for \( \psi \) (some index \( e \) such that \( \varphi_{e} = \psi \)).

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

\(^1\)Note that \( \text{fun}(M) \) can be defined as \( \text{id}[x: \sigma. (Mx): \tau] \), where \( x \notin FV(M) \).
For every $m \in \mathbb{N}$, for every $e \in \mathbb{N}$, index of a total recursive function of $m + 1$ arguments, for every finite sequence $\rho = (\rho_1, \ldots, \rho_m)$ of natural numbers, let $e[\rho]$ denote the function in $[A^\sigma \Rightarrow A^\tau]$ defined such that

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

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

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

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

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

$$\text{fun}(\text{abst}(\varphi))(\langle n, \sigma \rangle) = \langle \varphi_{s(e, m, \rho_1, \ldots, \rho_m)}(n), \tau \rangle = \langle \varphi_e(\rho_1, \ldots, \rho_m, n), \tau \rangle,$$

by the s-m-n-theorem, and thus, $\text{fun}(\text{abst}(e[\rho])) = e[\rho]$. The other conditions of definition 3.1 are easily verified. These structures are not extensional.

## 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 [13]). 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 [13] 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 [5], 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 $\beta$-structures (and thus, we will omit the prefix $\beta$).

**Definition 4.1** Given a pre-applicative structure $A$, 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 $A$ is a family $\text{Cov}$ of binary relations $\text{Cov}_\sigma$ on $2^{A^\sigma} \times A^\sigma$, relating subsets
of $A^\sigma$ called covers, to elements of $A^\sigma$. Equivalently, $\text{Cov}$ can be defined as a family of functions $\text{Cov}_\sigma : A^\sigma \to 2^{2^A}$ assigning to every element $M \in A^\sigma$ a set $\text{Cov}(M)$ of subsets of $A^\sigma$ (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 $\text{triv}(M)$ 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 $P \subseteq A \times \mathcal{T}$, such that $P(M, \sigma)$ implies $M \in A^\sigma$, and for every $\sigma \in \mathcal{T}$, there is some $M \in A^\sigma$ s.t. $P(M, \sigma)$. Equivalently, $P$ can be viewed as a family $P = (P_\sigma)_{\sigma \in \mathcal{T}}$, where each $P_\sigma$ is a nonempty subset of $A^\sigma$. The intuition behind $P$ 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 $P$ be a family $P = (P_\sigma)_{\sigma \in \mathcal{T}}$, where each $P_\sigma$ is a nonempty subset of $A^\sigma$. A $P$-cover algebra (or $P$-Grothendieck topology) on $\mathcal{A}$ is a family $\text{Cov}$ of binary relations $\text{Cov}_\sigma$ on $2^{A^\sigma} \times A^\sigma$ satisfying the following properties:

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

A triple $(\mathcal{A}, P, \text{Cov})$, where $\mathcal{A}$ is pre-applicative structure, $P$ is a property on $\mathcal{A}$, and $\text{Cov}$ is a $P$-Grothendieck topology, is called a $P$-site.

Condition (0) is needed to restrict attention to elements having the property $P$. 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 $(\mathcal{A}, \preceq)$. 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. Condition (5) is needed to take care of the 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 if $\text{Cov}(C, M)$ for at least two distinct covers $C$. We say that $M \in A^\sigma$ is stubborn if $\text{Cov}(M) = \text{triv}(M)$ (thus every stubborn element is simple). We say that a $P$-site $(\mathcal{A}, P, \text{Cov})$ is scenic if all elements of the form $\text{app}(M, N)$ (or $MN$) are simple.

From now on, we only consider scenic $P$-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.

---

2Readers who are anxious to see the full set of conditions should take a look at definition 8.1.
Definition 4.4 Let \( (\mathcal{A}, \mathcal{P}, \text{Cov}) \) be a \( \mathcal{P} \)-site. Properties (P1)-(P3) are defined as follows:

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

(2) If \( \mathcal{P}(M, \sigma) \) and \( M \geq N \), then \( \mathcal{P}(N, \sigma) \).

(3) 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 \( P_{\sigma} \) 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.

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

(1) If \( \sigma \in S(M) \), then \( M \in P_{\sigma} \).

(2) If \( \sigma \in S(M) \) and \( M \geq N \), then \( \sigma \in S(N) \).

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

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

(1) \( S_{\sigma} \subseteq P_{\sigma} \).

(2) If \( M \in S_{\sigma} \) and \( M \geq N \), then \( N \in S_{\sigma} \).

(3) 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, \( S \) can also be viewed as a mapping

\[
S: \mathcal{A} \to \text{Sets}.
\]

Then, (S2) means that \( M \geq N \) implies \( S(M) \subseteq S(N) \). Thus, \( S \) is in fact a functor

\[
S: \mathcal{A}^{\text{op}} \to \text{Sets},
\]

viewing \( \mathcal{A}^{\text{op}} \) equipped with the preorder \( \geq \), the opposite of the preorder \( \leq \), 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 \( S_{\sigma} \) is nonempty and contains all stubborn elements in \( P_{\sigma} \) (because stubborn elements have the empty cover).
By (P3), if \( M \in P_{\sigma \to \tau} \) 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 \( M \in P_{\sigma \to \tau} \) 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 [13]). 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 \( a \to b \) from \( a \) to \( b \) iff \( a \leq b \). 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 \( a \geq b \)).

Let \( F: \mathcal{A}^{op} \to \mathbf{Sets} \) be a functor. Thus, \( F \) assigns a set \( F(a) \) to every element \( a \in \mathcal{A} \), and a function \( F(b \to a): F(b) \to F(a) \) to every pair \( a, b \in \mathcal{A} \) such that \( a \leq b \) (with the usual functorial conditions). For the sake of brevity, let us denote \( F(b \to a): F(b) \to F(a) \) as \( F^b_a: F(b) \to F(a) \).

Given any \( a \in \mathcal{A} \), for any \( x \in F(a) \) and any \( b \in \mathcal{A} \) such that \( b \leq a \), \( F^b_a(x) \) is a member of the set \( F(b) \) that we will also denote as \( x \mid b \). We can think of \( x \mid b \) as the restriction of \( x \in F(a) \) to \( b \).

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

\[
x_c|d = x_d \quad \text{for every } d \leq c.
\]

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 \( \mathbf{Cov}(C, a) \)), 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 \( a \in \mathcal{A} \), \( F(a) \subseteq T \) and \( \sigma \in F(a) \) implies that \( a \in P_\sigma \).

Since a cover is a sieve, \( d \leq c \) 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

\[
x_c|c \land d = x_d|c \land d.
\]

If the functor \( F \) is a sheaf and has the property that the maps \( F^b_a: F(b) \to F(a) \) (with \( a \leq b \)) 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 \( S: \mathcal{A}^{op} \to \mathbf{Sets} \) defined earlier is such that \( M \geq N \) implies \( S(M) \subseteq S(N) \). Thus, it is indeed technically true that definition 4.5 means that the functor \( S \) is a \( \mathcal{P} \)-sheaf with respect to the Grothendieck topology defined by \( \mathbf{Cov} \).
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 \((A, \mathcal{P}, \text{Cov})\) be a \( \mathcal{P} \)-site. The sets \( r[\sigma] \) of realizers of \( \sigma \) are defined as follows:

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

Note that instead of defining the family of sets \( r[\sigma] \), we could have defined a binary relation \( r \) such that \( M \stackrel{r}{\rightarrow} \sigma \) iff \( M \in r[\sigma] \). This is the more standard way of defining realizability. Another important point worth noting is that in the definition of \( r[\sigma \rightarrow \tau] \), 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 \((A, \mathcal{P}, \text{Cov})\), if \( \mathcal{P} \) satisfies conditions \((P1)-(P3)\), then \( (r[\sigma])_{\sigma \in \mathcal{T}} \) has the sheaf property, and each \( r[\sigma] \) contains all stubborn elements in \( P_\sigma \).

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

We now consider the induction step.

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

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

(S3). Assume that \( \text{Cov}_{\sigma \rightarrow \tau}(C, M) \), and that \( M' \in r[\sigma \rightarrow \tau] \) 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 \( N, \text{ if } N \in r[\sigma] \), then \( MN \in r[\tau] \). 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 \( r[\sigma] \). By (S1), \( N \in P_\sigma \). By the induction hypothesis, all stubborn elements in \( P_\sigma \) are in \( r[\tau] \). 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 \( M \in r[\sigma \rightarrow \tau] \).

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

\(^3\)In fact, if \( r[\sigma] = P_\sigma \), (S3) holds trivially even at nonbase types. This remark is useful if we allow type variables.
If \( \tau \) is a base type, then \( r[\tau] = P_\tau \) and \( MN \in r[\tau] \).

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

Since \( M \in P_{\sigma \to \tau} \) and \( MN \in r[\tau] \) whenever \( N \in r[\sigma] \), we conclude that \( M \in r[\sigma \to \tau] \). \( \square \)

We now need to relate \( \lambda \)-terms and realizers.

6 Interpreting terms in \( \lambda^- \) in Pre-Applicative Structures

We show how terms in \( \lambda^- \) are interpreted in pre-applicative structures. For this, we define a *meaning function*.

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

\[
A[x]\rho = \rho(x) \\
A[\lambda x: \sigma. M]\rho = \text{abst}(f),
\]

where \( f \) is the function defined such that, \( f(a) = A[M]\rho[x: = a] \), for every \( a \in A^\sigma \)

It is routine to show that the following property holds:

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

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

7 The Realizability Theorem For \( \lambda^- \)

In this section, we prove the realizability lemma (lemma 7.6) for \( \lambda^- \), 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 \( \langle A, P, \text{Cov} \rangle \) is well-behaved iff the following conditions hold:

1. For any \( a \in A^\sigma \), any \( \varphi \in [A^\sigma \Rightarrow A^\tau] \), if \( \text{abst}(\varphi) \) exists, \( \text{Cov}_r(C, \text{app}(\text{abst}(\varphi), a)) \), 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**

1. For any \( a \in A^\sigma \), if \( \text{Cov}_r(C, \text{app}(A[\lambda x : \sigma. M]_\rho, a)) \) and \( C \) is a nontrivial cover, then \( A[\lambda x : \sigma. M]_\rho[x := a] \) 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 \( \langle A, P, \text{Cov} \rangle \), properties (P4) and (P5) are defined as follows:

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

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

**Definition 7.4**

- **(P4)** If \( A[M]_\rho \in P_r \), then \( A[\lambda x : \sigma. M]_\rho \in P_{\sigma \rightarrow \tau} \).

- **(P5)** If \( a \in P_{\sigma} \) and \( A[M]_\rho[x := a] \in P_{\sigma} \), then \( \text{app}(A[\lambda x : \sigma. M]_\rho, a) \in P_r \).

