Infinite Intersection and Union Types for the Lazy Lambda Calculus

Marcello M. Bonsangue and Joost N. Kok

Leiden Institute of Advanced Computer Science
Leiden University, Leiden, The Netherlands
marcello@liacs.nl and joost@liacs.nl

Abstract. A type theory with *infinitary* intersection and union types for the lazy λ-calculus is introduced. Types are viewed as upper closed subsets of a Scott domain. Intersection and union type constructors are interpreted as the set-theoretic intersection and union, respectively, even when they are not finite. The assignment of types to λ-terms extends naturally the basic type assignment system. We prove soundness and completeness using a generalization of Abramsky's finitary domain logic for applicative transition systems.

1 Introduction

Intersection types were introduced in [10] as an extension of Curry's simply typed λ-calculus. A remarkable property of intersection types is their ability to express computational information of λ-terms, such as termination properties (e.g. strong normalization [16] and weak head normalization [1]) and reduction properties (e.g. reduction to a closed term [11]). In [6], an intersection type system was proved to be sound and complete with respect to Scott's set-theoretic semantics for simple types.

Since then, several variations of the original intersection type discipline have been explored, including enrichment with new type constructors like union of types [15, 3, 4], conservative extensions like infinite intersection types [14, 3], and modifications inducing more computationally adequate models, like the lazy λ-calculus [1].

Leivant showed in [14] that considering infinite intersection types has several advantages over the ordinary intersection type discipline. Firstly, infinite intersection types form a framework in which several typing mechanisms, such as parametric polymorphism, stratified parametric polymorphism, and recursive types can be interpreted. Secondly, there is a relationship between the size of the types allowed and the computational nature of the functions representable by finite λ-terms. And finally, infinite intersection types allow for typing in a uniform way a number of interesting classes of λ-terms, like those representing the Berarducci numerals (a result independently obtained also in [3]).

Although infinite intersection types conceptually express better the idea of multiple typing, not much work has been done in generalizing the filter model of [6] towards a complete set-theoretical model for infinite intersection types. The reason is maybe due to the fact that a study of infinite intersection types cannot be carried out easily without considering infinite union types. As a consequence, a set theoretic-model has to be a completely distributive lattice, in contrast to the fact that canonical models for λ-calculi, like some Scott domains, induce topologies that do not satisfy the complete distributivity laws.

In this paper we solve this open problem by proving soundness and completeness of a type system for the lazy λ-calculus, in which arbitrary intersection and union types are allowed. Types are interpreted as upper closed subsets of a Scott domain, the so-called saturated sets. The meaning of the intersection and union type constructors is the set-theoretic intersection and union, respectively, even when they are not finite. Types are assigned to λ-terms in a such way that the interpretation of a λ-term belongs to the interpretation of all types it is assigned to. The completeness result for our infinitary type system is based on the connection between Abramsky's domain logic for applicative transition systems and the Scott compact opens of a suitable domain [1].

We have chosen to focus on the lazy λ-calculus, but similar results hold also for the λ-calculus. The reason of our choice is simplicity. In fact, to obtain a non-trivial model for the λ-calculus one needs to consider a fixed collection of basic types. To fall into the scope of the general theory developed by Abramsky, a logical presentation of the associated Scott domains from scratch would use lifting to get off the ground. However the resulting type expression would be rather complicated.

This work fits into our studies towards an infinitary logic of domains. In [8, 9] we focussed on an infinitary logical form for Plotkin's powerdomain construction, while in this paper we concentrate on the function space construction. The key ingredients are a Stone-like duality for $\mathcal{T}_0$ topological spaces [7] and a characterization of sober spaces in terms of their completely distributive lattice of saturated sets (i.e. upper closed sets with respect to the specialization preorder induced by the opens) [9]. These results allow us to freely extend Abramsky's finitary logic of compact opens [2] to the infinitary logic of the saturated sets.

The paper is organized as follows. In the next section we introduce the lazy λ-calculus, its syntax, and its domain theoretical model. In Section 3 we present our infinitary type theory and prove soundness and completeness. Then, in Section 4 we give similar results for a type assignment system. We conclude in Section 5 with some final remarks.

Acknowledgments We are thankful to M. Dezani-Ciancaglini, M. Coppo and J. Rutten for pointing out infinite intersection types as a possible application of our results towards an infinitary logic of domains [9]. We are also grateful to S. Abramsky and D. Scott for discussions on this topic.

2 The Lazy Lambda Calculus

In this section we recall some basic definitions of the lazy λ-calculus. Its syntax coincides with that of the λ-calculus.

