Intersection Types and Denotational Semantics: An Extended Abstract

Simona Ronchi Della Rocca
Università degli Studi di Torino, dept Computer Science, Torino, Italy
ronchi@di.unito.it

Abstract

This is a short survey of the use of intersection types for reasoning in a finitary way about terms interpretations in various models of lambda-calculus.

2012 ACM Subject Classification Theory of computation → Denotational semantics

Keywords and phrases Lambda-calculus, Lambda-models, Intersection types

Digital Object Identifier 10.4230/LIPIcs.TYPES.2016.2

Category Invited Paper

1 Introduction

Intersection types have been introduced by Coppo and Dezani [6], with the aim of enforcing the typability power of simple types, but they quite immediately turned out to be a very powerful tool to reason about the semantics properties of programming languages.

In the general framework of denotational semantics of λ-calculus, intersection types supply a logical description of various kind of λ-models. In particular they allow for a finitary description of the interpretation of terms, through type assignment systems, assigning types to terms starting from a context (finite assignment of types to free variables). Terms are interpreted by sets either of types or of pairs of the shape (context, type), so reasoning about the interpretation of a term can be done via type inference; in fact, in order to prove the equivalence between two terms, it is sufficient to show that they share the same type derivations. Although the type inference is usually undecidable, such a logical description of models supplies concrete tools to reason in a finitary way about the interpretation of terms, since a derivation grasps a finite piece of the semantic-interpretations.

The aim of this brief survey is to illustrate how this technique can be applied to three different classes of models, in the general settings of continuous, stable and relational semantics. I recall that a λ-model is a reflexive object in a cartesian closed category, i.e., a space $\mathtt{D}$ such that the set of morphisms from $\mathtt{D}$ to $\mathtt{D}$ is a retract of $\mathtt{D}$ (in a more concise way $\mathtt{D} \rhd [\mathtt{D} \to \mathtt{D}]$) [2].

2 Continuous Semantics

The first local description of a continuous λ-model through intersection types is in [3]. Then various instances of continuous models have been studied (see, between others, [7, 8, 13, 1]). A general correspondence between intersection types and continuous λ-models has been described in [18].

22nd International Conference on Types for Proofs and Programs (TYPES 2016).
Editors: Silvia Ghilezan, Herman Geuvers, and Jelena Ivetić; Article No. 2; pp. 2:1–2:7
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

$$\frac{}{\mathtt{x}:\{\mathtt{A}\}\vdash_{\mathtt{C},\nabla}\mathtt{x}:\mathtt{A}}\ ax \qquad \frac{\Gamma\vdash_{\mathtt{C},\nabla}\mathtt{M}:\mathtt{A}\quad \mathtt{A}\leq_\nabla\mathtt{B}}{\Gamma\vdash_{\mathtt{C},\nabla}\mathtt{M}:\mathtt{B}}\ \leq_\nabla \qquad \frac{\Gamma,\mathtt{x}:\sigma\vdash_{\mathtt{C},\nabla}\mathtt{M}:\mathtt{A}}{\Gamma,\mathtt{x}:\sigma\cup\tau\vdash_{\mathtt{C},\nabla}\mathtt{M}:\mathtt{A}}\ w$$

$$\frac{\Gamma,\mathtt{x}:\sigma\vdash_{\mathtt{C},\nabla}\mathtt{M}:\mathtt{A}}{\Gamma\vdash_{\mathtt{C},\nabla}\lambda\mathtt{x}.\mathtt{M}:\sigma\to\mathtt{A}}\ \to I \qquad \frac{\Gamma\vdash_{\mathtt{C},\nabla}\mathtt{M}:\sigma\to\mathtt{A}\quad(\Delta_\mathtt{B}\vdash_{\mathtt{C},\nabla}\mathtt{N}:\mathtt{B})_{\mathtt{B}\in\sigma}}{\Gamma\bigcup_{\mathtt{B}\in\sigma}\Delta_\mathtt{B}\vdash_{\mathtt{C},\nabla}\mathtt{MN}:\mathtt{A}}\ \to E$$

Figure 1 The continuous parametric type assignment system.

▶ Definition 1.

