Relational semantics: from simple to non-idempotent intersection types and back

Giulio Guerrieri[1] and Vincent Sommella[1]

University of Sussex, Department of Informatics, Brighton, UK.
{g.guerrieri,v.sommella}@sussex.ac.uk

Abstract

Relational semantics is a simple and well-studied denotational model for the untyped lambda-calculus. It can be presented syntactically as a non-idempotent intersection type system and provides qualitative and quantitative information, such as the characterization of normalizing terms and their execution time (i.e., the number of steps to reach the normal form). We study the relational semantics for the simply typed lambda-calculus extended with the fixpoint combinator; we show that the interpretation of a simply typed term is nothing but the interpretation of the untyped term restricted to the non-idempotent intersection types refining the simple type. Our approach is based on merging a simply typed derivation with a non-idempotent intersection one, which is not trivial since simply typed terms do not enjoy subject expansion and may not be normalizing (because of the fixpoint combinator). We also show that some well-known qualitative and quantitative completeness results provided by relational semantics and non-idempotent intersection type derivations for the untyped lambda-calculus lifts to the simply typed lambda-calculus with fixpoint combinator nearly for free, thanks to our results.

1 Content

Relational semantics is a simple and well-studied denotational model for the untyped λ-calculus [5, 14, 3]. Inspired by Girard's linear logic [10, 11, 4], it can be presented syntactically as a non-idempotent intersection type system [9, 13, 12] and provides qualitative and quantitative information, such as the characterization of normalizing terms and their execution time (i.e., the number of steps to reach the normal form) [7, 8, 6, 2, 1].

We study the relational semantics for the simply typed λ-calculus extended with the fixpoint combinator. Syntactically, this can be presented by defining a type system similar to the one for the untyped λ-calculus where:

1. each rule presents a non-idempotent intersection type *refining* a simple type, for an obvious definition of refinement;

2. the interpretation of the fixpoint combinator is the same as for *any* fixpoint combinator in the untyped λ-calculus.

We show that the interpretation of a simply typed term is nothing but the interpretation of the untyped term *restricted* to the non-idempotent intersection types *refining* the simple type of the term. Our approach is based on merging a simply typed derivation with a non-idempotent intersection one, showing how the refinement information propagates top-down (which is obvious) and bottom-up (which is harder to prove) in a derivation. This allows us to circumvent two technical problems in our proof:

1. a simply typed term obtained from β-reducing another one may not enjoy subject expansion (because simple types are too restrictive), and

2. in the language under study, simply typed terms may not be normalizing (because of the fixpoint combinator).

We then prove that some well-known qualitative and quantitative information provided by relational semantics and non-idempotent intersection type derivations for the untyped λ-calculus lifts to the simply typed λ-calculus with fixpoint combinator almost for free.

Specifically, following the methodology exposed in [1], we show how normalization of head reduction and leftmost-outmost reduction strategies can be characterized by typability (with a suitable non-idempotent intersection type), how their evaluation length plus size of normal form can be bounded by the size of a derivation and finally how sizes of types bounds the sizes of normal forms. All the bounds become tight when the non-idempotent intersection types are of a suitable form and the simply typed term is η-long. Our result about the interpretation of a simply typed term in terms of its untyped interpretation is essential to show the completeness part of those qualitative and quantitative results.

Our contribution is distinct from Pautasso' and Ronchi Della Rocca's [15] in several respects:

1. Pautasso and Ronchi Della Rocca use a variant of non-idempotent intersection types with the same expressive power as simple types (hence they can type less terms than us).

2. The non-idempotent intersection type system used by Pautasso and Ronchi Della Rocca provides different qualitative and quantitative information than the ones we use.

3. Pautasso and Ronchi Della Rocca use a different technique and their results rely on the fact that simple types guarantee strong normalization, hence their approach cannot be extended to the simply typed λ-calculus with fixpoint combinator.

References

[1] Beniamino Accattoli. Semantic bounds and multi types, revisited. In Aniello Murano and Alexandra Silva, editors, *32nd EACSL Annual Conference on Computer Science Logic, CSL 2024, February 19-23, 2024, Naples, Italy*, volume 288 of *LIPIcs*, pages 7:1–7:24. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024.

[2] Beniamino Accattoli, Stéphane Graham-Lengrand, and Delia Kesner. Tight typings and split bounds, fully developed. *J. Funct. Program.*, 30:e14, 2020.

[3] Henk Barendregt and Giulio Manzonetto. *A Lambda Calculus Satellite*. College Publications, 2022.

[4] Antonio Bucciarelli and Thomas Ehrhard. On phase semantics and denotational semantics: the exponentials. *Ann. Pure Appl. Log.*, 109(3):205–241, 2001.

[5] Antonio Bucciarelli, Thomas Ehrhard, and Giulio Manzonetto. Not enough points is enough. In Jacques Duparc and Thomas A. Henzinger, editors, *Computer Science Logic, 21st International Workshop, CSL 2007, 16th Annual Conference of the EACSL, Lausanne, Switzerland, September 11-15, 2007, Proceedings*, volume 4646 of *Lecture Notes in Computer Science*, pages 298–312. Springer, 2007.

[6] Antonio Bucciarelli, Delia Kesner, and Daniel Ventura. Non-idempotent intersection types for the lambda-calculus. *Log. J. IGPL*, 25(4):431–464, 2017.

[7] Daniel de Carvalho. Execution time of lambda-terms via denotational semantics and intersection types. *CoRR*, abs/0905.4251, 2009.

[8] Daniel de Carvalho. Execution time of λ-terms via denotational semantics and intersection types. *Math. Struct. Comput. Sci.*, 28(7):1169–1203, 2018.

[9] Philippa Gardner. Discovering needed reductions using type theory. In Masami Hagiya and John C. Mitchell, editors, *Theoretical Aspects of Computer Software, International Conference TACS '94, Sendai, Japan, April 19-22, 1994, Proceedings*, volume 789 of *Lecture Notes in Computer Science*, pages 555–574. Springer, 1994.

[10] Jean-Yves Girard. Linear logic. *Theor. Comput. Sci.*, 50:1–102, 1987.

[11] Jean-Yves Girard. Normal functors, power series and λ-calculus. *Ann. Pure Appl. Log.*, 37(2):129–177, 1988.

[12] Martin Hyland, Misao Nagayama, John Power, and Giuseppe Rosolini. A category theoretic formulation for engeler-style models of the untyped lambda. In Anthony Karel Seda, Ted Hurley, Michel P. Schellekens, Mícheál Mac an Airchinnigh, and Glenn Strong, editors, *Proceedings of the Third Irish Conference on the Mathematical Foundations of Computer Science and Information Technology, MFCSIT 2004*, volume 161 of *Electronic Notes in Theoretical Computer Science*, pages 43–57. Elsevier, 2006.

[13] Assaf J. Kfoury. A linearization of the lambda-calculus and consequences. *J. Log. Comput.*, 10(3):411–436, 2000.

[14] Luca Paolini, Mauro Piccolo, and Simona Ronchi Della Rocca. Essential and relational models. *Math. Struct. Comput. Sci.*, 27(5):626–650, 2017.

[15] Daniele Pautasso and Simona Ronchi Della Rocca. A quantitative version of simple types. In Marco Gaboardi and Femke van Raamsdonk, editors, *8th International Conference on Formal Structures for Computation and Deduction, FSCD 2023, July 3-6, 2023, Rome, Italy*, volume 260 of *LIPIcs*, pages 29:1–29:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.