Definition 1. *The set Λ of λ-term is defined by the following grammar*

$$M ::= x \mid \lambda x.M \mid MM,$$

where x is a variable from a fixed set $\mathrm{V}ar$.

The definitions of free variables, substitution and closed terms are standard [5]. As usual, the set of closed λ-terms is denoted by Λ^o.

The operational model of the lazy λ-calculus is described by a convergence relation.

Definition 2. *The convergence to principal weak head normal form $\Downarrow$ on closed λ-terms is defined inductively by*

1. $\lambda x.M \Downarrow \lambda x.M$,
2. *if* $M \Downarrow \lambda x.P$ *and* $P[N/x] \Downarrow Q$ *then* $MN \Downarrow Q$.

Informally, $M \Downarrow \lambda x.P$ means that given a term M, the environment sees whether it converges to some function $\lambda x.P$. If this is the case, then it can continue by providing another term N and applying it to $\lambda x.P$. As for ordinary transition systems in concurrency theory, the above convergence relation can be used to identify λ-terms with the same observable behavior. A natural observational equivalence is applicative bisimulation.

Definition 3. *A binary relation R on closed λ-terms is called an* applicative bisimulation *whenever, if $\langle M, N\rangle \in R$ then*

- $M \Downarrow \lambda x.M_1$ *implies there exists* N_1 *such that* $N \Downarrow \lambda x.N_1$ *and* $\langle M_1[P/x], N_1[P/x]\rangle \in R$ *for all closed λ-terms P.*

We write $M \lesssim^B N$ if there exists an applicative bisimulation R with $\langle M, N\rangle \in R$.

Applicative bisimulation as an observational equivalence for the lazy λ-calculus coincides with the contextual congruence $\lesssim^C$ defined on closed λ-terms by $M \lesssim^C N$ if and only if for every closed context $C[\cdot]$, whenever $C[M] \Downarrow M_1$ for some closed term M_1, also $C[N] \Downarrow N_1$ for some closed term N_1 [1].

Abramsky [1] gives a domain theoretical model of the lazy λ-calculus. Let **SDom** be the category with Scott domains as objects and Scott continuous function as morphisms. Define $\mathcal{D}$ to be the Scott domain obtained as the canonical solution in the category **SDom** of the recursive domain equation

$$X \cong (X \to X)_\bot$$

where $(-)_\bot$ is the lifting construction, and $- \to -$ is the set of all Scott continuous functions endowed with the pointwise order. Furthermore, let *unfold* be the isomorphism mapping $\mathcal{D} \to (\mathcal{D} \to \mathcal{D})_\bot$, and *fold* be its inverse. The meaning of a λ-term at an environment $\rho{:}\mathrm{V}ar \to \mathcal{D}$ is defined by induction on its structure as follows:

$$\begin{aligned}
[\![x]\!]_\rho &= \rho(x)\\
[\![MN]\!]_\rho &= \mathit{lift}(Ap)(\mathit{unfold}([\![M]\!]_\rho))([\![N]\!]_\rho)\\
[\![\lambda x.M]\!]_\rho &= \mathit{fold}(\mathit{up}(\lambda d \in \mathcal{D}.[\![M]\!]_{\rho[d/x]}))
\end{aligned}$$

where $Ap{:}(\mathcal{D} \to \mathcal{D}) \to \mathcal{D} \to \mathcal{D}$ is the standard (continuous) application function, $\mathit{lift}(Ap)$ is the lifting of Ap to the strict function space $(\mathcal{D} \to \mathcal{D})_\perp \to_\perp \mathcal{D} \to \mathcal{D}$, and $\mathit{up}(d)$ is the lifting of $d \in \mathcal{D}$ to the correspondent element in the domain $(\mathcal{D})_\perp$.

The above semantics when restricted to closed terms is fully abstract with respect to applicative bisimulation, meaning that for all closed terms M and N, $M \lesssim^B N$ if and only if $[\![M]\!] \sqsubseteq [\![N]\!]$, where $\sqsubseteq$ is the partial order of the domain $\mathcal{D}$ [1].

3 An Infinitary Type Theory

Types are syntactical entities built by means of the function type constructor '$\to$', lifting '$(\cdot)_\perp$', an infinitary intersection type constructor '$\bigwedge$' and an infinitary union type constructor '$\bigvee$'.