**Lemma 7.5** Given a well-behaved scenic site \( \langle A, P, \text{Cov} \rangle \) and a family \( P \) satisfying conditions (P1)-(P5), for every \( \rho \) such that \( \rho(y) \in r[\gamma] \) for every \( y : \gamma \in \text{FV}(M) \), if for every \( a \), \( (a \in r[\sigma] \) implies \( A[M]_\rho[x := a] \in r[\tau] \)), then \( A[\lambda x : \sigma. M]_\rho \in r[\sigma \rightarrow \tau] \).

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

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

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

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

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

We now prove the main realizability lemma for \( \lambda^- \).

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

**Proof.** We proceed by induction on the structure of \( M \). If \( M \) is a variable, then \( A[x]\rho = \rho(x) \in r[\sigma] \) by the assumption on \( \rho \).

If \( M = M_1N_1 \), where \( M_1 \) has type \( \sigma \rightarrow \tau \) and \( N_1 \) has type \( \sigma \), by the induction hypothesis,

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

By the definition of \( r[\sigma \rightarrow \tau] \), we get \( \text{app}(A[M_1]\rho, A[N_1]\rho) \in r[\tau] \), i.e., \( A[(M_1N_1)]\rho \in r[\tau] \), by definition 6.1.

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

If \( M \) is a closed term of type \( \sigma \), the independence condition of definition 6.1 implies that \( A[M]\rho \) is independent of \( \rho \), and thus we denote it as \( \lambda^- M \). We get the following important theorem for \( \lambda^- \).

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

**Proof.** Apply lemma 7.6 to the closed term \( M \) of type \( \sigma \) and to any arbitrary valuation \( \rho \). \( \Box \)

8 \( P \)-Realizability for the Arrow, Product, and Sum Types

In this section, we extend the semantic notion of realizability defined in section 5 to the calculus \( \lambda^- x^+ \). 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 8.1 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 $P_\sigma$ is a nonempty subset of $\mathcal{A}^\sigma$. A $\mathcal{P}$-cover algebra (or $\mathcal{P}$-Grothendieck topology) on $\mathcal{A}$ is a family Cov of binary relations Cov on $2^\mathcal{A}^\sigma \times \mathcal{A}^\sigma$ satisfying the following properties:

1. Cov(C, M) implies $M \in P_\sigma$ (equivalently, $\mathcal{P}(M, \sigma)$).
2. If $M \in P_\sigma$, then Cov({N | $N \leq M$}, M) ($M \in P_\sigma$) is covered by the principal sieve on M).
3. (stability) If Cov(C, M) and $N \leq M$, then Cov({Q | $Q \in C$, $Q \leq N$}, N).
4. (transitivity) If Cov(C, M), D is a sieve on M, and Cov({Q | $Q \in D$, $Q \leq N$}, N) for every $N \in C$, then Cov(D, M).
5. If Cov(M) = triv(M), then Cov(MN) = triv(MN), and if Cov(C, M) and Cov(D, MN) with C and D nontrivial, then for every $Q \in D$, there is some $M' \in C$ such that $Q \leq M' N$.
6. If Cov(M) = triv(M), then Cov($\pi_1(M)$) = triv($\pi_1(M)$), Cov($\pi_2(M)$) = triv($\pi_2(M)$), and if Cov(C, M) and Cov(D, $\pi_1(M)$) (resp. Cov(D, $\pi_2(M)$)) with C and D nontrivial, then for every $Q \in D$, there is some $M' \in C$ such that $Q \leq \pi_1(M')$ (resp. $Q \leq \pi_2(M')$).

A triple $(\mathcal{A}, P, \text{Cov})$, where $\mathcal{A}$ is pre-applicative structure, $P$ is a property on $\mathcal{A}$, and Cov is a $P$-Grothendieck topology, is called a $P$-site.

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

Definition 8.2 We say that $M \in \mathcal{A}^\sigma$ is simple iff Cov(C, M) for at least two distinct covers C. We say that $M \in \mathcal{A}^\sigma$ is stubborn iff Cov(M) = triv(M) (thus every stubborn element is simple). We say that a $P$-site $(\mathcal{A}, P, \text{Cov})$ is scenic iff all elements of the form app$(M, N)$ (or $MN$), $\pi_1(M)$, and $\pi_2(M)$ are simple.

Definition 4.4 is extended as follows.

Definition 8.3 Let $(\mathcal{A}, P, \text{Cov})$ be a $P$-site. Properties (P1)-(P3) are defined as follows:

1. $\mathcal{P}(M, \sigma)$, for some stubborn element $M \in \mathcal{A}^\sigma$.
2. If $\mathcal{P}(M, \sigma)$ and $M \geq N$, then $\mathcal{P}(N, \sigma)$.
3. (P3)
   1. If Cov$_{\sigma \rightarrow \tau}$(C, M), $\mathcal{P}(N, \sigma)$, and $\mathcal{P}(M' N, \tau)$ whenever $M' \in C$, then $\mathcal{P}(M N, \tau)$.
   2. If 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 conditions (P1)-(P3) of definition 8.3. Condition (P1) says that each $P_\sigma$ contains some stubborn element.

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 8.1.

Definition 4.5 remains unchanged. The notion of $\mathcal{P}$-realizability is defined as follows.
Definition 8.4 Let \((A,\mathcal{P},\text{Cov})\) be a \(\mathcal{P}\)-site. The sets \(r[\sigma]\) of realizers of \(\sigma\) are defined as follows:

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

We now prove a generalization of lemma 5.2.

Lemma 8.5 Given a scenic \(\mathcal{P}\)-site \((A,\mathcal{P},\text{Cov})\), if \(\mathcal{P}\) satisfies conditions \((P1)-(P3)\), then \((r[\sigma])_{\sigma \in \mathcal{T}}\) has the sheaf property, and each \(r[\sigma]\) contains all stubborn elements in \(P_\sigma\).

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 product and sum types.

\begin{itemize}
  \item[(S1).] This is trivial by the definitions of \(r[\sigma \to \tau]\), \(r[\sigma \times \tau]\), and \(r[\sigma + \tau]\),
  \item[(S2).] There are three cases depending on the type.
    \begin{enumerate}
      \item Arrow type \(\sigma \to \tau\). The proof is as in lemma 5.2.
      \item Product type \(\sigma \times \tau\). Assume that \(M \succeq M'\) for \(M \in r[\sigma \times \tau]\). We need to prove that \(M' \in P_{\sigma \times \tau}\), \(\pi_1(M') \in r[\sigma]\), and \(\pi_2(M') \in r[\tau]\). Since \(M \in r[\sigma \times \tau]\), by (S1), \(M \in P_{\sigma \times \tau}\), and by (P2) \(M' \in P_{\sigma \times \tau}\). Since \(M \in r[\sigma \times \tau]\), we have \(\pi_1(M) \in r[\sigma]\) and \(\pi_2(M) \in r[\tau]\). 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 \(\pi_1(M') \in r[\sigma]\) and \(\pi_2(M') \in r[\tau]\).
      \item Sum type \(\sigma + \tau\). Assume that \(M \succeq M'\) for \(M \in r[\sigma + \tau]\). Since \(M \in r[\sigma + \tau]\), we have
        \[
        \text{Cov}_{\sigma + \tau}(\{\text{inl}(M_1) \mid M_1 \in r[\sigma] \text{ and } M \succeq \text{inl}(M_1)\} \cup \\
        \{\text{inr}(M_2) \mid M_2 \in r[\tau] \text{ and } M \succeq \text{inr}(M_2)\}), M).
        \]
    \end{enumerate}
\end{itemize}

Consider the cover \(D\) of \(M\):

\[
D = \{\text{inl}(M_1) \mid M_1 \in r[\sigma] \text{ and } M \succeq \text{inl}(M_1)\} \cup \\
\{\text{inr}(M_2) \mid M_2 \in r[\tau] \text{ and } M \succeq \text{inr}(M_2)\}.
\]

By property (3) of definition 8.1, 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 8.1, \(M' \in D\), and it is clear that

\[
\{Q \mid Q \in D, Q \preceq M'\} = \{\text{inl}(M_1) \mid M_1 \in r[\sigma] \text{ and } M' \succeq \text{inl}(M_1)\} \cup \\
\{\text{inr}(M_2) \mid M_2 \in r[\tau] \text{ and } M' \succeq \text{inr}(M_2)\}.
\]

Then, we have

\[
\text{Cov}_{\sigma + \tau}(\{\text{inl}(M_1) \mid M_1 \in r[\sigma] \text{ and } M' \succeq \text{inl}(M_1)\} \cup \\
\{\text{inr}(M_2) \mid M_2 \in r[\tau] \text{ and } M' \succeq \text{inr}(M_2)\}), M').
\]
showing that $M' \in r[\sigma + \tau]$.

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

1. Arrow type $\sigma \to \tau$. 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 $M' \in r[\sigma \times \tau]$ whenever $M' \in C$, where $M$ is simple. By property (0) of definition 8.1, we have $M \in P_{\sigma \times \tau}$. We need to show that $\pi_1(M) \in r[\sigma]$ and $\pi_2(M) \in r[\tau]$.

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 $P_{\sigma}$ are in $r[\sigma]$ and all stubborn elements in $P_{\tau}$ are in $r[\tau]$. Thus, when $M$ is stubborn, $\pi_1(M) \in r[\sigma]$ and $\pi_2(M) \in r[\tau]$.

Next, assume that $M$ is not stubborn. Since $M' \in r[\sigma \times \tau]$ whenever $M' \in C$, we have $\pi_1(M') \in r[\sigma]$ and $\pi_2(M') \in r[\tau]$. 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 $\sigma$ is a base type, then $r[\sigma] = P_{\sigma}$ and $\pi_1(M) \in r[\sigma]$. By (S3), we conclude that $\pi_1(M) \in r[\sigma]$.

Let us now consider the case where $\sigma$ is not a base type, the case where $\tau$ is not a base type being similar. Then, $\pi_1(M) \in P_{\sigma}$ and $\pi_1(M) \in P_{\tau}$ is simple (since the site is scenic). We use (S3) to prove that $\pi_1(M) \in r[\sigma]$. 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 8.1, for every $Q \in D$, there is some $M' \in C$ such that $Q \leq \pi_1(M')$. By the assumption, $M' \in r[\sigma \times \tau]$. This implies that $\pi_1(M') \in r[\sigma]$, and by the induction hypothesis and (S2), we have $Q \in r[\sigma]$. By (S3), we conclude that $\pi_1(M) \in r[\sigma]$.