- Types ($\mathcal{T}_\mathtt{C}$) are defined (starting from a countable set $\mathtt{C}$ of constants) as:

  $$\mathtt{A},\mathtt{B} ::= \mathtt{a} \mid \sigma\to\mathtt{A}$$

  where $\mathtt{a}\in\mathtt{C}$, and σ is a finite set of types.
- Let ∇ be any pre-order on $\mathcal{T}_\mathtt{C}$, extended to sets in the following way:

  $$\begin{array}{lcl}\sigma\subseteq\tau & \Rightarrow & \tau\leq\sigma\\ \sigma\leq\tau,\mathtt{A}\leq\mathtt{B} & \Rightarrow & \sigma\cup\{\mathtt{A}\}\leq\tau\cup\{\mathtt{B}\}\end{array}$$

  and closed under:

  $$\sigma'\leq\sigma,\mathtt{A}\leq\mathtt{B} \quad\Leftrightarrow\quad \sigma\to\mathtt{A}\leq\sigma'\to\mathtt{B}$$
- Let $\simeq_\nabla$ be the congruence induced by ∇, and $\leq_\nabla$ the partial order on $\mathcal{T}_\mathtt{C}/\simeq_\nabla$.

▶ Remark. In order to describe different approaches in a uniform manner, I do not use explicit the intersection connective. A set of types $\{\mathtt{A}_1,\ldots,\mathtt{A}_n\}$ corresponds to the more standard notation $\mathtt{A}_1\wedge\ldots\wedge\mathtt{A}_n$, where the intersection connective $\wedge$ is considered modulo idempotency, associativity and commutativity.

The continuous type assignment system, parametric with respect to C,∇, assigning to λ-terms types in $\mathcal{T}_C$ is defined in Figure 1, where Γ,Δ (contexts) are functions from variables to finite subsets of $\mathcal{T}_C$, such that $\Gamma(\mathtt{x})\neq\emptyset$ for a finite number of variables.

Every type assignment system of this kind can be seen as a finitary description of the interpretation of terms in a continuous λ-model $\mathcal{D}_{\mathtt{C},\nabla}$. In fact, for every set of types $\mathcal{T}_\mathtt{C}$ and every partial order $\leq_\nabla$, the set of subsets of $\mathcal{T}_\mathtt{C}/\simeq_\nabla$ equipped with the partial order defined as: ($\sigma\sqsubseteq_{\mathtt{C},\nabla}\tau$ if and only if $\tau\leq_\nabla\sigma$) is a prime-algebraic lattice $\mathtt{D}_{\mathtt{C},\nabla}$, whose prime elements are the singleton over $\mathcal{T}_\mathtt{C}/\simeq_\nabla$.

Moreover, types are notations for step functions, interpreting $\sigma\to\mathtt{A}$ as a step function approximating f, for every continuous function f such that $\mathtt{A}\in f(\tau)$, for each $\tau\sqsupseteq_{\mathtt{C},\nabla}\sigma$. Under this interpretation $\mathtt{D}_{\mathtt{C},\nabla}$ is a linear solution of the domain equation $D\rhd[D\Rightarrow_c D]$, where $[.\Rightarrow_c .]$ denotes the space of continuous functions ordered pointwise, and the linearity of the solution means that both the immersion-projection functions map prime elements into prime elements. Let us call *linear continuous models* the models of this class. So $\mathtt{D}_{\mathtt{C},\nabla}$ gives rise to a linear continuous λ-model $\mathcal{D}_{\mathtt{C},\nabla}$. On the other direction, every linear continuous model can be described through a set of types equipped by a suitable intersection relation. Let $[\![.]\!]^{\mathcal{D}_{\mathtt{C},\nabla}}$ be the interpretation function in $\mathcal{D}_{\mathtt{C},\nabla}$, defined as usual: the following theorem holds.

▶ **Theorem 2.** $\Gamma \vdash_{\mathtt{C},\nabla} \mathtt{M} : \mathtt{A}$ *if and only if* $\mathtt{A} \in [\![\mathtt{M}]\!]_{\rho}^{\mathcal{D}_{\mathtt{C},\nabla}}$, *for all* ρ *such that* $\Gamma(\mathtt{x}) \subseteq \rho(\mathtt{x})$.

So the interpretation of a term in $\mathcal{M}_{\mathtt{C},\nabla}$ is simply the set of types that can be assigned to it.