Definition 4. *The class* $(\phi \in)\mathbf{T}$ *is defined inductively by the following grammar:*

$$\phi ::= \bigvee_{i\in I} \phi_i \mid \bigwedge_{i\in I} \phi_i \mid (\psi)_\perp$$
$$\psi ::= \bigvee_{i\in I} \psi_i \mid \bigwedge_{i\in I} \psi_i \mid \phi_1 \to \phi_2,$$

where I is an arbitrary index set. If $I = \emptyset$ then we write tt for $\bigwedge_{i\in I}\phi_i$ and $\bigwedge_{i\in I}\psi_i$, and we write ff for $\bigvee_{i\in I}\phi_i$ and $\bigvee_{i\in I}\psi_i$.

We denote by $\mathbf{T}_\omega$ the sub-language of $\mathbf{T}$ obtained by restricting to finite intersection type and finite union type constructors. Semantics of types in $\mathbf{T}$ is given in terms of the Scott domain $\mathcal{D}$. It is defined inductively as follows:

$$\begin{array}{lcl}
d \models \bigvee_{i\in I}\phi_i & \iff & \exists i \in I.\ d \models \phi_i \\
d \models \bigwedge_{i\in I}\phi_i & \iff & \forall i \in I.\ d \models \phi_i \\
d \models (\psi)_\perp & \iff & \mathit{unfold}(d) = f \neq \perp \text{ and } f \models \psi
\end{array}$$

$$\begin{array}{lcl}
f \models \bigvee_{i\in I}\psi_i & \iff & \exists i \in I.\ f \models \psi_i \\
f \models \bigwedge_{i\in I}\psi_i & \iff & \forall i \in I.\ f \models \psi_i \\
f \models \phi_1 \to \phi_2 & \iff & \forall d \in \mathcal{D}.\ d \models \phi_1 \text{ implies } f(d) \models \phi_2.
\end{array}$$

As usual, for $d \in \mathcal{D}$, we write $\mathbf{T}(d)$ for the set $\{\phi \in \mathbf{T} \mid d \models \phi\}$. Next we present a formal proof system for deriving *assertions* over $\mathbf{T}$ of the form $\phi_1 \leq \phi_2$ and $\phi_1 = \phi_2$. Their semantics is defined by

$$\begin{array}{l}
\models \phi_1 \leq \phi_2 \iff \forall d \in \mathcal{D}.\ d \models \phi_1 \text{ implies } d \models \phi_2 \\
\models \phi_1 = \phi_2 \iff \models \phi_1 \leq \phi_2 \text{ and } \models \phi_2 \leq \phi_1\,.
\end{array}$$

Our goal is to define a complete theory in the sense that $\phi_1 \leq \phi_2$ is provable in the theory if and only $\models \phi_1 \leq \phi_2$.

For a complete axiomatization of the above assertions we will need an auxiliary *coprimeness predicate* $\mathsf{C}(\phi)$ on types ϕ of $\mathbf{T}$. Types for which the coprimeness predicate holds will play a role in validating the axiom about the distribution of union types over the $\to$ constructor when appearing at its right-hand side. Clearly such an axiom is not valid in general. The following table presents a formal system for the *coprimeness judgment*.

- $\mathsf{C}(tt)$

- $\dfrac{\mathsf{C}(\phi_1) \text{ and } \phi_1 = \phi_2}{\mathsf{C}(\phi_2)}$

- $\dfrac{\{\mathsf{C}(\phi_i)\}_{i\in I} \text{ and } \{\mathsf{C}(\phi'_i)\}_{i\in I} \quad (\forall J \subseteq I.\ \bigwedge_{j\in J} \phi'_j = \mathit{ff} \Longrightarrow \bigwedge_{j\in J} \phi_j = \mathit{ff})}{\mathsf{C}(\bigwedge_{i\in I}(\phi_i \to \phi'_i)_\perp)}\ I \text{ finite}$

The last rule bears great resemblance to the definition of the finite elements of the function space between two Scott domains, the so-called step functions [18]. Indeed, types for which the coprimeness predicate holds are precisely the upper closure of finite elements of the Scott domain $\mathcal{D}$ [2].

Note that $\mathsf{C}(\phi)$ implies that no disjunctions (up-to $=$) occur in ϕ, whereas the type theory Π given in [4] uses a predicate $\mathbf{P}(\phi)$ which essentially allows for disjunctions only if they occur on the left-hand side of the $\to$ constructors.

The following *logical axioms and rules* give to $\mathbf{T}$ the structure of a large completely distributive lattice.

