ADVANCED COURSE:
The Complexity of β-Reduction

Teacher

Name: Beniamino Accattoli (beniamino.accattoli@inria.fr).

Affiliation: INRIA & Lix, École Polytechnique.

Homepage: `https://sites.google.com/site/beniaminoaccattoli/`

Abstract

The λ-calculus is the computational model behind functional programming languages and proof assistants, as well as a proof formalism for intuitionistic logic. Its dynamics is based on a single rewriting rule, *β-reduction*, whose study, historically, introduced most of the key concepts of rewriting theory.

Along the decades the λ-calculus has been studied from a multitude of point of view. Yet, the understanding of its cost models and its connections to complexity theory were very limited [BG95, AM98, SGM02, DLM08], untill very recently. The reason is that the abstract, higher-order character of the λ-calculus comes at a price, known as *size-explosion*: there are malicious programs that in a linear number of β-steps produce an exponential output, thus apparently forbidding to take the number of β-steps as a reasonable cost model.

In the last few years such a problem has been studied systematically. Notably, Accattoli and Dal Lago have shown that the number of (leftmost) β-steps is, suprisingly, a reasonable cost model, by relying on a simulation of λ-calculus in a calculus with a new sophisticated form of sharing [AL14]. Such a result is the apex of an emerging quantitative theory of β-reduction [Acc12, ABKL14, ABM14, AC14, AC15, ABM15, Acc16, AG16], based on a fine-grained understanding of the rewriting theory of the λ-calculus, coming from its linear logic interpretation.

This course will gently introduce the study of cost models and size-explosion, as well as the basic concepts of the quantitative theory. It will focus on the underlying rewriting principles (the standardisation theorem) and on the complexity analyses of abstract machines for dialects of the λ-calculus of increasing difficulty.

Tentative Outline

We show a preliminary organization of the two 90 minutes classes of the course. Each class is organized in three parts, half an hour each:

1. *First Class*: the first two parts are mostly frontal teaching, while the third part is interactive
   (a) *Overview*: examples of size-explosion, survey of the state of the art, (in)efficiency and cost models, Lévy's optimality is unreasonable;
   (b) *Theoretical Principles*: standardisation theorem, explicit substitutions, subterm property;
   (c) *Case Study*: the weak head call-by-name λ-calculus, via the interactive (between the teacher and the audience) complexity analysis of Krivine Abstract Machine.

2. *Second Class*: the three parts are all supposed to be interactive. Interactivity has priority over the amount of material, so it may be that less is taught.

(a) *More Efficient Evaluation Strategies*: definition, abstract machines, and complexity analyses of the weak call-by-value and the weak call-by-need λ-calculus (adapting the case study of the first class);

(b) *Creations and Useful Sharing*: redex creations in the λ-calculus, useless and useful replacements, unfoldings

(c) *Towards the Open and the Strong λ-Calculus*: hints on how to obtain reasonable abstract machines for open and strong λ-calculi. Time for questions and free discussions.

Expected Level and Prerequisites

The students are supposed to know the basics of λ-calculus or functional programming. No special knowledge of complexity theory is required (apart from the basic notion of asymptotic complexity), nor of linear logic. Basic knowledge of rewriting notions (such as confluence) is welcome.

References

[ABKL14] Beniamino Accattoli, Eduardo Bonelli, Delia Kesner, and Carlos Lombardi. A nonstandard standardization theorem. In *POPL*, pages 659–670, 2014.

[ABM14] Beniamino Accattoli, Pablo Barenbaum, and Damiano Mazza. Distilling abstract machines. In *ICFP 2014*, pages 363–376, 2014.

[ABM15] Beniamino Accattoli, Pablo Barenbaum, and Damiano Mazza. A strong distillery. In *APLAS 2015*, pages 231–250, 2015.

[AC14] Beniamino Accattoli and Claudio Sacerdoti Coen. On the value of variables. In *WoLLIC 2014*, pages 36–50, 2014.

[AC15] Beniamino Accattoli and Claudio Sacerdoti Coen. On the relative usefulness of fireballs. In *LICS 2015*, pages 141–155, 2015.

[Acc12] Beniamino Accattoli. An abstract factorization theorem for explicit substitutions. In *RTA*, pages 6–21, 2012.

[Acc16] Beniamino Accattoli. The useful mam, a reasonable implementation of the strong λ-calculus. In *WoLLIC 2016*, pages 1–21, 2016.

[AG16] Beniamino Accattoli and Giulio Guerrieri. Open call-by-value. In *APLAS 2016*, pages 206–226, 2016.

[AL14] Beniamino Accattoli and Ugo Dal Lago. Beta reduction is invariant, indeed. In *CSL-LICS '14*, pages 8:1–8:10, 2014.

[AM98] Andrea Asperti and Harry G. Mairson. Parallel beta reduction is not elementary recursive. In *POPL*, pages 303–315, 1998.

[BG95] Guy E. Blelloch and John Greiner. Parallelism in sequential functional languages. In *FPCA*, pages 226–237, 1995.

[DLM08] Ugo Dal Lago and Simone Martini. The weak lambda calculus as a reasonable machine. *Theor. Comput. Sci.*, 398(1-3):32–50, 2008.

[SGM02] David Sands, Jörgen Gustavsson, and Andrew Moran. Lambda calculi and linear speedups. In *The Essence of Computation*, pages 60–84, 2002.