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

$$D = \{\text{inl}(M_1) \mid M_1 \in r[\sigma] \text{ and } M \succeq \text{inl}(M_1)\} \cup \{\text{inr}(M_2) \mid M_2 \in r[\tau] \text{ and } M \succeq \text{inr}(M_2)\}.$$  

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

$$\text{Cov}_{\sigma + \tau}(\{\text{inl}(M_1) \mid M_1 \in r[\sigma] \text{ and } N \succeq \text{inl}(M_1)\} \cup \{\text{inr}(M_2) \mid M_2 \in r[\tau] \text{ and } N \succeq \text{inr}(M_2)\}, N).$$

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

$$\{Q \mid Q \in D, Q \preceq N\} = \{\text{inl}(M_1) \mid M_1 \in r[\sigma] \text{ and } N \succeq \text{inl}(M_1)\} \cup \{\text{inr}(M_2) \mid M_2 \in r[\tau] \text{ and } N \succeq \text{inr}(M_2)\}.$$  

Then, by property (4) of definition 8.1, we have $\text{Cov}_{\sigma + \tau}(D, M)$, that is, $M \in r[\sigma + \tau]$. □
9 Interpreting $\lambda$-Terms in $\lambda^{\rightarrow, x, +}$

We extend definition 6.1 to take care of product and sum types.

**Definition 9.1** Given a pre-applicative structure $A$, a valuation, or environment, is any function $\rho: \mathcal{X} \rightarrow A$, such that $\rho(x) \in A^\sigma$ if $x: \sigma$. A meaning function for $A$ is a partial function $A[-](\_)$ from pairs of ($\alpha$-equivalence classes of) terms and valuations to $A$, such that $A[M] \rho$ is defined whenever $M: \sigma$, in which case $A[M] \rho \in A^\sigma$. In addition, a meaning function satisfies the following conditions:

- $A[x] \rho = \rho(x)$
- $A[MN] \rho = \text{app}(A[M] \rho, A[N] \rho)$
- $A[\lambda x: \sigma. M] \rho = \text{abst}(f)$,

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

- $A[\pi_1(M)] \rho = \pi_1(A[M] \rho)$
- $A[\pi_2(M)] \rho = \pi_2(A[M] \rho)$
- $A[(M_1, M_2)] \rho = (A[M_1] \rho, A[M_2] \rho)$
- $A[\text{inl}(M)] \rho = \text{inl}(A[M] \rho)$
- $A[\text{inr}(M)] \rho = \text{inr}(A[M] \rho)$

Using an induction on the structure of terms, it is easily verified that $LT_\beta[M] \rho = M[\varphi]$, where $\varphi$ is the substitution defined by the restriction of $\rho$ to $FV(M)$. As far as realizability is concerned, if $M: \sigma$, then $LT_\beta[M] \rho$ is a typed $\lambda$-term realizing $\sigma$. Definition 8.4 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 $HR\rho$ of definition 3.6. A valuation is a function $\rho$ such that $\rho(x) = \langle k, \sigma \rangle$ for every $x: \sigma$, where $k \in \mathbb{N}$. Thus, given a term $M$ such that $FV(M) = \{x_1: \sigma_1, \ldots, x_m: \sigma_m\}$, a valuation $\rho$ defines a finite sequence $\langle \rho_1, \ldots, \rho_m \rangle$ of natural numbers, where $\rho_i = \rho(x_i)$. It is easily shown by induction on the structure of $M: \sigma$ that $HR\rho[M] \rho = \langle \varphi_e(\rho_1, \ldots, \rho_m), \sigma \rangle$, where $e$ is the index a total recursive function $\varphi_e$ in the arguments $\langle \rho_1, \ldots, \rho_m \rangle$. Thus, every typed $\lambda$-terms can be interpreted in $HR\rho$, and $HR\rho[M] \rho$ is given by a function recursive in the restriction of $\rho$ to $FV(M)$. As far as realizability is concerned, if $M: \sigma$, then $HR\rho[M] \rho \in r[\sigma]$ yields a realizer for $\sigma$ which is given by a recursive function of $\rho$. In this case, definition 8.4 is equivalent to Kleene's recursive realizability (for $\rightarrow, x$, and $\oplus$).

10 The Realizability Theorem For $\lambda^{\rightarrow, x, +}$

In this section, we generalize the realizability lemma (lemma 7.6) and its main corollary (theorem 7.7) to the calculus $\lambda^{\rightarrow, x, +}$. In order to do so, we need to add conditions to definition 7.1 to take care of product and sum types.
Definition 10.1 We say that a site \( (A, P, \text{Cov}) \) is well-behaved iff the following conditions hold:

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

2. If \( \text{Cov}_s(C, \pi_1((a_1, a_2))) \) and \( C \) is a nontrivial cover, then \( c \preceq a_1 \) for every \( c \in C \).
   \[
   \text{Cov}_s(C, \pi_2((a_1, a_2))) \quad \text{and} \quad C \text{ is a nontrivial cover, then } c \preceq a_2 \text{ for every } c \in C.
   \]

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 \( \text{Cov}_{s+T}(C, p), \text{Cov}_s(D, \text{app}([M, N], p)) \), 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 \( d \preceq \text{app}(g, p_2) \), where \( f \in A^{\sigma \rightarrow \delta} \) and \( g \in A^{\tau \rightarrow \delta} \).

In view of definition 9.1, definition 10.1 implies the following conditions.

Definition 10.2

1. For any \( a \in A^\sigma \), if \( \text{Cov}_s(C, \text{app}([Mx: \sigma]. M], p, a)) \) and \( C \) is a nontrivial cover, then \( c \preceq A[M]p[x := a] \) for every \( c \in C \).

2. If \( \text{Cov}_s(C, \pi_1(A[(M_1, M_2)]\rho)) \) and \( C \) is a nontrivial cover, then \( c \preceq A[M_1]\rho \) for every \( c \in C \).

3. If \( \text{Cov}(p) = \text{triv}(p) \), then \( \text{Cov}([M, N], p) = \text{triv}(\text{app}([M, N], p)) \), and if \( \text{Cov}_{s+T}(C, p), \text{Cov}_s(D, \text{app}([M, N], p)) \), 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 \( d \preceq \text{app}(g, p_2) \), where \( f \in A^{\sigma \rightarrow \delta} \) and \( g \in A^{\tau \rightarrow \delta} \).

We also need to add conditions to definition 7.3 to take care of product and sum types.

Definition 10.3 Given a well-behaved site \( (A, P, \text{Cov}) \), 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 \( (a_1, a_2) \in P_{\sigma \times \tau} \).

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 \( [a_1, a_2] \in P_{(\sigma + \tau) \rightarrow \delta} \).

(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 \( \pi_1((a_1, a_2)) \in P_\sigma \) and \( \pi_2((a_1, a_2)) \in P_\tau \).

3. If \( \text{Cov}_{s+T}(C, p), f \in P_{\sigma \rightarrow \delta}, g \in P_{\tau \rightarrow \delta}, \text{app}(f, p_1) \in P_\delta \) whenever \( \text{inl}(p_1) \in C \), and \( \text{app}(g, p_2) \in P_\delta \) whenever \( \text{inr}(p_2) \in C \), then \( \text{app}([f, g], p) \in P_\delta \).
It is easy to verify that \( \text{app}(f, g, p) \in P_\delta \) is stubborn if \( p \in P_{\sigma + \tau} \) is stubborn, \( f \in P_{\sigma - \delta} \), and \( g \in P_{\tau - \delta} \). This follows from condition (3) of definition 10.1.

In view of definition 9.1, definition 10.3 implies the following conditions.

**Definition 10.4**

(P4)

1. If \( A[M]\rho \in P_\tau \), then \( A[\lambda x: \sigma. M]\rho \in P_{\sigma - \tau} \).
2. If \( A[M]\rho \in P_\sigma \) and \( A[N]\rho \in P_\tau \), then \( A[M, N]\rho \in P_{\sigma + \tau} \).
3. If \( A[M]\rho \in P_\sigma \), then \( \text{inl}(A[M]\rho) \in P_{\sigma + \tau} \), and if \( A[M]\rho \in P_\tau \), then \( \text{inr}(A[M]\rho) \in P_{\sigma + \tau} \).
4. If \( A[M]\rho \in P_{\sigma - \delta} \) and \( A[N]\rho \in P_{\tau - \delta} \), then \( A[M, N]\rho \in P_{(\sigma + \tau) - \delta} \).

(P5)

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

We have the following generalization of lemma 7.5.

**Lemma 10.5** Given a well-behaved scenic site \((A, P, \text{Cov})\), and a family \( P \) satisfying conditions (P1)-(P5), for every \( \rho \), the following properties hold: (1) If \( \rho(y) \in r[\gamma] \) for every \( y : \gamma \in \text{FV}(M) \), and for every \( a \), \((a \in r[\sigma] \implies A[M]\rho[x := a] \in r[\tau]) \), then \( A[\lambda x: \sigma. M]\rho \in r[\sigma \rightarrow \tau] \). (2) If \( A[M]\rho \in r[\sigma] \) and \( A[N]\rho \in r[\tau] \), then \( A[M, N]\rho \in r[\sigma \times \tau] \); (3) If \( A[M]\rho \in r[\sigma \rightarrow \delta] \) and \( A[N]\rho \in r[\tau \rightarrow \delta] \), then \( A[M, N]\rho \in r[\sigma + \tau \rightarrow \delta] \).

**Proof.** It is similar to the proof of lemma 7.5, except that we need to prove more clauses. By lemma 8.5, we know that the sets of the form \( r[\sigma] \) have the properties (S1)-(S3).

1. This has already been proved in lemma 7.5.

2. We need to show that \( A[M, N]\rho \in P_{\sigma + \tau} \), \( \pi_1(A[M, N]\rho) \in r[\sigma] \), and \( \pi_2(A[M, N]\rho) \in r[\tau] \). Since \( A[M]\rho \in r[\sigma] \) and \( A[N]\rho \in r[\tau] \), by (S1), \( A[M]\rho \in P_\sigma \) and \( A[N]\rho \in P_\tau \). By (P4)(2), we get \( A[M, N]\rho \in P_{\sigma + \tau} \). By (P5)(2), we also have \( \pi_1(A[M, N]\rho) \in P_\sigma \) and \( \pi_2(A[M, N]\rho) \in P_\tau \). If \( \sigma \) is a base type then \( r[\sigma] = P_\sigma \) and \( \pi_1(A[M, N]\rho) \in r[\sigma] \). Similarly, if \( \tau \) is a base type then \( r[\tau] = P_\tau \) and \( \pi_2(A[M, N]\rho) \in r[\tau] \).

If both \( \sigma \) and \( \tau \) are nonbase types, \( \pi_1(A[M, N]\rho) \in P_\sigma \) and \( \pi_2(A[M, N]\rho) \in P_\tau \) are simple (since the site is scenic). We prove that \( \pi_1(A[M, N]\rho) \in r[\sigma] \) and \( \pi_2(A[M, N]\rho) \in r[\tau] \) using (S3). We consider the case of \( \pi_1(A[M, N]\rho) \), the case of \( \pi_2(A[M, N]\rho) \) being similar. The case where \( \pi_1(A[M, N]\rho) \) is stubborn is trivial. Otherwise, assume that \( \text{Cov}_\sigma(C, \pi_1(A[M, N]\rho)) \), where \( C \) is a nontrivial cover. We need to prove that \( c \in r[\sigma] \) whenever \( c \in C \). By condition (2) of definition 10.2, \( c \preceq A[M]\rho \) for every \( c \in C \). Since \( A[M]\rho \in r[\sigma] \) and \( c \preceq A[M]\rho \), by (S2), we have \( c \in r[\sigma] \).