- $\phi \leq \phi$
- $\dfrac{\phi_1 \leq \phi_2 \text{ and } \phi_2 \leq \phi_3}{\phi_1 \leq \phi_3}$
- $\dfrac{\phi_1 \leq \phi_2 \text{ and } \phi_2 \leq \phi_1}{\phi_1 = \phi_2}$
- $\dfrac{\phi_1 = \phi_2}{\phi_1 \leq \phi_2 \text{ and } \phi_2 \leq \phi_1}$
- $\dfrac{\{\phi \leq \phi_i\}_{i\in I}}{\phi \leq \bigwedge_{i\in I} \phi_i}$
- $\dfrac{k \in I}{\bigwedge_{i\in I} \phi_i \leq \phi_k}$
- $\dfrac{\{\phi_i \leq \phi\}_{i\in I}}{\bigvee_{i\in I} \phi_i \leq \phi}$
- $\dfrac{k \in I}{\phi_k \leq \bigvee_{i\in I} \phi_i}$
- $\bigwedge_{i\in I} \bigvee_{j\in J_i} \phi_{i,j} = \bigvee_{f\in \Phi(\{J_i \mid i\in I\})} \bigwedge_{i\in I} \phi_{i,f(i)}$

The last axiom is about complete distributivity (its dual can be derived [17]). We use $\Phi(\mathcal{S})$, for a set of sets $\mathcal{S}$, to denote the set of all functions choosing an element from their input, that is, all functions $f\colon \mathcal{S} \to \bigcup \mathcal{S}$ such that $f(S) \in S$ for all $S \in \mathcal{S}$. Because of the presence of arbitrary choice functions, proofs involving the axiom of complete distributivity require the axiom of choice.

Finally, we give *structural axioms and rules* that relate the atomic formulas to the logical structure.

- $\dfrac{\phi_2 \leq \phi_1 \text{ and } \phi_3 \leq \phi_4}{\phi_1 \rightarrow \phi_3 \leq \phi_2 \rightarrow \phi_4}$
- $\dfrac{\psi_1 \leq \psi_2}{(\psi_1)_\perp \leq (\psi_2)_\perp}$
- $\dfrac{I \neq \emptyset}{(\bigwedge_{i\in I} \psi_i)_\perp = \bigwedge_{i\in I} (\psi_i)_\perp}$
- $(\bigvee_{i\in I} \psi_i)_\perp = \bigvee_{i\in I} (\psi_i)_\perp$
- $\bigwedge_{i\in I}(\phi \rightarrow \phi_i) \leq \phi \rightarrow \bigwedge_{i\in I} \phi_i$
- $\bigwedge_{i\in I}(\phi_i \rightarrow \phi) \leq \bigvee_{i\in I} \phi_i \rightarrow \phi$
- $\dfrac{\mathsf{C}(\phi) \text{ and } I \text{ finite}}{\phi \rightarrow \bigvee_{i\in I} \phi_i \leq \bigvee_{i\in I}(\phi \rightarrow \phi_i)}$
- $\dfrac{\{\phi_i \in \mathbf{T}_\omega\}_{i\in I} \text{ and } \{\phi_j \in \mathbf{T}_\omega\}_{j\in J}}{\bigwedge_{i\in I} \phi_i \rightarrow \bigvee_{j\in J} \phi_j \leq \bigvee_{H\in Fin(I)} \bigvee_{K\in Fin(J)} (\bigwedge_{h\in H} \phi_h \rightarrow \bigvee_{k\in K} \phi_k)}$

In the last rule we use $Fin(S)$, for a set S, to denote the set of all finite subsets of S. As usual, we write $\vdash A$ if the assertion A of $\mathbf{T}$ is derivable from the above axioms and rules. If we restrict our proof system to the finitary language $\mathbf{T}_\omega$ then it coincides with Abramsky's domain logic $\mathcal{L}(\sigma)$, for the type expression $\sigma \equiv \mathbf{rec}t.(t \rightarrow t)_\perp$ [2]. The last axiom scheme is new, and makes sense only in a language with arbitrary disjunctions and conjunctions. It is a statement of continuity of the '$\rightarrow$' type constructor, both on its left- and right-hand side. Note that because '$\rightarrow$' is contravariant on its left argument, filtered intersections are transformed into directed unions.

The next proposition shows that coprimeness judgment is essentially defined for the finitary sub-language $\mathbf{T}_\omega$. Indeed, it extends only to those infinitary types in $\mathbf{T}$ which are provably equivalent to finitary types in $\mathbf{T}_\omega$ for which the predicate C holds.