▶ Remark. If we define a filter to be a set of types closed under $\leq_\nabla$, it turns out that the set of types derivable for a given term is a filter. This justifies the fact that models of this kind have been called "filter models" in the literature.

▶ **Example 3.**
1. The filter model in [3] (where $\mathtt{C}$ is an infinite set of type constants and ∇ is the preorder induced by the empty set of rules) supplies a solution of the domain equation $\mathtt{D} = \mathtt{C} \times [\mathtt{D} \Rightarrow \mathtt{D}]$, through a coding where the type constant $\mathtt{a}$ codes the prime element $(\mathtt{a}, \bot)$, the type $\sigma \to \mathtt{A}$ codes the prime element $(\bot, \sigma \to \mathtt{A})$, and the equation is solved by the bijection $(\bot, \sigma \to \mathtt{A}) \mapsto \sigma \to \mathtt{A}$.
2. The $\mathcal{D}_\infty$ model of Scott [19] is described by $\mathtt{C} = \{\mathtt{a}\}$ and ∇ is induced by the rules $\{\mathtt{a} \leq \emptyset \to \mathtt{a}, \emptyset \to \mathtt{a} \leq \mathtt{a}\}$ ([18]).
3. The Park model [17] is described by $\mathtt{C} = \{\mathtt{a}\}$, and ∇ is induced by the rules $\{\mathtt{a} \leq \{\mathtt{a}\} \to \mathtt{a}, \{\mathtt{a}\} \to \mathtt{a} \leq \mathtt{a}\}$ ([13]).

3 Stable semantics

The stable semantics is based on the notion of stable functions, introduced by Berry [4]. Here I will consider a particular class of models based on stable functions, namely the coherence spaces of Girard [11, 10]. A general correspondence between intersection types and stable λ-model has been described in [15], based on a previous work studying the correspondence between intersection types and qualitative models [12].

▶ **Definition 4.** Let $\mathtt{C}$ be a countable set of constants. Types and coherence type theory are mutually defined as follows.

- Types ($\mathcal{T}_{\mathtt{C},\nabla}$) are defined as:

$$\mathtt{A}, \mathtt{B} ::= \mathtt{a} \mid \sigma \to \mathtt{A}$$

  where $\mathtt{a} \in \mathtt{C}$, and σ is a finite set of types, such that $\frown_\nabla (\sigma)$. i.e., $\mathtt{A} \frown_\nabla \mathtt{B}$, for all $\mathtt{A}, \mathtt{B} \in \sigma, \mathtt{A} \not\simeq_\triangle \mathtt{B}$.
- A coherence type theory ∇ is a pair $(\frown_\nabla, \simeq_\nabla)$, where
  - $\frown_\nabla$ is a symmetric and antireflexive relation on $\mathcal{T}_{\mathtt{C},\nabla}$, closed under

$$\sigma \to \mathtt{A} \frown_\nabla \tau \to \mathtt{B} \Leftrightarrow \text{ either } \mathtt{A} \frown_\nabla \mathtt{B} \text{ or } \exists \mathtt{A}' \in \sigma, \exists \mathtt{B}' \in \tau \text{ such that } \mathtt{A}' \smile_\nabla \mathtt{B}'$$

    (where $\mathtt{A}' \smile_\nabla \mathtt{B}'$ means $\mathtt{A}' \not\frown_\nabla \mathtt{B}'$ and $\mathtt{A}' \not\simeq_\nabla \mathtt{B}'$).
  - $\simeq_\nabla$ is an equivalence on types, extended to sets to sets in such a way that:

$$\sigma \simeq_\nabla \tau \Leftrightarrow \text{ their elements are pairwise } \simeq_\nabla .$$

The stable parametric type assignment system, parametric with respect to $\mathtt{C}, \nabla$ is defined in Figure 2, where Γ, Δ (contexts) are functions from variables to finite sets of types, such that $\Gamma(\mathtt{x}) = \sigma$ implies $\frown_\nabla (\sigma)$ and such that $\Gamma(\mathtt{x}) \neq \emptyset$ for a finite number of variables. Moreover $\bigstar_\nabla(\Delta_1, ..., \Delta_n)$ means ($\mathtt{x} : \sigma \in \Delta_i$ and $\mathtt{x} : \tau \in \Delta_j$ imply either $\frown_\nabla (\sigma, \tau)$ or $\sigma \simeq_\triangle \tau$).

