Theoretical Computer Science 412 (2011) 1884–1902

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

A semantic measure of the execution time in linear logic

D. de Carvalho [a,*], M. Pagani [a], L. Tortora de Falco [b]

[a] *Laboratoire d'Informatique de Paris Nord, UMR CNRS 7030, France*

[b] *Dipartimento di Filosofia – Università Roma Tre, Italy*

ARTICLE INFO

Keywords:
Linear logic
Denotational semantics
Computational complexity

ABSTRACT

We give a semantic account of the execution time (i.e. the number of cut elimination steps leading to the normal form) of an untyped *MELL* net. We first prove that: (1) a net is head-normalizable (i.e. normalizable at depth 0) if and only if its interpretation in the multiset based relational semantics is not empty and (2) a net is normalizable if and only if its *exhaustive* interpretation (a suitable restriction of its interpretation) is not empty. We then give a semantic measure of execution time: we prove that we can compute the number of cut elimination steps leading to a cut free normal form of the net obtained by connecting two cut free nets by means of a cut-link, from the interpretations of the two cut free nets. These results are inspired by similar ones obtained by the first author for the untyped lambda-calculus.

1. Introduction

Right from the start, Linear Logic (LL, [12]) appeared as a potential logical tool to study computational complexity. The logical status given by the exponentials (the new connectives of LL) to the operations of erasing and copying (corresponding to the structural rules of intuitionistic and classical logic) shed a new light on the duplication process responsible for the "explosion" of the size and time during the cut elimination procedure. This is witnessed by the contribution given by LL to the wide research area called Implicit Computational Complexity: a true breakthrough with this respect is Girard's Light Linear Logic (LLL, [13]). A careful handling of LL's exponentials allows the author to keep enough control on the duplication process, and to prove that a function f is representable in LLL if and only if f is polytime.

One of the main questions arisen from [13] is the quest of a denotational semantics suitable for light systems (a semantics of proofs in logical terms, or more generally a model). Among the main attempts in this direction we can quote on the one hand [18,1], where the structures (games, coherent spaces) associated with logical formulas[1] are modified so that the principles valid in LL but not in the chosen light system do not hold in the semantics, and on the other hand [17] which deals with a property of the elements of the structures (the interpretations of proofs) characterizing those elements which *can* interpret proofs with bounded complexity.

A different approach to the semantics of bounded time complexity is possible: the basic idea is to measure by semantic means the execution of any program, regardless of its computational complexity. The aim is to compare different computational behaviors and to learn afterwards something on the very nature of bounded time complexity. This line of research springs out from the quantitative semantics of the untyped λ-calculus, interpreting terms as power series. The work by Ehrhard and Regnier [10,11] relates the powers appearing in the interpretation of a λ-term with the number of steps needed by the so-called Krivine machine to evaluate the head-normal form of the λ-term (if any). Following this approach, in [5,6] one of the authors of the present paper could compute the execution time of an untyped λ-term from its interpretation in the Kleisli category of the comonad associated with the finite multisets functor on the category **Rel** of sets

* Corresponding author.
E-mail addresses: carvalho@lipn.univ-paris13.fr (D. de Carvalho), pagani@lipn.univ-paris13.fr (M. Pagani), tortora@uniroma3.it (L. Tortora de Falco).

[1] The basic pattern of denotational semantics is to associate with every formula an object of some category and with every proof of the formula a morphism of this category called the interpretation of the proof.

0304-3975/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2010.12.017

and relations. Such an interpretation is the same as the interpretation of the net translating the λ-term in the multiset based relational model of linear logic. The execution time is measured here in terms of elementary steps of the Krivine machine. Also, [5,6] give a precise relation between an intersection type system introduced by [2] and experiments in the multiset based relational model. Experiments are a tool introduced by Girard in [12] allowing to compute the interpretation of proofs pointwise. An experiment corresponds to a type derivation and the result of an experiment corresponds to a type.[2]

We apply here this approach to Multiplicative and Exponential Linear Logic (MELL), and we show how it is possible to compute the number of steps of cut elimination by semantic means (notice that our measure being the number of cut elimination steps, here is a first difference with [5,6] where Krivine's machine was used to measure execution time). Linear Logic offers a very sharp way to study Gentzen's cut elimination by representing proofs as graphs with boxes, called *proof-nets* [12]. The peculiarity of proof-nets is to reduce the number of commutative cut elimination steps, which instead abounded in sequent calculi. If π' is a proof-net obtained by applying some steps of cut elimination to π, the main property of any model is that the interpretation $[\![\pi]\!]$ of π is the same as the interpretation $[\![\pi']\!]$ of π', so that from $[\![\pi]\!]$ it is clearly impossible to determine the number of steps leading from π to π'. Nevertheless, if we consider two cut free proof-nets π_1 and π_2 connected by means of a cut-link, we can wonder:

(1) is it the case that the thus obtained net can be reduced to a cut free one?
(2) if the answer to the previous question is positive, what is the number of cut reduction steps leading from the net with cut to a cut free one?