3. We need to prove that \( A[M, N]\rho \in P_{(\sigma + \tau) - \delta} \), and that \( \text{app}(A[M, N]\rho, p) \in r[\delta] \), for every \( p \in r[\sigma + \tau] \). Since \( A[M]\rho \in r[\sigma \rightarrow \delta] \) and \( A[N]\rho \in r[\tau \rightarrow \delta] \), by (S2), we have \( A[M]\rho \in P_{\sigma - \delta} \) and \( A[N]\rho \in P_{\tau - \delta} \), and by (P4)(4), we get \( A[M, N]\rho \in P_{(\sigma + \tau) - \delta} \).
Next, we prove that \( \text{app}(A[[M, N]]\rho, p) \in P_\delta \). Assume that the hypothesis of (3) holds. By assumption, \( p \in r[\sigma + \tau], A[M]\rho \in r[\sigma \rightarrow \delta], \) and \( A[N]\rho \in r[\tau \rightarrow \delta] \). By (S1), we have \( p \in P_{\sigma + \tau}, A[M]\rho \in P_{\sigma \rightarrow \delta}, \) and \( A[N]\rho \in P_{\tau \rightarrow \delta} \). If \( p \) is stubborn, we have shown that \( \text{app}(A[[M, N]]\rho, p) \in P_\delta \) is stubborn, and thus \( \text{app}(A[[M, N]]\rho, p) \in r[\delta] \) by (S3).

Otherwise, since \( p \in r[\sigma + \tau] \), the cover \( C \) given by
\[
C = \{ \text{inl}(p_1) \mid p_1 \in r[\sigma] \text{ and } p \succeq \text{inl}(p_1) \} \cup \\
\{ \text{inr}(p_2) \mid p_2 \in r[\tau] \text{ and } p \succeq \text{inr}(p_2) \}
\]
is a nontrivial cover, and \( \text{Cov}_{\sigma + \tau}(C, p) \). Then, since by the assumptions of the lemma, \( A[M]\rho \in r[\sigma \rightarrow \delta] \) and \( A[N]\rho \in r[\tau \rightarrow \delta] \), we have \( \text{app}(A[M]\rho, p_1) \in r[\delta] \) whenever \( \text{inl}(p_1) \in C \), and \( \text{app}(A[N]\rho, p_2) \in r[\delta] \) whenever \( \text{inr}(p_2) \in C \), since \( p_1 \in r[\sigma] \) and \( p_2 \in r[\tau] \), by definition of \( C \). Now, the conditions of (P5)(3) are met for \( C \), and we have \( \text{app}(A[[M, N]]\rho, p) \in P_\delta \). If \( \delta \) is a base type, then \( \text{app}(A[[M, N]]\rho, p) \) is simple (since the site is scenic). We use (S3) to prove that \( \text{app}(A[[M, N]]\rho, p) \in r[\delta] \). The case where \( \text{app}(A[[M, N]]\rho, p) \) is stubborn is trivial.

Otherwise, assume that \( \text{Cov}_\delta(D, \text{app}(A[[M, N]]\rho, p)) \), where \( D \) is a nontrivial cover. Since \( p \in r[\sigma + \tau] \), the cover \( C \) given by
\[
C = \{ \text{inl}(p_1) \mid p_1 \in r[\sigma] \text{ and } p \succeq \text{inl}(p_1) \} \cup \\
\{ \text{inr}(p_2) \mid p_2 \in r[\tau] \text{ and } p \succeq \text{inr}(p_2) \}
\]
is a nontrivial cover, and \( \text{Cov}_{\sigma + \tau}(C, p) \). Since \( C \) and \( D \) are nontrivial, by condition (3) of definition 10.2, for every \( d \in D \), either there is some \( \text{inl}(p_1) \in C \) such that \( d \preceq \text{app}(A[M]\rho, p_1) \), or there is some \( \text{inr}(p_2) \in C \) such that \( d \preceq \text{app}(A[N]\rho, p_2) \). Since by definition of \( C \), \( p_1 \in r[\sigma] \) and \( p_2 \in r[\tau] \), and by assumption, \( A[M]\rho \in r[\sigma \rightarrow \delta] \) and \( A[N]\rho \in r[\tau \rightarrow \delta] \), we have \( \text{app}(A[M]\rho, p_1) \in r[\delta] \) and \( \text{app}(A[N]\rho, p_2) \in r[\delta] \), and by (S2), we get \( d \in r[\delta] \). Finally, by (S3), we have \( \text{app}(A[[M, N]]\rho, p) \in r[\delta] \).

We now prove the main realizability lemma for \( \lambda^{\rightarrow, \times, +} \).

**Lemma 10.6** Given a well-behaved scenic site \( (A, P, \text{Cov}) \), if \( P \) is a family satisfying conditions (P1)-(P5), then for every term \( M \) of type \( \sigma \), for every valuation \( \rho \) such that \( \rho(y) \in r[\gamma] \) for every \( y: \gamma \in FV(M) \), we have \( A[M]\rho \in r[\sigma] \).

**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 \( M = (M_1, N_1) \), where \( M_1 \) has type \( \sigma \) and \( N_1 \) has type \( \tau \), then by the induction hypothesis, \( A[M_1]\rho \in r[\sigma] \) and \( A[N_1]\rho \in r[\tau] \). By lemma 10.5, we have \( A[(M_1, N_1)]\rho \in r[\sigma \times \tau] \).

If \( M = \pi_1(M_1) \) where \( M_1 \) has type \( \sigma \times \tau \), then by the induction hypothesis, \( A[M_1]\rho \in r[\sigma \times \tau] \). By the definition of \( r[\sigma \times \tau] \), this implies that \( \pi_1(A[M_1]\rho) \in r[\sigma] \), that is, \( A[\pi_1(M_1)]\rho \in r[\sigma] \), by definition 6.1. Similarly, we get \( A[\pi_2(M_1)]\rho \in r[\sigma] \).
If \( M = \text{inl}(M_1) \) where \( M \) has type \( \sigma + \tau \), then by the induction hypothesis, \( \mathcal{A}[M_1]\rho \in r[\sigma] \). By (P4)(3), we have \( \text{inl}(\mathcal{A}[M_1]\rho) \in P_{\sigma+\tau} \). Consider the cover \( D \) of \( \text{inl}(\mathcal{A}[M_1]\rho) \):

\[
D = \{ \text{inl}(p_1) \mid p_1 \in r[\sigma] \text{ and } \text{inl}(\mathcal{A}[M_1]\rho) \succeq \text{inl}(p_1) \} \cup \\
\{ \text{inr}(p_2) \mid p_2 \in r[\tau] \text{ and } \text{inl}(\mathcal{A}[M_1]\rho) \succeq \text{inr}(p_2) \}.
\]

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 \( \mathcal{A}[M_1]\rho \in r[\sigma] \), and by (S2), \( p_1 \in r[\sigma] \) 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 8.1, 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 6.1, \( \mathcal{A}[\text{inl}(M_1)]\rho = \text{inl}(\mathcal{A}[M_1]\rho) \), we have \( \mathcal{A}[\text{inl}(M_1)]\rho \in r[\sigma+\tau] \). 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 \( M_1, N_1 \), we have \( \mathcal{A}[M_1]\rho \in r[\sigma \rightarrow \delta] \), and \( \mathcal{A}[N_1]\rho \in r[\tau \rightarrow \delta] \). Thus, by lemma 10.5, we have \( \mathcal{A}[[M_1, N_1]]\rho \in r[(\sigma + \tau) \rightarrow \delta] \).

Theorem 7.7 is generalized to the calculus \( \lambda^{-,X,+,\bot} \) as follows.

**Theorem 10.7** 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 \( \sigma \), 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 10.6 to the closed term \( M \) of type \( \sigma \) and to any arbitrary valuation \( \rho \). □

### 11 Syntax of the Typed \( \lambda \)-Calculus \( \lambda^{-,X,+,\bot} \)

In this and the remaining sections, we consider the simply-typed \( \lambda \)-calculus obtained by adding the (constant) type \( \bot \) to the type constructors \( \rightarrow, X, \) and \( + \). First, we review the syntax. The type-checking rules of the system are summarized in the following definition.

**Definition 11.1** The terms of the typed \( \lambda \)-calculus \( \lambda^{-,X,+,\bot} \) 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 : \bot}{\triangledown_{\sigma}(M) : \sigma} \quad \text{(\( \bot \)-elim)}
\]

with \( \sigma \neq \bot \),

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

28
where \( x \in X_{\sigma}; \)

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

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

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

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

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

We also recall the reduction rules.

**Definition 11.2** The reduction rules of the system \( \lambda^-, \times, +, \bot \) are listed below:

\[
(\lambda x: \sigma. M)N \rightarrow M[N/x],
\]

\[
\pi_1((M, N)) \rightarrow M,
\]

\[
\pi_2((M, N)) \rightarrow N,
\]

\[
[M, N]\text{inl}(P) \rightarrow MP,
\]

\[
[M, N]\text{inr}(P) \rightarrow NP,
\]

\[
\nabla_{\sigma \rightarrow \tau}(M)N \rightarrow \nabla_\tau(M),
\]

\[
\pi_1(\nabla_{\sigma \times \tau}(M)) \rightarrow \nabla_\sigma(M),
\]

\[
\pi_2(\nabla_{\sigma \times \tau}(M)) \rightarrow \nabla_\tau(M),
\]

\[
[M, N]\nabla_{\sigma + \tau}(P) \rightarrow \nabla_\delta(P).
\]

The reduction relation defined by the rules of definition 11.2 is still denoted as \( \rightarrow_\beta \) (even though there are reductions other than \( \beta \)-reduction). Next, we need to generalize the definition of a pre-applicative structure to deal with the type \( \bot \).

### 12 Pre-Applicative Structures For \( \lambda^-, \times, +, \bot \)

In this section, the various concepts of a pre-applicative structure defined in section 3 are generalized to the calculus \( \lambda^-, \times, +, \bot \).