Proposition 1. *For each infinitary type $\phi \in \mathbf{T}$, if $\mathsf{C}(\phi)$ holds then there exists a finitary type $\phi' \in \mathbf{T}_\omega$ such that $\vdash \phi = \phi'$.*

Proof. By induction on the lenght of the derivation of $\mathsf{C}(\phi)$. □

As an immediate consequence we have that the restriction of the index set I to be finite in the rule

$$\frac{\mathsf{C}(\phi) \text{ and } I \text{ finite}}{\phi \rightarrow \bigvee_{i\in I} \phi_i \leq \bigvee_{i\in I}(\phi \rightarrow \phi_i)}$$

can be omitted. To prove this, we can use the above propositon to substitute a finitary type $\phi' \in \mathbf{T}_\omega$ for the type ϕ; then apply the last structural rule for moving outside the $\rightarrow$ constructor the infinite unions at its right; distribute the remaining finite unions at the right of the $\rightarrow$ constructor using the above proof rule, and finally substituting again ϕ for ϕ'.

The last structural rule can also be understood as an expression of compactness and finite approximability (as it will become clear by reading the sketch of the proof following Theorem 1). Logically, it allows for rewriting an arbitrary infinitary type in $\mathbf{T}$ as intersections of unions of finitary types in $\mathbf{T}_\omega$. This fact will be essential in the proof of the completeness result.

Lemma 1. *(Normal form) For each infinitary type $\phi \in \mathbf{T}$ there exist finitary types $\phi_{i,j} \in \mathbf{T}_\omega$, $i \in I$ and $j \in J$, such that $\vdash\ \phi = \bigwedge_{i\in I} \bigvee_{j\in J} \phi_{i,j}$.*

Proof. The proof proceed by induction on the structure of ϕ. We only treat the case $\phi \ \equiv\ \phi^l \rightarrow \phi^r$. By induction hypothesis we have that, for some types $\phi^l_{i,j}$'s and $\phi^r_{h,k}$'s in $\mathbf{T}_\omega$, the assertions $\phi^l = \bigwedge_{i\in I} \bigvee_{j\in J} \phi^l_{i,j}$ and $\phi^r = \bigwedge_{h\in H} \bigvee_{k\in K} \phi^r_{h,k}$ are both derivable in $\mathbf{T}$. Thus we have

$$
\begin{aligned}
&\phi^l \rightarrow \phi^r\\
&= \textstyle\bigwedge_{i\in I} \bigvee_{j\in J} \phi^l_{i,j} \rightarrow \bigwedge_{h\in H} \bigvee_{k\in K} \phi^r_{h,k} \quad \text{[induction hypothesis]}\\
&= \textstyle\bigvee_{m\in M} \bigwedge_{i\in I} \phi^l_{i,m} \rightarrow \bigwedge_{h\in H} \bigvee_{k\in K} \phi^r_{h,k} \quad \text{[complete distribuitivity, for some } M]\\
&= \textstyle\bigwedge_{h\in H} (\bigvee_{m\in M} \bigwedge_{i\in I} \phi^l_{i,m} \rightarrow \bigvee_{k\in K} \phi^r_{h,k}) \quad [\bigwedge \text{ distributes over} \rightarrow \text{at the right}]\\
&= \textstyle\bigwedge_{h\in H} \bigwedge_{m\in M} (\bigwedge_{i\in I} \phi^l_{i,m} \rightarrow \bigvee_{k\in K} \phi^r_{h,k}) \quad [\bigvee \text{ distributes over} \rightarrow \text{at the left}]\\
&= \textstyle\bigwedge_{h\in H} \bigwedge_{m\in M} \bigvee_{I'\in Fin(I)} \bigvee_{K'\in Fin(K)} (\bigwedge_{i\in I'} \phi^l_{i,m} \rightarrow \bigvee_{k\in K'} \phi^r_{h,k}),
\end{aligned}
$$

where in the last equality we used the axiom about distribution of infinite intersections and infinite unions over $\rightarrow$. Clearly, $\bigwedge_{i\in\ I'} \phi^l_{m,i} \rightarrow \bigvee_{k\in K'} \phi^r_{h,k}$ is a type in $\mathbf{T}_\omega$. Therefore $\phi^l \rightarrow \phi^r$ is provably equivalent in $\mathbf{T}$ to the intersection of unions of types in $\mathbf{T}_\omega$. □

Next we come to the soundness and completeness of the above system. Both results rely on the duality between the Scott domain $\mathcal{D}$ and the Lindenbaum algebra of $\mathbf{T}_\omega$. The following proposition is due to Abramsky [1].

Proposition 2. *Let $\mathcal{KO}(\mathcal{D})$ be the distributive lattice of Scott compact open sets of $\mathcal{D}$ ordered by subset inclusion. There is a well-defined order isomorphism between $\mathcal{KO}(\mathcal{D})$ and the Lindenbaum algebra $(\mathbf{T}_\omega/ =, \leq / =)$ which restricts to an isomorphism between the coprime elements of $\mathcal{KO}(\mathcal{D})$ and (equivalence classes of) types ϕ in $\mathbf{T}_\omega$ for which $\mathsf{C}(\phi)$ is provable.*

In other words, we can identify the finitary types in $\mathbf{T}_\omega$ with Scott compact open subsets of $\mathcal{D}$. This fact is used in the following soundness theorem.