Every type assignment system of this kind can be seen as a finitary description of the interpretation of terms in a stable linear λ-model. In fact, for every choice of $\mathtt{C}$ and ∇, $\mathtt{S}_{\mathtt{C},\nabla} = (\mathcal{T}_{\mathtt{C}}/ \simeq_\nabla, \supset)$, where $\supset$ (*coherence*) is the set theoretic union of $\frown_\nabla$ and $\simeq_\nabla$, is a

$$\frac{}{\mathtt{x}:\{\mathtt{A}\}\vdash_{\mathtt{C},\triangledown}\mathtt{x}:\mathtt{A}}\,ax \qquad \frac{\Gamma\vdash_{\mathtt{C},\triangledown}\mathtt{M}:\mathtt{A}\quad \mathtt{A}\simeq_{\triangledown}\mathtt{B}}{\Gamma\vdash_{\mathtt{C},\triangledown}\mathtt{M}:\mathtt{B}}\,\simeq_{\triangledown}$$

$$\frac{\Gamma,\mathtt{x}:\sigma\vdash_{\mathtt{C},\triangledown}\mathtt{M}:\mathtt{A}}{\Gamma\vdash_{\mathtt{C},\triangledown}\lambda\mathtt{x}.\mathtt{M}:\sigma\to\mathtt{A}}\to I$$

$$\frac{\Gamma\vdash_{\mathtt{C},\triangledown}\mathtt{M}:\sigma\to\mathtt{A}\quad(\Delta_{\mathtt{B}}\vdash_{\mathtt{C},\triangledown}\mathtt{N}:\mathtt{B})_{\mathtt{B}\in\sigma}\quad\bigstar_{\triangledown}(\Gamma\cup(\bigcup_{\mathtt{B}\in\sigma}\Delta_{\mathtt{B}}))}{\Gamma\bigcup_{\mathtt{B}\in\sigma}\Delta_{B}\vdash_{\mathtt{C},\triangledown}\mathtt{MN}:\mathtt{A}}\to E$$

Figure 2 The stable parametric type assignment system.

coherence space. Moreover, types supply a notation for finite pieces of stable functions, interpreting $\sigma\to\mathtt{A}$ as an element of the trace of every stable function f, such that $\mathtt{A}\in f(\sigma)$ and $\forall\tau\subseteq\sigma$, $\mathtt{A}\in f(\tau)$ if and only if $\tau\simeq_{\triangledown}\sigma$. Under this interpretation $\mathtt{S}_{\mathtt{C},\triangledown}$ is a linear solution of the domain equation $D\triangleright[D\Rightarrow_s D]$, where $[.\Rightarrow_s .]$ denotes the space of stable functions ordered pointwise. Let define *linear stable models* the models of this class.

So $\mathtt{S}_{\mathtt{C},\triangledown}$ gives rise to a λ-model, let $\mathcal{S}_{\mathtt{C},\triangledown}$. On the other direction, every linear stable model can be described through a set of types equipped by a suitable type theory. Let $[\![.]\!]^{\mathcal{S}_{\mathtt{C},\triangledown}}$ be the interpretation function in $\mathcal{S}_{\mathtt{C},\triangledown}$, defined in the usual way; the following theorem holds.

▶ **Theorem 5.** $\Gamma\vdash_{\mathtt{C},\triangledown}\mathtt{M}:\mathtt{A}$ *if and only if* $\mathtt{A}\in[\![\mathtt{M}]\!]_{\rho}^{\mathcal{S}_{\mathtt{C},\triangledown}}$, *for all* ρ *such that* $\Gamma(\mathtt{x})\subseteq\rho(\mathtt{x})$.

So the interpretation of a term in $\mathcal{S}_{\mathtt{C},\triangledown}$ is simply the set of types that can be assigned to it.

