Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 323 (2016) 75–91

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

Strong Normalization through Intersection Types and Memory

Antonio Bucciarelli Delia Kesner

Univ Paris Diderot, Sorbonne Paris Cité, PPS, CNRS, Paris, France

Daniel Ventura

Univ. Federal de Goiás, Instituto de Informática, Goiânia, Brazil

Abstract

We characterize β-strongly normalizing λ-terms by means of a non-idempotent intersection type system. More precisely, we first define a memory calculus $\mathtt{K}$ together with a non-idempotent intersection type system $\mathcal{K}$, and we show that a $\mathtt{K}$-term t is typable in $\mathcal{K}$ if and only if t is $\mathtt{K}$-strongly normalizing. We then show that β-strong normalization is equivalent to $\mathtt{K}$-strong normalization. We conclude since λ-terms are strictly included in $\mathtt{K}$-terms.

Keywords: Lambda-calculus, memory calculus, strong normalization, intersection types.

1 Introduction

It is well known that the β-strongly normalizing λ-terms can be characterized as those being typable in suitable intersection type (IT) systems. This result dates back to the late 1970s and early 1980s, when intersection types were invented to endow the pure lambda calculus with powerful type-assignment systems [2,12,28,26]. A survey of these results, out of the scope of this paper, can be found for instance in [35,3].

In more recent years, a revisitation of those early results has been driven by the introduction of resource aware semantics of λ-calculi [21,6,15,7] and the corresponding *non-idempotent* intersection types assignment systems. The inhabitation problem for instance, known to be undecidable in an idempotent setting [32], was proved to be decidable for non-idempotent types [8].

Just like their idempotent precursors, these type systems allow for a characterization of strong normalization [5,14] (as well as weak normalization and head normalization [15,9]), but they also grant a substantial improvement: proving that

http://dx.doi.org/10.1016/j.entcs.2016.06.006
1571-0661/

typable terms are strongly normalizing becomes much simpler. Let us provide a brief account of this improvement, by highlighting in the way the *quantitative* character of non-idempotent intersection types versus the *qualitative* flavor of the idempotent ones. The proof of the highlighted statement above, in the non-idempotent case, goes roughly as follows: given a typing derivation for a term t, and willing to prove that t is strongly normalizing, take whatever β-reduct t' of t. The subject reduction lemma, in this case, ensures *not only* that t' is typable *but also* that there exists a typing derivation for t' whose size is smaller than the one of the typing derivation for t we started from. Hence any β-reduction sequence starting from t is finite.

This shrinking of the size of typing derivations along reduction sequences, in sharp contrast to what happens in the idempotent setting, is essentially due to the fact that a type derivation for a term of the shape $(\lambda x.u)v$ may require as many sub-derivations for v as the number of occurrences of x in u [1]. Let us provide a simple example involving the Church numeral $\underline{n} := \lambda y.\lambda x.y(y(...yx)...)$.

Why is the term $u = \lambda x.t(t(...tx)...)$, t being an arbitrarily complex term, "simpler to type" than its β-expanded form $\underline{n}t$? The point is that the typical non-idempotent intersection type [2] that can be assigned to the Church numeral $\underline{n}$ is, in our notation, $[[\sigma] \to \sigma, ..., [\sigma] \to \sigma] \to [\sigma] \to \sigma$, the leftmost multiset containing n copies of $[\sigma] \to \sigma$. Thus, in order to assign a type to $\underline{n}t$, n typing derivations assigning $[\sigma] \to \sigma$ to t must be provided, exactly like in a type derivation for u. At the same time, the outermost application $\underline{n}t$ vanishes with the reduction $\underline{n}t \to_\beta u$, so the typing derivation for u is smaller than that for $\underline{n}t$.

In the idempotent case, on the other hand, a type [3] for $\underline{n}$ is an instance of $\{\{\sigma\} \to \sigma\} \to \{\sigma\} \to \sigma$, and the typing derivations for u may be hugely bigger than those for $\underline{n}t$, the former requiring n sub-derivations for t, the latter just one. That's why, for idempotent intersection type systems, the proof of the result above cannot be *combinatorial*, and is typically based on the *reducibility* argument [31,17,25].

This shift of perspective goes beyond lowering the logical complexity of the proof: the quantitative information provided by typing derivations in the non-idempotent setting unveils interesting relations between typings (static) and reductions (dynamic) of λ-terms. For instance, in [15], a correspondence between the size of a typing derivation for t and the number of steps taken by the Krivine machine to reduce t is presented, and in [5] it is shown how to compute the length of the longest β-reduction sequence starting from any typable strongly normalizing λ-term.

In this paper, we provide a characterization of strongly normalizing λ-terms via a typing system based on non-idempotent intersection types. The structure of the proof is the following:

- We define the K-calculus, reminiscent of Klop's I-calculus [24], where terms are defined by enriching λ-terms with a *memory* operator, β-reduction is split into two different non-erasing reductions, and terms are considered modulo an equivalence

[1] More precisely, it requires exactly as many sub-derivations for v as the number of *typed* occurrences of x in u.

[2] Non-idempotent intersections are denoted by multisets, *e.g.* $[\tau, \tau, \sigma]$ stands for $\tau \wedge \tau \wedge \sigma$.

[3] Idempotent intersections are denoted by sets, *e.g.* $\{\tau, \sigma\}$ stands for $\tau \wedge \sigma$.