Theorem 1. *(Soundness) Let ϕ_1 and ϕ_2 be two infinitary types in $\mathbf{T}$. If $\vdash\ \phi_1 \ \leq\ \phi_2$ then $\models \phi_1 \leq \phi_2$.*

The proof of the above theorem follows easily from the soundness proof given in [2], except for the soundness of the last structural rule. To validate it, we proceed as follows. First we apply the Scott open filter theorem by Hofmann and Mislove [12] to prove that for all Scott continuous functions $f:\mathcal{D} \to \mathcal{D}$, filtered collections of Scott compact saturated subsets $\{k_i\}_I$ of $\mathcal{D}$, and Scott open subsets u of $\mathcal{D}$, if $f(\bigcap_{i\in I} k_i) \subseteq u$ then there exists $i \in I$ such that $f(k_i) \subseteq u$. Secondly, we use compactness to prove that for all compact saturated subsets k of $\mathcal{D}$ and directed collection of open subsets $\{u_i\}_I$ of $\mathcal{D}$, if $f(k) \subseteq \bigcup_{i\in I} u_i$ then there exists $i \in I$ such that $f(k) \subseteq u_i$.

For proving the completeness of the infinitary type system $\mathbf{T}$ we need the following characterization of sober topological spaces [8, 9].

Proposition 3. *A T_0-space X is sober if and only if the completely distributive lattice of saturated sets $\mathcal{Q}(X)$ (ordered by subset inclusion) is free over the frame of open sets $\mathcal{O}(X)$ (ordered by subset inclusion).*

The collection of Scott compact open subsets of a Scott domain X is closed under finite intersection and it forms a basis for its Scott topology. More abstractly, the frame of Scott open sets $\mathcal{O}(X)$ (ordered by subset inclusion) is free over the distributive lattice of Scott compact opens $\mathcal{KO}(X)$ (ordered by subset inclusion) [13]. Applying this result to the Scott domain $\mathcal{D}$ and combining it with the previous proposition (since a Scott domain together with its Scott topology forms a sober space) we obtain that the completely distributive lattice of saturated sets $\mathcal{Q}(\mathcal{D})$ is free over the distributive lattice of Scott compact opens $\mathcal{KO}(\mathcal{D})$.

Using the normal form Lemma 1 we obtain a more syntactic description of the free completely distributive lattice over $\mathcal{KO}(\mathcal{D})$.

Lemma 2. *The Lindenbaum algebra $(\mathbf{T}/=, \leq/=)$ forms a completely distributive lattice which is free over the distributive lattice $(\mathbf{T}_\omega/=, \leq/=)$.*

Combining the above lemma with Proposition 3, and using the fact that free constructions are unique up-to isomorphism, we can now lift the isomorphism of Proposition 2 to an isomorphism between infinitary types and saturated sets of the Scott topology of $\mathcal{D}$.

Theorem 2. *The order isomorphism of Proposition 2 extends uniquely to a well-defined order isomorphism between the completely distributive lattice $\mathcal{Q}(\mathcal{D})$ of saturated sets of the Scott topology of $\mathcal{D}$ ordered by subset inclusion and the Lindenbaum algebra $(\mathbf{T}/=, \leq/=)$.*

Strong completeness of $\mathbf{T}$ now follows easily.

Corollary 1. *(Completeness) For ϕ_1 and ϕ_2 in $\mathbf{T}$, we have $\models \phi_1 \leq \phi_2$ if and only if $\vdash \phi_1 \leq \phi_2$.*

Another consequence of the above order isomorphism is that the order of $\mathcal{D}$ coincides with the preorder induced by $\mathbf{T}$ on $\mathcal{D}$: For every $d_1, d_2 \in \mathcal{D}$, $d_1 \sqsubseteq d_2$ if and only if $\mathbf{T}(d_1) \subseteq \mathbf{T}(d_2)$. From the discussion at the end of the previous section, we can conclude that the infinitary type theory $\mathbf{T}$ characterizes applicative bisimulation in terms of the types of the closed terms: $M \lesssim^B N$ if and only if $\mathbf{T}([\![M]\!]) \subseteq \mathbf{T}([\![N]\!])$.