▶ **Example 6.**

1. Let $\mathtt{C}$ be a countable infinite set, $\triangledown=(\frown,\simeq)$, where $\frown$ is the minimum relation induced by the rules: $\mathtt{a}\frown\mathtt{A},\forall\mathtt{a}\in\mathtt{C},\mathtt{A}\in\mathcal{T}_{\mathtt{C},\triangledown}$ and $\simeq$ is the minimum congruence on types, satisfying the conditions of Def.4. Then $(\mathcal{T}_{\mathtt{C},\triangledown},\frown)$ supplies a solution of the equation $\mathtt{S}=\mathtt{C}\,\&\,[\mathtt{S}\Rightarrow_s\mathtt{S}]$, choosing the type $\mathtt{a}$ to describe the token $(1,\mathtt{a})$, and $\sigma\to\mathtt{A}$ to describe the token $(2,\sigma\to\mathtt{A})$. The solution is induced by the bijection $(2,\sigma\to\mathtt{A})\mapsto\sigma\to\mathtt{A}$.
2. Let $\mathtt{C}=\{\mathtt{a}\}$, let $\frown$ be the minimum relation and let $\simeq$ be the minimum congruence induced by $\{\mathtt{a}\simeq\emptyset\to\mathtt{a}\}$. Then the resulting λ-model is (in some sense) the stable corresponding to $\mathcal{D}_\infty$, built in [12].
3. Let $\mathtt{C}=\{\mathtt{a}\}$, let $\frown$ be as in the previous point, and let $\simeq$ be the minimum congruence induced by $\{\mathtt{a}\simeq\{\mathtt{a}\}\to\mathtt{a}\}$. Then the resulting λ-model is (in some sense) the stable corresponding to the Park model, built in [12].

4 Relational semantics

The relational semantics has been developed by Bucciarelli, Manzonetto and Ehrhard ([5]). A general correspondence between intersection types and relational λ-model has been described in [16].

$$\frac{}{\mathtt{x} : [\mathtt{A}] \vdash_{\mathtt{C},\nabla} \mathtt{x} : \mathtt{A}}\ ax \qquad \frac{\Gamma \vdash_{\mathtt{C},\nabla} \mathtt{M} : \mathtt{A} \quad \mathtt{A} \simeq_\nabla \mathtt{B}}{\Gamma \vdash_{\mathtt{C},\nabla} \mathtt{M} : \mathtt{B}}\ \simeq_\nabla$$

$$\frac{\Gamma, \mathtt{x} : \sigma \vdash_{\mathtt{C},\nabla} \mathtt{M} : \mathtt{A}}{\Gamma \vdash_{\mathtt{C},\nabla} \lambda \mathtt{x}.\mathtt{M} : \sigma \to \mathtt{A}}\ \to I$$

$$\frac{\Gamma \vdash_{\mathtt{C},\nabla} \mathtt{M} : \sigma \to \mathtt{A} \quad (\Delta_\mathtt{B} \vdash_{\mathtt{C},\nabla} \mathtt{N} : \mathtt{B})_{\mathtt{B} \in \sigma}}{\Gamma \uplus_{\mathtt{B} \in \sigma} \Delta_B \vdash_{\mathtt{C},\nabla} \mathtt{MN} : \mathtt{A}}\ \to E$$

Figure 3 The relational parametric type assignment system.

▶ **Definition 7.**

- Types ($\mathcal{T}_\mathtt{C}$) are defined (starting from a countable set $\mathtt{C}$ of constants) as:

  $$\mathtt{A}, \mathtt{B} ::= \mathtt{a} \mid \sigma \to \mathtt{A}$$

  where σ is a finite multiset of types.
- A relational type theory $\simeq_\nabla$ is a congruence on types, behaving on multisets in the following way:

  $$\sigma \simeq_\nabla \tau \quad \Leftrightarrow \quad \sigma = [\mathtt{A}_1, ..., \mathtt{A}_n], \tau = [\mathtt{B}_1, ..., \mathtt{B}_n] \text{ and } \mathtt{A}_i \simeq_\nabla \mathtt{B}_i$$

  and closed under:

  $$\sigma \to \mathtt{A} \simeq_\nabla \sigma' \to \mathtt{B} \quad \Leftrightarrow \quad \sigma \simeq_\nabla \sigma', \mathtt{A} \simeq \mathtt{B}$$

▶ Remark. The multisets of types $[\mathtt{A}_1, ..., \mathtt{A}_n]$ is an alternative notation for $\mathtt{A}_1 \wedge ... \wedge \mathtt{A}_n$, where the intersection connective enjoys associativity and commutativity, but not idempotence; so the congruence on multisets needs to take into account the multiplicity of elements. Let $[\,]$ denote the empty multiset.