**Definition 12.1** A **pre-applicative \( \beta \)-structure** is a structure

\[
\mathcal{A} = \langle A, \text{fun}, \text{abst}, \Pi, \langle - , - \rangle, \text{inl}, \text{inr}, \langle - , - \rangle, \delta \rangle
\]

where

\[
A = (A^\sigma)_{\sigma \in T} \text{ is a family of (nonempty) sets called carriers;}
\]
$(\leq^\sigma)_{\sigma \in T}$ is a family of preorders, each $\leq^\sigma$ on $A^\sigma$;
abst^{\sigma,+}: [A^\sigma \Rightarrow A^\tau] \rightarrow A^{\sigma \rightarrow \tau}$, a family of partial operators;
\(\text{fun}^{\sigma,+}: A^{\sigma \rightarrow \tau} \rightarrow [A^\sigma \Rightarrow A^\tau]\), a family of (total) operators;
\((-,-)^{\sigma,+}: A^\sigma \times A^\tau \rightarrow A^{\sigma \times \tau}\), a family of partial pairing operators;
\(\Pi^{\sigma,+}: A^{\sigma \times \tau} \rightarrow A^\sigma \times A^\tau\), a family of (total) projection operators;
\([-, -]^{\sigma,+}: A^{\sigma \rightarrow \delta} \times A^{\tau \rightarrow \delta} \rightarrow A^{(\sigma + \tau) \rightarrow \delta}\), a family of partial copairing operators;
in^{\sigma,+}: A^\sigma \rightarrow A^{\sigma + \tau}, a family of (total) operators;
inr^{\sigma,+}: A^{\tau} \rightarrow A^{\sigma + \tau}, a family of (total) operators;
$\triangledown^{\sigma}: A^\perp \rightarrow A^\sigma$, 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

\[
\text{cinf}(h)(c) = \text{fun}(h)(\triangledown^{\sigma + \tau}(c)),
\]
for every \(c \in A^\perp\).

It is assumed that \(\text{fun}, \text{abst}, \Pi, (-,-), \text{inl}, \text{inr}, \) and \([-, -], \) and \(\triangledown\), are monotonic. Furthermore, the following conditions are satisfied

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

The operators \(\text{fun}^{\sigma,+}\) induce (total) operators

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

\[
\text{app}^{\sigma,+}(f, a) = \text{fun}^{\sigma,+}(f)(a).
\]

Then, condition (1) can be written as

1. \(\text{app}^{\sigma,+}(\text{abst}^{\sigma,+}(\varphi), a) \succeq \varphi(a), \) for all \(a \in A^\sigma, \) and \(\text{app}^{\sigma,+}(\triangledown^{\sigma \rightarrow \tau}(c), a) \succeq \triangledown^{\tau}(c), \) 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 \(\text{cinf}([(f, g)])(c) \succeq \triangledown^{\delta}(c), \) 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, $N \leq \text{inl}(M_1)$ implies that $N = \text{inl}(N_1)$ for some $N_1 \leq M_1$, $N \leq \text{inr}(M_1)$ implies that $N = \text{inr}(N_1)$ for some $N_1 \leq M_1$, and $N \leq \bigvee \sigma(M_1)$ implies that $N = \bigvee \sigma(N_1)$ for some $N_1 \leq M_1$.

We say that a pre-applicative $\beta$-structure is an applicative $\beta$-structure iff in conditions (1)-(3), $\geq$ is replaced by the identity relation $\equiv$.

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 $\bigvee \sigma$).

The projection operators $\Pi$ induce projections $\pi^{\sigma,\tau}_1 : A^{\sigma \times \tau} \rightarrow A^\sigma$ and $\pi^{\sigma,\tau}_2 : A^{\sigma \times \tau} \rightarrow A^\tau$, such that for every $a \in A^{\sigma \times \tau}$, if $\pi^{\sigma,\tau}_1(a) = (a_1, a_2)$, then
\[
\pi^{\sigma,\tau}_1(a) = a_1 \quad \text{and} \quad \pi^{\sigma,\tau}_2(a) = a_2.
\]

When $A$ is an applicative $\beta$-structure, then, in definition 12.1, conditions (1)-(3) amounts to
(1) $\text{fun}^{\sigma,\tau} \circ \text{abst}^{\sigma,\tau} = \lambda a \in A^\sigma. \bigvee \tau$; and $\text{fun}^{\sigma,\tau} \circ \bigvee \sigma \circ \tau = (\bigvee \sigma, \bigvee \tau);$
(2) $\Pi^{\sigma,\tau} \circ (\cdot, \cdot) = \lambda a \in A^\sigma. \bigvee \tau$; and $\Pi^{\sigma,\tau} \circ \bigvee \sigma \circ \tau = (\bigvee \sigma, \bigvee \tau);$
(3) $(\text{cinl}, \text{cinr}) \circ [-,-] = \text{fun}^{\sigma,\delta} \times \text{fun}^{\tau,\delta}$ on the domain of definition of $[-,-]$, and $\text{cinr} \circ [-,-] = \lambda f \in A^{\sigma \rightarrow \delta}. \lambda g \in A^{\tau \rightarrow \delta}. \bigvee \delta$, where $\lambda f \in A^{\sigma \rightarrow \delta}. \lambda g \in A^{\tau \rightarrow \delta}. \bigvee \delta$ denotes the constant function from $A^{\sigma \rightarrow \delta} \times A^{\tau \rightarrow \delta}$ to $[A^\perp \Rightarrow A^\delta]$, whose value is $\bigvee \delta$ for all $f \in A^{\sigma \rightarrow \delta}$ and $g \in A^{\tau \rightarrow \delta}$.

In view of (1), from (3), we get
$$(\text{cinl}, \text{cinr}) \circ (\cdot, \cdot) \circ (\text{abst}^{\sigma,\delta} \times \text{abst}^{\tau,\delta}) = \lambda a \in A^\sigma. \bigvee \tau.$$ 

However, we have no left inverse to $\bigvee \delta$, and we don’t have an analogous identity for $\text{cinf}$. Extensional pre-applicative structures and $\beta \eta$-structures are defined just as in definition 3.2 and definition 3.3, and the same remarks apply. However, these remarks only apply for types different from $\perp$.

13 $\mathcal{P}$-Realizability for the Arrow, Product, Sum, and $\perp$ Types

In this section, we extend the semantic notion of realizability defined in section 8 to the calculus $\lambda^{\rightarrow,\times,+,\perp}$. Definition 4.1, definition 8.1, definition 8.2, definition 8.3, and definition 4.5, are unchanged. However, for the reader's convenience, we repeat definition 8.3 and definition 4.5.

**Definition 13.1** Let $(A, \mathcal{P}, \text{Cov})$ 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 $M \geq N$, then $\mathcal{P}(N, \sigma)$.

(P3)
(1) If $\text{Cov}_{\sigma \rightarrow \tau}(C, M), \mathcal{P}(N, \sigma)$, and $\mathcal{P}(M'N, \tau)$ whenever $M' \in C$, then $\mathcal{P}(MN, \tau)$.
(2) If $\text{Cov}_{\sigma \times \tau}(C, M)$, 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 conditions (P1)-(P3) of definition 13.1.

**Definition 13.2** Let $(A, \mathcal{P}, \text{Cov})$ be a $\mathcal{P}$-site. A function $S: A \rightarrow 2^T$ has the *sheaf property* (or is a $\mathcal{P}$-*sheaf*) iff it satisfies the following conditions:

(S1) If $\sigma \in S(M)$, then $M \in P_\sigma$.

(S2) If $\sigma \in S(M)$ and $M \succeq N$, then $\sigma \in S(N)$.

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

A function $S: A \rightarrow 2^T$ as in definition 13.2 can also be viewed as a family $S = (S_\sigma)_{\sigma \in T}$, where $S_\sigma = \{M \in A \mid \sigma \in S(M)\}$. Then, the sets $S_\sigma$ are called $\mathcal{P}$-*candidates*. The conditions of definition 13.2 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 $M \in S_\sigma$.

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, recall that (S3) and (P1) imply that $S_\sigma$ is nonempty and contains all stubborn elements in $P_\sigma$.

We now generalize the definition of realizers to take into accounts elements of the form $\nabla\sigma(M)$. We define $\mathcal{P}$-*realizability* as follows.

**Definition 13.3** Let $(A, \mathcal{P}, \text{Cov})$ be a $\mathcal{P}$-site. The sets $r[\sigma]$ of *realizers of $\sigma$* are defined as follows:

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

Lemma 8.5 still holds.

**Lemma 13.4** Given a scenic $\mathcal{P}$-site $(A, \mathcal{P}, \text{Cov})$, if $\mathcal{P}$ satisfies conditions (P1)-(P3), then the family $(r[\sigma])_{\sigma \in T}$ has the sheaf property, and each $r[\sigma]$ contains all stubborn elements in $P_\sigma$.

**Proof.** It only differs in an inessential way from the proof of definition 8.5. The differences have to do with the sum type.

(S2). The only new case is the sum type.
3. Sum type $\sigma + \tau$. Assume that $M \geq M'$ for $M \in r[\sigma + \tau]$. Since $M \in r[\sigma + \tau]$, we have
\[
\text{Cov}_{\sigma + \tau} \left( \{ \text{inl}(M_1) \mid M_1 \in r[\sigma] \text{ and } M \geq \text{inl}(M_1) \} \cup \\
\{ \text{inr}(M_2) \mid M_2 \in r[\tau] \text{ and } M \geq \text{inr}(M_2) \} \cup \\
\{ \triangledown_{\sigma + \tau}(M_3) \mid M_3 \in P_\bot \text{ and } M \geq \triangledown_{\sigma + \tau}(M_3) \} \right), M' \}.
\]
Consider the cover $D$ of $M$:
\[
D = \{ \text{inl}(M_1) \mid M_1 \in r[\sigma] \text{ and } M \geq \text{inl}(M_1) \} \cup \\
\{ \text{inr}(M_2) \mid M_2 \in r[\tau] \text{ and } M \geq \text{inr}(M_2) \} \cup \\
\{ \triangledown_{\sigma + \tau}(M_3) \mid M_3 \in P_\bot \text{ and } M \geq \triangledown_{\sigma + \tau}(M_3) \}.
\]
By property (3) of definition 8.1, 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 8.1, $M' \in D$, and it is clear that
\[
\{ Q \mid Q \in D, Q \preceq M' \} = \{ \text{inl}(M_1) \mid M_1 \in r[\sigma] \text{ and } M' \geq \text{inl}(M_1) \} \cup \\
\{ \text{inr}(M_2) \mid M_2 \in r[\tau] \text{ and } M' \geq \text{inr}(M_2) \} \cup \\
\{ \triangledown_{\sigma + \tau}(M_3) \mid M_3 \in P_\bot \text{ and } M' \geq \triangledown_{\sigma + \tau}(M_3) \}.
\]
Then, we have
\[
\text{Cov}_{\sigma + \tau} \left( \{ \text{inl}(M_1) \mid M_1 \in r[\sigma] \text{ and } M' \geq \text{inl}(M_1) \} \cup \\
\{ \text{inr}(M_2) \mid M_2 \in r[\tau] \text{ and } M' \geq \text{inr}(M_2) \} \cup \\
\{ \triangledown_{\sigma + \tau}(M_3) \mid M_3 \in P_\bot \text{ and } M' \geq \triangledown_{\sigma + \tau}(M_3) \} \right), M' \}.
\]
showing that $M' \in r[\sigma + \tau]$.