Furthermore, $\mathcal{D}$ is a good model for $\mathbf{T}$ in the sense that continuous functions preserve logical properties, that is, for every $d \in \mathcal{D}$, if $\mathit{unfold}(d) = f \neq \bot$ then $\mathbf{T}(d_1) \subseteq \mathbf{T}(d_2)$ implies $\mathbf{T}(f(d_1)) \subseteq \mathbf{T}(f(d_2))$.

4 The Type Assignment System

In this section we define a type assignment system relative to the infinitary type theory $\mathbf{T}$. Judgments have the form $\Gamma \vdash M{:}\phi$, where the *subject* M is an untyped λ-term, the *predicate* ϕ is a type in $\mathbf{T}$, and the *assumption* Γ is a set consisting of pairs $x{:}\phi_x$ for x a variable in $\mathrm{V}ar$ and ϕ_x a type in $\mathbf{T}$. We assume that each variable in Γ appears at most once, and that all free variables of M are declared in Γ. Furthermore we denote by $\Delta \leq \Gamma$ the fact that Δ and Γ are both assumptions about the same variables, and $\vdash \phi_x \leq \phi'_x$ for all $x{:}\phi_x \in \Delta$ and $x{:}\phi'_x \in \Gamma$.

Validity of the typing judgments is defined in terms of the meaning of λ-terms as elements of $\mathcal{D}$ defined previously:

$$\begin{aligned} \Gamma \models M{:}\phi \iff & \forall \rho{:}\mathrm{V}ar \to \mathcal{D}.\ \forall x{:}\phi_x \in \Gamma. \\ & \rho(x) \models \phi_x \text{ implies } [\![M]\!]_\rho \models \phi\,. \end{aligned}$$

The following system gives an axiomatization of the type judgments.

- $$\frac{\Gamma \leq \Delta \text{ and } \Delta \vdash M{:}\phi_1 \text{ and } \phi_1 \leq \phi_2}{\Gamma \vdash M{:}\phi_2}$$
- $$\frac{\{\Gamma \vdash M{:}\phi_i\}_{i\in I}}{\Gamma \vdash M{:}\bigwedge_{i\in I}\phi_i}$$
- $$\frac{\{\Gamma, x{:}\phi_i \vdash M{:}\phi\}_{i\in I}}{\Gamma, x{:}\bigvee_{i\in I}\phi_i \vdash M{:}\phi}$$
- $$\frac{\Gamma \vdash M{:}(\phi_1 \to \phi_2)_\bot \text{ and } \Gamma \vdash N{:}\phi_1}{\Gamma \vdash MN{:}\phi_2}$$
- $$\frac{\Gamma, x{:}\phi_1 \vdash M{:}\phi_2}{\Gamma \vdash \lambda x.M{:}(\phi_1 \to \phi_2)_\bot}$$
- $$\frac{x{:}\phi \in \Gamma}{\Gamma \vdash x{:}\phi}$$

The above type judgment is standard. Basically, it is Abramsky's endogenous logic $\Lambda(\sigma)$ for the type expression $\sigma \equiv \mathbf{rec}\,t.(t \to t)_\bot$ [2] augmented to allow for infinite meets and joins. Its presentation is rather intuitive, and differs from the standard intersection type assignment system because abstractions receive a type that is build by means of the type constructors '$\to$' and '$(\cdot)_\bot$'.

Soundness is proved by induction on the lengh of the proofs.

Theorem 3. *For all assumptions Γ, λ-terms M, and types $\phi \in T$,*

$$\Gamma \vdash M{:}\phi \text{ implies } \Gamma \models M{:}\phi\,.$$

To get completeness we proceed as before. First we consider the finitary restriction of the above type judgment and apply Abramsky's completeness theorem for the endogenous logic $\Lambda(\sigma)$ [2].

Proposition 4. *For all λ-terms M, finitary types $\phi \in \mathbf{T}_\omega$, and assumptions Γ such that $\phi_x \in \mathbf{T}_\omega$ for every $x{:}\phi_x \in \Gamma$,*

$$\Gamma \models M{:}\phi \text{ implies } \Gamma \vdash M{:}\phi\,.$$

Using now Lemma 1 we can finally state the following completeness theorem.

Theorem 4. *(Completeness) For all assumptions Γ, λ-terms M, and types $\phi \in T$,*

$$\Gamma \models M{:}\phi \text{ if and only if } \Gamma \vdash M{:}\phi\,.$$

5 Conclusion

We have presented a complete infinitary intersection type system with respect to the ordinary set-theoretic semantics. Although we have defined it only for the lazy λ-calculus, our result is general enough to be applied to other λ-calculi based on other domain theoretical function spaces (e.g. the Scott function space, the strict function space, and the Landin-Plotkin function space [1]). Also extensions with basic types can be considered by associating each basic type to a fixed Scott domain, and extending coprimeness, entailment, and typing judgements by taking into account upper-closed subsets of these domains as formulae.