The relational parametric type assignment system, parametric with respect to $\mathtt{C}, \nabla$ is defined in Figure 3, where Γ, Δ (contexts) are functions from variables to finite multisets of types, such that $\Gamma(\mathtt{x}) \neq [\,]$ for a finite number of variables. Moreover $\uplus$ denotes the multiset union.

An arrow type denotes a relation from $\mathcal{M}_{fin}(\mathcal{T}_\mathtt{C}/\simeq_\nabla)$ and $\mathcal{T}_\mathtt{C}/\simeq_\nabla$, where $\mathcal{M}_{fin}(.)$ is the set of finite multisets. It turns out that $\mathcal{T}_\mathtt{C}/\simeq_\nabla$ supplies a solution of the equation $\mathtt{U} \rhd [\mathtt{U} \Rightarrow_r \mathtt{U}]$, where $\mathtt{U}$ is a set and $[. \Rightarrow_r .]$ denotes the space of relations between $\mathcal{M}_{fin}(\mathtt{U})$ and $\mathtt{U}$. But this space does not supply directly a λ-model, as shown in [9], since it has not enough points. It is necessary to consider the space $Fin(\mathtt{U}^{Var}, \mathtt{U})$, which consists of the finitary morphisms from $\mathtt{U}^{Var}$ to $\mathtt{U}$, where Var is a countable set of variables, and gives rise to a λ-model. An element of such a space can be represented by a pre-typing, which is a pair $(\Gamma; \mathtt{A})$ of a context and a type, both considered modulo $\simeq_\nabla$: in fact pre-typings are elements of a space $\mathtt{R}_{\mathtt{C},\nabla} = Fin((\mathcal{T}_\mathtt{C}/\simeq_\nabla)^{Var}, \mathcal{T}_\mathtt{C}/\simeq_\nabla)$ which supplies a λ-model $\mathcal{R}_{\mathtt{C},\nabla}$; let $[\![.]\!]^{\mathcal{R}_{\mathtt{C},\nabla}}$ be its interpretation function, defined in the usual way. The following theorem holds.

▶ **Theorem 8.** $\Gamma \vdash_{\mathtt{C},\nabla} \mathtt{M} : \mathtt{A}$ *if and only if* $(\Gamma'; \mathtt{A}) \in [\![\mathtt{M}]\!]_\rho^{\mathcal{R}_{\mathtt{C},\nabla}}$, *where* $\Gamma' = \uplus_i \Delta_i, (\Delta_i; \mathtt{A}^i) \in \rho(\mathtt{x}_i)$, *for every* $\mathtt{A}^i \in \Gamma(\mathtt{x}_i)$.

▶ **Example 9.**
- Choosing $\mathtt{C} = \{\mathtt{a}\}$ and the type theory induced by $[\,] \to \mathtt{a} \simeq \mathtt{a}$, we obtain the model studied in [5].
- Choosing $\mathtt{C} = \{\mathtt{a}\}$ and the type theory induced by $[\mathtt{a}] \to \mathtt{a} \simeq \mathtt{a}$, we obtain the model studied in [14].

5 Conclusion

Comparing the three parametric type assignment systems in Figure 1, 2 and 3, it turns out that the three systems are quite different, from a proof theoretical point of view. The continuous system uses idempotent intersection, it enjoys weakening, and requires a subsumption rule. The stable system uses idempotent intersection too, but it is relevant (in the sense that weakening is unsound) and requires an equivalence rule. Finally, the relational system uses non-idempotent intersection, is relevant, and requires an equivalence rule. Moreover the interpretation of a term is the set of types derivable for it in the first two systems, while is the set of pre-typings in the third one. Despite these differences, they supply a logical tools for reasoning in a uniform way about the denotational semantics of terms, in particular for comparing terms from a semantics point of view. In fact, the following theorem holds.