(S3). Let $M$ be simple. The only new case is the sum type.

3. Sum type $\sigma + \tau$. Assume that $\text{Cov}_{\sigma + \tau}(C, M)$ and that $N \in r[\sigma + \tau]$ for every $N \in C$. Let
\[
D = \{ \text{inl}(M_1) \mid M_1 \in r[\sigma] \text{ and } M \geq \text{inl}(M_1) \} \cup \\
\{ \text{inr}(M_2) \mid M_2 \in r[\tau] \text{ and } M \geq \text{inr}(M_2) \} \cup \\
\{ \triangledown_{\sigma + \tau}(M_3) \mid M_3 \in P_\bot \text{ and } M \geq \triangledown_{\sigma + \tau}(M_3) \}.
\]
Using the properties of $\preceq$, it is clear that $D$ is a sieve on $M$. We need to prove that $\text{Cov}_{\sigma + \tau}(D, M)$, since this is equivalent to $M \in r[\sigma + \tau]$. Let $N \in C$, and consider the set $\{ Q \mid Q \in D, Q \preceq N \}$. We prove that $\text{Cov}(\{ Q \mid Q \in D, Q \preceq N \}, N)$. However, since $N \in C$ and by assumption, $N \in r[\sigma + \tau]$ for every $N \in C$, we have
\[
\text{Cov}_{\sigma + \tau} \left( \{ \text{inl}(M_1) \mid M_1 \in r[\sigma] \text{ and } N \geq \text{inl}(M_1) \} \cup \\
\{ \text{inr}(M_2) \mid M_2 \in r[\tau] \text{ and } N \geq \text{inr}(M_2) \} \cup \\
\{ \triangledown_{\sigma + \tau}(M_3) \mid M_3 \in P_\bot \text{ and } N \geq \triangledown_{\sigma + \tau}(M_3) \} \right), N \}.
\]
Since $N \preceq M$, it is clear that
\[
\{ Q \mid Q \in D, Q \preceq N \} = \{ \text{inl}(M_1) \mid M_1 \in r[\sigma] \text{ and } N \geq \text{inl}(M_1) \} \cup \\
\{ \text{inr}(M_2) \mid M_2 \in r[\tau] \text{ and } N \geq \text{inr}(M_2) \} \cup \\
\{ \triangledown_{\sigma + \tau}(M_3) \mid M_3 \in P_\bot \text{ and } N \geq \triangledown_{\sigma + \tau}(M_3) \}.
\]
Then, by property (4) of definition 8.1, we have $\text{Cov}_{\sigma + \tau}(D, M)$, that is, $M \in r[\sigma + \tau]$. □

We also need to extend definition 9.1 to give an interpretation to terms of the form $\triangledown_{\sigma}(M)$.
14 Interpreting $\lambda$-Terms in $\lambda^{-,\times,+,\bot}$

We extend definition 9.1 to take care of $\bot$.

**Definition 14.1** Given a pre-applicative structure $A$, a valuation, or environment, is any function $p : X \to A$, such that $\rho(x) \in A^\sigma$ if $x : \sigma$. A meaning function for $A$ is a partial function $A[-](-)$ from pairs of ($\alpha$-equivalence classes of) terms and valuations to $A$, such that $A[M]p$ is defined whenever $M : \sigma$, in which case $A[M]p \in A^\sigma$. In addition, a meaning function satisfies the following conditions:

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

where $f$ is the function defined such that, $f(a) = A[M]p[x = a]$, for every $a \in A^\sigma$

- $A[\pi_1(M)]p = \pi_1(A[M]p)$
- $A[\pi_2(M)]p = \pi_2(A[M]p)$
- $A[(M_1, M_2)]p = (A[M_1]p, A[M_2]p)$
- $A[\text{inl}(M)]p = \text{inl}(A[M]p)$
- $A[\text{inr}(M)]p = \text{inr}(A[M]p)$
- $A[\nabla_\sigma(M)]p = \nabla_\sigma(A[M]p)$.

It is routine to show that the following property holds:

- $A[M]p_1 = A[M]p_2$, whenever $p_1(x) = p_2(x)$ for every $x \in FV(M)$ (independence)

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

15 The Realizability Theorem For $\lambda^{-,\times,+,\bot}$

In this section, we generalize the realizability lemma (lemma 10.6) and its main corollary (theorem 10.7) to the calculus $\lambda^{-,\times,+,\bot}$. In order to do so, we need some modify definition 10.1 to take care of elements of the form $\nabla_\sigma(M)$.

**Definition 15.1** We say that a site $(A, P, \text{Cov})$ is well-behaved iff the following conditions hold:

1. For any $a \in A^\sigma$, any $\varphi \in [A^\sigma \to A^\sigma]$, if $\text{abst}(\varphi)$ exists, $\text{Cov}(C, \text{app}(\text{abst}(\varphi), a))$, and $C$ is a nontrivial cover, then $c \preceq \varphi(a)$ for every $c \in C$;
   For any $a \in A^\bot$, any $b \in A^\sigma$, if $\text{Cov}(C, \text{app}(\nabla_\sigma^{-\to}(a), b))$ and $C$ is a nontrivial cover, then $c \preceq \nabla_\sigma(a)$ for every $c \in C$;
If \( \text{Cov}(C, \pi((a_1, a_2))) \) and \( C \) is a nontrivial cover, then \( c \preceq a_1 \) for every \( c \in C \).
If \( \text{Cov}(C, \pi_2((a_1, a_2))) \) and \( C \) is a nontrivial cover, then \( c \preceq a_2 \) for every \( c \in C \).
If \( \text{Cov}(C, \pi_3((\sigma \times \tau)(a))) \) and \( C \) is a nontrivial cover, then \( c \preceq \sigma(a) \) for every \( c \in C \).
If \( \text{Cov}(C, \pi_2((\sigma \times \tau)(a))) \) and \( C \) is a nontrivial cover, then \( c \preceq \sigma(a) \) for every \( c \in C \).

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 \( f \in A^{\sigma-\delta} \) and \( g \in A^{\tau-\delta} \).

In view of definition 14.1, definition 15.1 implies the following conditions.

**Definition 15.2**

1. For any \( a \in A^{\sigma} \), if \( \text{Cov}_{\sigma}(C, \text{app}(\lambda x: \sigma. M \rho, a)) \) and \( C \) is a nontrivial cover, then \( c \preceq A[M] \rho[x := a] \) for every \( c \in C \).
2. For any \( b \in A^{\sigma} \), if \( \text{Cov}_{\sigma}(C, \text{app}(\nabla_{\sigma-\tau}(M) \rho, b)) \) and \( C \) is a nontrivial cover, then \( c \preceq A[\nabla_{\tau}(M)] \rho \) for every \( c \in C \);
3. If \( \text{Cov}_{\sigma+\tau}(C, \pi_1((M_1, M_2)) \rho) \) and \( C \) is a nontrivial cover, then \( c \preceq A[M_1] \rho \) for every \( c \in C \).
4. If \( \text{Cov}_{\sigma+\tau}(C, \pi_2((M_1, M_2)) \rho) \) and \( C \) is a nontrivial cover, then \( c \preceq A[M_2] \rho \) for every \( c \in C \).
5. If \( \text{Cov}_{\sigma+\tau}(C, \pi_3((\sigma \times \tau)(M)) \rho) \) and \( C \) is a nontrivial cover, then \( c \preceq A[\nabla_{\sigma}(M)] \rho \) for every \( c \in C \).
6. If \( \text{Cov}_{\sigma+\tau}(C, \pi_3((\sigma \times \tau)(M)) \rho) \) and \( C \) is a nontrivial cover, then \( c \preceq A[\nabla_{\tau}(M)] \rho \) for every \( c \in C \).
7. If \( \text{Cov}(p) = \text{triv}(p) \), then \( \text{Cov}(\text{app}(A[[M, N]] \rho, p)) = \text{triv}(\text{app}(A[[M, N]] \rho, p)) \), and if \( \text{Cov}_{\sigma+\tau}(C, p) \), \( \text{Cov}_{\delta}(D, \text{app}(A[[M, N]] \rho, p)) \), and \( C \) and \( D \) are nontrivial, then for every \( d \in D \), either there is some \( \text{inl}(p_1) \in C \) such that \( d \preceq \text{app}(A[M] \rho, p_1) \), or there is some \( \text{inr}(p_2) \in C \) such that \( d \preceq \nabla_{\sigma+\tau}(p_3) \), 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 10.3 to take care of the type \( \bot \).

**Definition 15.3** Given a well-behaved site \( (A, P, \text{Cov}) \), 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 \( (a_1, a_2) \in P_{\sigma \times \tau} \).
3. If \( a \in P_{\sigma} \), then \( \text{inl}(a) \in P_{\sigma \rightarrow \tau} \), 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 \( a \in P_{\bot} \), then \( \nabla_{\sigma}(a) \in P_{\sigma} \).
It is easy to verify that \( \text{app}(f, g, p) \in P_5 \) is stubborn if \( p \in P_{\sigma + \tau} \) is stubborn, \( f \in P_{\sigma - \delta} \), and \( g \in P_{\tau - \delta} \). This follows from condition (3) of definition 15.1.

In view of definition 14.1, definition 15.3 implies the following conditions.

**Definition 15.4**

(P4)

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

(P5)

1. If \( a \in P_{\sigma} \) and \( A[M]p[x := a] \in P_{\tau} \), then \( \text{app}(A[\lambda x : \sigma. M]p, a) \in P_{\tau} \).
2. If \( A[M]p \in P_{\sigma} \) and \( A[N]p \in P_{\tau} \), then \( \pi_1(A[(M, N)]p) \in P_{\sigma} \) and \( \pi_2(A[(M, N)]p) \in P_{\tau} \).
3. If \( \text{cov}_{\sigma + \tau}(C, p) \), \( A[M]p \in P_{\sigma - \delta} \), \( A[N]p \in P_{\tau - \delta} \), \( \text{app}(A[M]p, p_1) \in P_5 \) whenever \( \text{inl}(p_1) \in C \), and \( \text{app}(A[N]p, p_2) \in P_5 \) whenever \( \text{inr}(p_2) \in C \), and \( p_3 \in P_\tau \) whenever \( \nabla_{\sigma + \tau}(p_3) \in C \), then \( \text{app}(A[(M, N)]p, p) \in P_5 \).
4. If \( A[M]p \in P_{\tau} \) and \( b \in P_{\sigma} \), then \( \text{app}(A[\nabla_{\sigma}(M)]p, b) \in P_{\tau} \).
5. If \( A[M]p \in P_{\tau} \), then \( \pi_1(A[\nabla_{\sigma + \tau}(M)]p) \in P_{\sigma} \) and \( \pi_2(A[\nabla_{\sigma + \tau}(M)]p) \in P_{\tau} \).