Our type system includes union types, and inherits from [2] the need of a 'coprimeness predicate' for proving completeness. This implies that it is not clear how to interpret our type system, maintaining completeness, to applicative transition systems rather than to a single Scott domain. One possibility could be to restrict the syntax of the types, for example by not allowing for (infinite) union types to occur at the right-hand side of the $\rightarrow$ constructor. Another possibility would be to define a type interpretation directly connected to the usual one for intersection types only, as in [4].

The types we consider in this paper are infinite. However there are interesting fragments that have a finite and transparent representation [14]. The size of the types can be limited without losing completeness. Since $\mathcal{D}$ is an ω-algebraic cpo, its cardinality is at most $2^{\aleph_0}$, while that of the collection of its saturated sets is at most $2^{|\mathcal{D}|} = 2^{2^{\aleph_0}}$. By Theorem 2 types in $\mathbf{T}$ are identified (up-to logical equivalence) with saturated sets of $\mathcal{D}$. Thus we can constrain the size of the index set of the intersection and union type constructors by any cardinal greater than $2^{2^{\aleph_0}}$, that is, under the Generalized Continuum Hypothesis, the infinite cardinal $\aleph_2$.

References

1. S. Abramsky. The lazy lambda calculus. In *Research Topics in Functional Programming*, pp. 65–117, Addison Wesley, 1990.
2. S. Abramsky. Domain theory in logical form. *Annals of Pure and Applied Logic*, 51(1):1–77, 1991.
3. F. Barbanera and M. Dezani-Ciancaglini. Intersection and union types. In *Proceedings of TACS'91*, vol. 526 of *Lecture Notes in Computer Science*, pp. 651–674. Springer-Verlag, 1991.
4. F. Barbanera, M. Dezani-Ciancaglini, and U. dé Liguoro. Intersection and union types: syntax and semantics. *Information and Computation*, 119:202–230, 1995.
5. H. Barendregt. *The Lambda Calculus: Its Syntax and Semantics.* North Holland, 1984.
6. H. Barendregt, M. Coppo, and M. Dezani-Ciancaglini. A filter lambda model and the completeness of type assignment. *Journal of Symbolic Logic* 48(4):931–940, 1983.
7. M.M. Bonsangue, B. Jacobs, and J.N. Kok. Duality beyond sober spaces: topological spaces and observation frames. *Theoretical Computer Science*, 151(1):79–124, 1995.
8. M.M. Bonsangue. *Topological Dualities in Semantics.* Vol. 8 of *Electronic Notes in Theoretical Computer Science.* Elsevier Science, 1997.
9. M.M. Bonsangue and J.N. Kok. Towards an infinitary logic of domains: Abramsky logic for transition systems. *Information and Computation*, 155:170–201, 1999.
10. M. Coppo and M. Dezani-Ciancaglini. A new type-assignment for λ-term. In *Archiv. Math. Logik*, 19:139–156, 1978.
11. M. Dezani-Ciancaglini, F. Honsell and Y. Motohama. Compositional characterizations of λ-terms using intersection types. In *Proceedings of MFCS 2000*, vol. 1893 of *Lecture Notes in Computer Science*, pp. 304–313. Springer-Verlag, 2000.
12. K.H. Hofmann and M.W. Mislove. Local compactness and continuous lattices. In *Continuous lattices - Proceedings Bremen 1979*, vol. 871 of *Lecture Notes in Mathematics*, pp. 209–248. Springer-Verlag, 1981.
13. P.T. Johnstone. *Stone Spaces.* Cambridge University Press, 1982.
14. D. Leivant. Discrete polymorphism. In *Proceedings of 1990 ACM Conference of LISP and Functional Programming*, pp. 288–297.
15. B.C. Pierce. Programming with intersection types, union types, and polymorphism. Technical Report CMU-CS-91-106, Carnegie Mellon University, 1991.
16. G. Pottinger. A type assignment for the strongly normalizable λ-term. In *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus, and Formalism*, pp. 561-577. Academic Press, 1980.
17. G. Raney. Completely distributive complete lattices. In *Proceedings of the American Mathematical Society* vol. 3(4), pp. 677–680, 1952.
18. D.S. Scott Domains for denotational semantics. In M. Nielson and E.M. Schmidt (eds.), *International Colloquioum on Automata, Languages, and Programs*, volume 140 of *Lecture Notes in Computer Science*, pp 577-613, Springer-Verlag, 1982.