The main point of the paper is to show that it is possible to answer both these questions by only referring to $[\![\pi_1]\!]$ and $[\![\pi_2]\!]$.[3]

The first question makes sense only in an untyped framework (in the typed case, cut elimination is strongly normalizing; see [12]), and indeed Section 2.1 is devoted to define an untyped version of Girard's proof-nets, based on previous works, mainly [3,21,17,20]. Terui [23] also introduced a calculus corresponding to an untyped and intuitionistic version of proof-nets of Light Affine Logic and [4] addressed the problem of characterizing the (head-)normalizable nets in this restricted setting. We shift here from the intuitionistic to the classical framework. Let us mention here that to improve readability we chose to state and prove our results for proof-nets (i.e. logically correct proof-structures), but correctness (in our framework Definition 3) is rarely used (see also the concluding remarks, Section 6). The cut elimination procedure we define is similar to λ-calculus β-reduction, in the sense that the exponential step (the step (!/?) of Definition 6) is more similar to a step of β-reduction than it usually is. This is essential to prove our results (see the discussion on Fig. 5).

We consider in the paper two reduction strategies: head reduction and stratified reduction. The first one consists in reducing the cuts at depth 0 and stop. The second one consists in reducing a cut only when there exists no cut with (strictly) smaller depth. These reduction strategies extend the head (resp. leftmost) reduction of λ-calculus.

We mention the recent papers [14] and [15], where the complexity of linear logic cut elimination is analysed by means of context and game semantics. It is very likely that our approach and those of [14,15] are closely related. A fine analysis of this relation should help to clarify the correspondences between relational and game semantics.

Section 2 is devoted to define our version of proof-net (Section 2.1) and the model allowing to measure the number of cut elimination steps (Section 2.2). In Section 3, we show how experiments provide a counter for head and stratified reduction steps (Lemmas 17 and 20). In Section 4 we answer question (1), and in Section 5 we answer question (2).

Let us conclude with a little remark. In [22], the question of injectivity for the relational and coherent semantics of LL is addressed: is it the case that for π_1 and π_2 cut free, from $[\![\pi_1]\!] = [\![\pi_2]\!]$ one can deduce $\pi_1 = \pi_2$? It is conjectured that relational semantics is injective for MELL, and there is still no answer to this question. Given π_1 and π_2, we do not know how to compute the normal form of the net obtained by connecting π_1 and π_2 by means of a cut-link from $[\![\pi_1]\!]$ and $[\![\pi_2]\!]$. The present paper shows that from $[\![\pi_1]\!]$ and $[\![\pi_2]\!]$ we can at least compute the number of cut elimination steps leading to the normal form.

2. Preliminaries

We introduce the syntax and the model for which we prove our results: the untyped nets and their interpretation in the category **Rel** of sets and relations.

2.1. Untyped nets

After their introduction by Girard in [12], proof-nets have been extensively studied and used as a proof-theoretical tool for several purposes. All this work led to many improvements of the original notion introduced by Girard.
We use here an untyped version of Girard's proof-nets. Danos and Regnier [3,21] introduced and studied "pure proof-nets" that is the exact notion of proof-net corresponding to pure λ-calculus. There has been no real need for a different notion of untyped proof-net until Girard's work on Light Linear Logic [13]: Terui [23] introduces a "light" untyped λ-calculus

[2] The intersection type system considered in [5,6] lacks idempotency and this fact was crucial in that work. In the present paper, this corresponds to the fact that we use multisets for interpreting exponentials and not sets as in the set based coherent semantics. The use of multisets is essential in our work too.

[3] The questions (and the answers) are more general than it seems: every proof-net with cuts is the reduct of some proof-net obtained by cutting two cut free proof-nets (Proposition 34).