We have the following generalization of lemma 10.5.

**Lemma 15.5** Given a well-behaved scenic site \( A, \mathcal{P}, \text{cov} \), and a family \( \mathcal{P} \) satisfying conditions (P1)-(P5), for every \( \rho \), 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 \( A[M]p[x := a] = \mathbf{r}[\tau] \), then \( A[\lambda x : \sigma. M]\rho \in \mathbf{r}[\sigma \rightarrow \tau] \). (2) If \( A[M]p \in \mathbf{r}[\sigma] \) and \( A[N]p \in \mathbf{r}[\tau] \), then \( A[(M, N)]p \in \mathbf{r}[\sigma \times \tau] \); (3) If \( A[M]p \in \mathbf{r}[\sigma \rightarrow \delta] \), and \( A[N]p \in \mathbf{r}[\tau \rightarrow \delta] \), then \( A[(M, N)]p \in \mathbf{r}[(\sigma + \tau) \rightarrow \delta] \). (4) If \( a \in P_{\tau} \), then \( \nabla_{\sigma}(a) \in \mathbf{r}[\sigma] \) for every \( \sigma \).

**Proof.** It is identical to the proof of lemma 10.5, except for (3), and we also need to prove (4).

3. We need to prove that \( A[(M, N)]p \in P_{(\sigma + \tau) - \delta} \), and that \( \text{app}(A[(M, N)]p, p) \in \mathbf{r}[\delta] \), for every \( p \in \mathbf{r}[\sigma + \tau] \). The first part of the proof is identical to that of lemma 10.5.

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

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

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

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

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

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

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

is a nontrivial cover, and $\text{Cov}_{\sigma + \tau}(C, p)$. Since $C$ and $D$ are nontrivial, by condition (3) of definition 15.2, for every $d \in D$, either there is some $\text{inl}(p_1) \in C$ such that $d \leq \text{app}(A[M] \rho, p_1)$, or there is some $\text{inr}(p_2) \in C$ such that $d \leq \text{app}(A[N] \rho, p_2)$, or there is some $\nabla_{\sigma + \tau}(p_3) \in C$ such that $d \leq \nabla_{\delta}(p_3)$. The first two cases are treated just as in the proof of lemma 10.5. 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 $\nabla_{\delta}(p_3) \in r[\delta]$. Then, by (S2), in all cases we get $d \in r[\delta]$. Finally, by (S3), we have $\text{app}(A[[M, N]] \rho, p) \in r[\delta]$.

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

1. Arrow type $\sigma \rightarrow \tau$. We prove that $\text{app}(\nabla_{\sigma \rightarrow \tau}(a), b) \in r[\tau]$ for every $b \in r[\sigma]$. Since $a \in P_\perp$ and by (S1) $b \in P_\sigma$, by (P5)(4), we have $\text{app}(\nabla_{\sigma \rightarrow \tau}(a), b) \in P_\tau$. If $\tau$ is a base type, $r[\tau] = P_\tau$ and $\text{app}(\nabla_{\sigma \rightarrow \tau}(a), b) \in r[\tau]$. Otherwise, $\text{app}(\nabla_{\sigma \rightarrow \tau}(a), b) \in P_\tau$ is a simple term and we use (S3). The case where $\text{app}(\nabla_{\sigma \rightarrow \tau}(a), b)$ is stubborn is trivial. Otherwise, assume that $\text{Cov}_\tau(C, \text{app}(\nabla_{\sigma \rightarrow \tau}(a), b))$ for some nontrivial cover $C$. Then, by condition (1) of definition 15.1, $c \leq \nabla_{\tau}(a)$ for every $c \in C$; By the induction hypothesis, $\nabla_{\tau}(a) \in r[\tau]$, and by (S2), we have $c \in r[\tau]$. Thus, by (S3), we have $\text{app}(\nabla_{\sigma \rightarrow \tau}(a), b) \in r[\tau]$.

2. Product type $\sigma \times \tau$. We prove that $\pi_1(\nabla_{\sigma \times \tau}(a)) \in r[\sigma]$ and $\pi_2(\nabla_{\sigma \times \tau}(a)) \in r[\tau]$. 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 $\sigma$ is a base type,
then \( r[\sigma] = P_\sigma \) and \( \pi_1(\nabla_{\sigma \times \tau}(a)) \in r[\sigma] \). Similarly, if \( \tau \) is a base type, then \( r[\tau] = P_\tau \) and \( \pi_2(\nabla_{\sigma \times \tau}(a)) \in r[\tau] \).

If \( \sigma \) 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 \( \text{Cov}_\sigma(C, \pi_1(\nabla_{\sigma \times \tau}(a))) \) where \( C \) is a nontrivial cover. Then, by condition (2) of definition 15.1, \( c \leq \nabla_\sigma(a) \) for every \( c \in C \). Since by the induction hypothesis, \( \nabla_\sigma(a) \in r[\sigma] \), by (S2), we have \( c \in r[\sigma] \). By (S3), we have \( \pi_1(\nabla_{\sigma \times \tau}(a)) \in r[\sigma] \). A similar argument applies to \( \pi_2(\nabla_{\sigma \times \tau}(a)) \).

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

\[
D = \{ \text{inl}(p_1) \mid p_1 \in r[\sigma] \text{ and } \nabla_{\sigma + \tau}(a) \geq \text{inl}(p_1) \} \cup \\
\{ \text{inr}(p_2) \mid p_2 \in r[\tau] \text{ and } \nabla_{\sigma + \tau}(a) \geq \text{inr}(p_2) \} \cup \\
\{ \nabla_{\sigma + \tau}(p_3) \mid p_3 \in P_\bot \text{ and } \nabla_{\sigma + \tau}(a) \geq \nabla_{\sigma + \tau}(p_3) \}.
\]

By the properties of \( \leq \), it is easy to verify that \( D \) is indeed a sieve. We need to prove that \( \text{Cov}_{\sigma + \tau}(D, \nabla_{\sigma + \tau}(a)) \), since this is equivalent to \( \nabla_{\sigma + \tau}(a) \in r[\sigma + \tau] \). Now, since \( q \leq \nabla_{\sigma + \tau}(a) \) implies that \( q = \nabla_{\sigma + \tau}(a_1) \) for some \( a_1 \leq a \), and since \( a \in P_\bot \), by (P2) we have \( a_1 \in P_\bot \). Thus, it is clear that \( D = \{ q \mid q \leq \nabla_{\sigma + \tau}(a) \} \), which is a principal sieve. However, since \( \nabla_{\sigma + \tau}(a) \in P_{\sigma + \tau} \), by property (2) of definition 8.1, \( \nabla_{\sigma + \tau}(a) \in P_{\sigma + \tau} \) is covered by the principal sieve \( D \), and thus \( \text{Cov}_{\sigma + \tau}(D, \nabla_{\sigma + \tau}(a)) \). Therefore, we have \( \nabla_{\sigma + \tau}(a) \in r[\sigma + \tau] \).

Finally, we now prove the main realizability lemma for \( \lambda_{\sigma + \tau} \).

**Lemma 15.6** Given a well-behaved scenic site \( (A, P, \text{Cov}) \), if \( P \) is a family satisfying conditions (P1)-(P5), then for every term \( M \) of type \( \sigma \), for every valuation \( \rho \) such that \( \rho(y) \in r[\gamma] \) for every \( y : \gamma \in \text{FV}(M) \), we have \( A[M] \rho \in r[\sigma] \).

**Proof.** We proceed by induction on the structure of \( M \). Some of the cases have already been covered in the proof of lemma 10.6, but we also need to handle terms of the form \( \text{inl}(M_1) \).

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

\[
D = \{ \text{inl}(p_1) \mid p_1 \in r[\sigma] \text{ and } \nabla_{\sigma + \tau}(a) \geq \text{inl}(p_1) \} \cup \\
\{ \text{inr}(p_2) \mid p_2 \in r[\tau] \text{ and } \nabla_{\sigma + \tau}(a) \geq \text{inr}(p_2) \} \cup \\
\{ \nabla_{\sigma + \tau}(p_3) \mid p_3 \in P_\bot \text{ and } \nabla_{\sigma + \tau}(a) \geq \nabla_{\sigma + \tau}(p_3) \}.
\]

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

\[
D = \{ p : \text{inl}(A[M_1] \rho) \geq p \}.
\]

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

\[
D = \{ p : \text{inl}(A[M_1] \rho) \geq p \}.
\]

However, by property (2) of definition 8.1, since \( \text{inl}(A[M_1] \rho) \in P_{\sigma + \tau} \) and \( D \) is a principal cover, \( \text{Cov}_{\sigma + \tau}(D, \text{inl}(A[M_1] \rho)) \) holds. Since by definition 14.1, \( A[\text{inl}(M_1)] \rho = \text{inl}(A[M_1] \rho) \), we have \( A[\text{inl}(M_1)] \rho \in r[\sigma + \tau] \). The case where \( M = \text{inr}(M_1) \) is similar.
If \( M = \nabla_{\sigma}(M_1) \), then by the induction hypothesis, \( A[M_1] \rho \in r[\perp] = P_\perp \). By lemma 15.5 (4), we have \( \nabla_{\sigma}(A[M_1] \rho) \in r[\sigma] \). Since by definition 14.1, \( A[\nabla_{\sigma}(M_1)] \rho = \nabla_{\sigma}(A[M_1] \rho) \), we have \( A[\nabla_{\sigma}(M_1)] \rho \in r[\sigma] \). □

Theorem 10.7 is generalized to the calculus \( \lambda^{\rightarrow, x, +, \perp} \) as follows.

**Theorem 15.7** Given a well-behaved scenic site \( (A, P, \text{Cov}) \), if \( P \) is a family satisfying conditions (P1)-(P5), then for every closed term \( M \) of type \( \sigma \), we have \( A[M] \in P_\sigma \). (in other words, the realizer \( A[M] \) satisfies the unary predicate defined by \( P \), i.e., every provable type is realizable).

**Proof.** Apply lemma 15.6 to the closed term \( M \) of type \( \sigma \) and to any arbitrary valuation \( \rho \). □

16 Applications to the System \( \lambda^{\rightarrow, x, +, \perp} \)

This section shows that theorem 15.7 can be used to prove a general theorem about terms of the system \( \lambda^{\rightarrow, x, +, \perp} \). As a corollary, it can be shown that all terms of \( \lambda^{\rightarrow, x, +, \perp} \) are strongly normalizing and confluent.

In order to apply theorem 15.7, we define a notion of cover for the site \( A \) whose underlying pre-applicative structure is the structure \( LT_\beta \).

**Definition 16.1** An \( I \)-term is a term of the form either \( \lambda x : \sigma. M, (M, N), \text{inl}(M), \text{inr}(M), \) \( [M, N] \), 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 \( LT_\beta \) as follows. Let \( P \) be a (unary) property of typed \( \lambda \)-terms.