▶ **Theorem 10.** *Let $\mathcal{M}_{\mathtt{C},\nabla}$ be a (continous, stable, relational) λ-model, and let $\sqsubseteq_{\mathcal{M}_{\mathtt{C},\nabla}}$ be the order relation between interpretations of terms in it. Then:*

$$(\forall \Gamma, \mathtt{A}.\ \Gamma \vdash_{\mathtt{C},\nabla} \mathtt{M} : \mathtt{A} \textit{ implies } \Gamma \vdash_{\mathtt{C},\nabla} \mathtt{N} : \mathtt{A}) \textit{ if and only if } \mathtt{M} \sqsubseteq_{\mathcal{M}_{\mathtt{C},\nabla}} \mathtt{N}.$$

References

1 Fabio Alessi, Mariangiola Dezani-Ciancaglini, and Furio Honsell. A complete characterization of complete intersection-type preorders. *ACM Transactions on Computational Logic*, 4(1):120–147, jan 2003.

2 Henk Barendregt. *The Lambda Calculus: Its Syntax and Semantics*, volume 103 of *Studies in logic and the foundation of mathematics*. North-Holland, revised edition, 1984.

3 Henk Barendregt, Mario Coppo, and Mariangiola Dezani-Ciancaglini. A filter lambda model and the completeness of type assignment. *The Journal of Symbolic Logic*, 48(4):931–940, dec 1983.

4 Gérard Berry. Stable models of typed λ-calculi. In Giorgio Ausiello and Corrado Böhm, editors, *Fifth International Colloquium on Automata, Languages and Programming - - ICALP'78, Udine, Italy, July 17-21, 1978*, volume 62 of *Lecture Notes in Computer Science*, pages 72–89. Springer-Verlag, 1978.

5 Antonio Bucciarelli, Thomas Ehrhard, and Giulio Manzonetto. Not enough points is enough. In *CSL'07*, volume 4646 of *Lecture Notes in Computer Science*, pages 298–312. Springer, 2007.

6 Mario Coppo and Mariangiola Dezani-Ciancaglini. An extension of the basic functionality theory for the λ-calculus. *Notre Dame Journal of Formal Logic*, 21(4):685–693, oct 1980.

7 Mario Coppo, Mariangiola Dezani-Ciancaglini, Furio Honsell, and Giuseppe Longo. Extended type structure and filter lambda models. In *Logic Colloquim'82*, pages 241–262, 1984.

8 Mario Coppo, Mariangiola Dezani-Ciancaglini, and Maddalena Zacchi. Type theories, normal forms, and D_∞-lambda-models. *Information and Computation*, 72(2):85–116, 1987.

9 Daniel de Carvalho. Execution time of lambda-terms via denotational semantics and intersection types. *CoRR*, abs/0905.4251, 2009. Available also as INRIA report RR 6638. URL: `http://arxiv.org/abs/0905.4251`.

10 Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.
11 Jean-Yves Girard, Yves Lafont, and Paul Taylor. *Proofs and Types*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, Cambridge, 1989.
12 Furio Honsell and Simona Ronchi della Rocca. Reasoning about interpretation in qualitative lambda-models. In *IFIP 2.2*, pages 505–521, 1990.
13 Furio Honsell and Simona Ronchi Della Rocca. An approximation theorem for topological lambda models and the topological incompleteness of lambda calculus. *Journal of Computer and System Sciences*, 45(1):49–75, aug 1992.
14 Giulio Manzonetto and Domenico Ruoppolo. Relational graph models, Taylor expansion and extensionality. *Electronic Notes in Theoretical Computer Science*, 308:245–272, 2014.
15 Luca Paolini, Mauro Piccolo, and Simona Ronchi Della Rocca. Logical semantics for stability. In *MFPS'09*, volume 249 of *Electronic Notes in Theoretical Computer Science*, pages 429–449, 2009.
16 Luca Paolini, Mauro Piccolo, and Simona Ronchi Della Rocca. Essential and relational models. *Mathematical Structures in Computer Science*, FirstView:1–25, 9 2015. `doi:10.1017/S0960129515000316`.
17 D. M. R. Park. The Y-combinator in scott's lambda-calculus models. Research Report CS-RR-013, Department of Computer Science, University of Warwick, Coventry, UK, jun 1976. URL: `http://www.dcs.warwick.ac.uk/pub/reports/rr/013.html`.
18 Simona Ronchi Della Rocca and Luca Paolini. *The Parametric λ-Calculus: a Metamodel for Computation*. Texts in Theoretical Computer Science: An EATCS Series. Springer-Verlag, 2004.
19 Dana S. Scott. Data types as lattices. *SIAM Journal of Computing*, 5:522–587, sep 1976.