**Definition 16.2** The cover algebra \( \text{Cov} \) 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
   \[
   \text{Cov}(M) = \{ N \mid M \xrightarrow{\beta} N, \{ N \mid M \xrightarrow{\beta} Q \xrightarrow{\beta} N, \text{for some } I \text{-term } Q \} \}.
   \]

Recall from definition 8.2 that \( M \) is simple iff it has at least two distinct covers. Thus, definition 16.2 implies that a term is simple in the sense of definition 16.1 iff it is simple in the sense of definition 8.2. Similarly a term is stubborn in the sense of definition 16.1 iff it is stubborn in the sense of definition 8.2. Also, definition 16.1 implies that \( LT_\beta \) is scenic.

Properties (P1-P3) are listed below.
Definition 16.3 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 \( \sigma \).

(P2) If \( M \in P_\sigma \) and \( M \xrightarrow{\beta} N \), then \( N \in P_\sigma \).

(P3) If \( M \) is simple, then:

1. If \( M \in P_\sigma \), \( N \in P_\sigma \), \( (\lambda x: \sigma. M')N \in P_\tau \) 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 \( M \in P_{\sigma \times \tau} \), \( \pi_1((M', N')) \in P_\sigma \) and \( \pi_2((M', N')) \in P_\tau \) whenever \( M \xrightarrow{\beta} (M', N') \), 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 16.3 are not simply a reformulation of condition (P3) of definition 13.1. This is because according to definition 16.2, 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 16.3 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 \( M \in P_\sigma \) 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 \( M \in P_\sigma \) is a stubborn term and \( N \in P_\sigma \) is any term, then \( MN \) is a stubborn term in \( P_\tau \). 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 \( P_\sigma \) and \( \pi_2(M) \) is a stubborn term in \( P_\tau \).

Properties (P4-P5) are listed below.

Definition 16.4 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 \( (M, N) \in P_{\sigma \times \tau} \).
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 \( \pi_1((M, N)) \in P_\sigma \) and \( \pi_2((M, N)) \in P_\tau \).
3. If \( P \in P_{\sigma + \tau} \), \( M \in P_{\sigma \rightarrow \delta} \), \( N \in P_{\tau \rightarrow \delta} \), \( MP_1 \in P \) whenever \( P \xrightarrow{\beta} \text{inl}(P_1) \), \( NP_2 \in P \) whenever \( P \xrightarrow{\beta} \text{inr}(P_2) \), and \( P_1 \in P_\perp \) whenever \( P \xrightarrow{\beta} \nabla_{\sigma + \tau}(P_1) \), then \([M, N]P \in P\).
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 16.4 are not simply a reformulation of conditions (P5) of definition 15.4. 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 $[M, N]P \in P_5$ is a stubborn term in $P_5$, if $P \in P_{\sigma+\tau}$ is stubborn, $M \in P_{\sigma \rightarrow \delta}$, and $N \in P_{\rho \rightarrow \delta}$. Indeed, $[M, N]P \in P_5$ can only reduce to an I-term if $P$ does. We now show that the conditions of definition 8.1 and the conditions of definition 15.2 hold.

**Lemma 16.5** Definition 16.2 defines a cover algebra, and the site $(\mathcal{LT}_\beta, \mathcal{P}, \text{Cov})$ is scenic and well-behaved.

**Proof.** Conditions (0)-(4) of definition 8.1 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 $Q \not\preceq M'N$.

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 16.1, $M$ 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 16.1 implies that $M' \in C$.

(6) If $\text{Cov}(M) = \text{triv}(M)$, then $\text{Cov}(\pi_1(M)) = \text{triv}(\pi_1(M))$, $\text{Cov}(\pi_2(M)) = \text{triv}(\pi_2(M))$, and if $\text{Cov}(C, M)$ and $\text{Cov}(D, \pi_1(M))$ (resp. $\text{Cov}(D, \pi_2(M))$) with $C$ and $D$ nontrivial, then for every $Q \in D$, there is some $M' \in C$ such that $Q \not\preceq \pi_1(M')$ (resp. $Q \not\preceq \pi_2(M')$).

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 16.1, $M$, $\pi_1(M)$, 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 16.1 implies that $M' \in C$. The same argument applies to $\pi_2(M)$.

Let us now verify the conditions of definition 15.2. First, recall that for the structure $\mathcal{LT}_\beta$, for every valuation $\rho$ (an infinite substitution) $\mathcal{LT}_\beta[M]_{\rho} = M[\varphi]$, where $\varphi$ is the substitution defined by the restriction of $\rho$ to $FV(M)$. Also $\text{app}(M, N) = MN$, and recall that $A^\sigma$ is the set of terms of type $\sigma$.

(1) For any $a \in A^\sigma$, if $\text{Cov}_\tau(C, \text{app}(\mathcal{LT}_\beta[\lambda x: \sigma. M]_{\rho}, a))$ and $C$ is a nontrivial cover, then $c \not\preceq \mathcal{LT}_\beta[M]_{\rho}[x: = a]$ for every $c \in C$.

For any $b \in A^\sigma$, if $\text{Cov}_\tau(C, \text{app}(\mathcal{LT}_\beta[\bigvee_{\sigma \rightarrow \tau}(M)]_{\rho}, b))$ and $C$ is a nontrivial cover, then $c \not\preceq \mathcal{LT}_\beta[\bigvee_{\tau}(M)]_{\rho}$ for every $c \in C$.
We have \( \text{app}(LT_\beta [\lambda x: \sigma. M]\rho, a) = ((\lambda x: \sigma. M)[\varphi])a \), where \( \varphi \) is the substitution defined by the restriction of \( \rho \) to \( FV(M) - \{x\} \). By definition 16.1, 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

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

However, we have \( (M[\varphi])[a/x] = M[\varphi[x := a]] \) (using a suitable renaming of \( x \)). By the definition of \( LT_\beta [\sigma]\), we have \( LT_\beta [\sigma]\rho[x := a] = M[\varphi[x := a]] \), and this part of the proof is complete. The proof for \( \nabla_{\sigma \rightarrow \tau} (M) \) is completely analogous.

(2) If \( \text{Cov}_{\sigma}((C, \pi_1(LT_\beta ([M_1, M_2])\rho)) \) and \( C \) is a nontrivial cover, then \( c \subseteq LT_\beta [M_1]\rho \) for every \( c \in C \).

If \( \text{Cov}_{\sigma}((C, \pi_2(LT_\beta ([M_1, M_2])\rho)) \) and \( C \) is a nontrivial cover, then \( c \subseteq LT_\beta [M_2]\rho \) for every \( c \in C \).

If \( \text{Cov}_{\sigma}((C, \pi_1(LT_\beta [\nabla_{\sigma \rightarrow \tau} (M)]\rho)) \) and \( C \) is a nontrivial cover, then \( c \subseteq LT_\beta [\nabla_{\sigma \rightarrow \tau} (M)]\rho \) for every \( c \in C \).

If \( \text{Cov}_{\sigma}((C, \pi_2(LT_\beta [\nabla_{\sigma \rightarrow \tau} (M)]\rho)) \) and \( C \) is a nontrivial cover, then \( c \subseteq LT_\beta [\nabla_{\sigma \rightarrow \tau} (M)]\rho \) for every \( c \in C \).

We have \( LT_\beta [([M_1, M_2])\rho = ([M_1, M_2][\varphi], \) where \( \varphi \) is the substitution defined by the restriction of \( \rho \) to \( FV(M_1) \cup FV(M_2) \). By definition 16.1, since \( C \) is nontrivial, \( c \in C \) means that

\[
\pi_1([M_1, M_2][\varphi]) \xrightarrow{\beta} Q \xrightarrow{\beta} c,
\]

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

\[
\pi_1([M_1, M_2][\varphi]) \xrightarrow{\beta} M_1[\varphi] \xrightarrow{\beta} c.
\]

Since \( LT_\beta [M_1]\rho = M_1[\varphi] \), this part of the proof is complete. The other cases are entirely analogous.

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

The first part says that \( [M[\varphi], N[\varphi]]P \) is stubborn if \( P \) is stubborn, which has already been shown (where \( \varphi \) is the substitution defined by the restriction of \( \rho \) to \( FV(M) \cup FV(N) \)). By definition 16.1, 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

\[
P \xrightarrow{\beta} \text{inl}(P_1),
\]

and

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

42
or \( \pi \xrightarrow{\beta} \text{inr}(P_2) \), and

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

or \( \pi \xrightarrow{\beta} \nabla_{\sigma+\tau}(P_3) \), and

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

In each case, since \( C \) is nontrivial, by definition 16.1, we have \( \text{inl}(P_1) \in C \), \( \text{inl}(P_2) \in C \), and \( \nabla_{\sigma+\tau}(P_3) \in C \). □

Thus, the site \( \langle L_T, P, \text{Cov} \rangle \) is scenic and well-behaved. Consequently, we can apply theorem 15.7, and get a general theorem for proving properties of terms of the system \( \lambda \rightarrow, \times, +, \bot \). In fact, for the structure \( L_T \), for a property \( P \) satisfying conditions \( (P1)-(P5) \), by \( (P1) \) and \( (P3) \), every variable \( x \) of type \( \sigma \) is stubborn (for every \( \sigma \)). Thus, we can apply lemma 15.6 with the valuation \( \rho \) such that \( \rho(x) = x \) for every variable \( x \), since by lemma 13.4, \( r[\sigma] \) contains every stubborn term. Consequently, we have the following theorem (compare with theorem A of the introduction).

**Theorem 16.6** If \( P \) is a family of \( \lambda \)-terms satisfying conditions \( (P1)-(P5) \), then \( P_\sigma = \Lambda_\sigma \) for every type \( \sigma \) (in other words, every term satisfies the unary predicate defined by \( P \)).

**Proof.** By lemma 16.5, the site \( \langle L_T, P, \text{Cov} \rangle \) is scenic and well-behaved. By the discussion just before stating theorem 16.6, the identity valuation \( \rho \) such that \( \rho(x) = x \) for every variable \( x \), is such that \( \rho(x) \in r[\sigma] \) for every \( x: \sigma \). Thus, we can apply lemma 15.6 to any term \( M \) of type \( \sigma \) and to \( \rho \), and we have \( L_T[\beta[M] \rho \in r[\sigma] \). However, in the present case, \( L_T[\beta[M] \rho = M \). Thus, \( M \in r[\sigma] \), and since \( r[\sigma] \subseteq P_\sigma \), we have \( M \in P_\sigma \), as claimed. □

As a corollary, strong normalization and confluence can be shown, see Gallier [2] for such a treatment. In part II of this paper, we show how the realizability theorem can be extended to the second-order (polymorphic) \( \lambda \)-calculus \( \lambda \rightarrow, \forall \).

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

**References**


