

Interaction Graphs and Complexity

Thomas Seiller*

IHÉS, Le Bois-Marie, 35 route de Chartres, 91440 Bures-sur-Yvette, France

Abstract

The aim of Implicit Computational Complexity is to study algorithmic complexity only in terms of restrictions of languages and computational principles. Based on previous work about realizability models for linear logic, we propose a new approach where we consider semantical restrictions instead of syntactical ones. This leads to a hierarchy of models mirroring subtle distinctions about computational principles. As an illustration of the method, we explain how to obtain characterizations of the set of regular languages and the set of logarithmic space predicates.

1 Introduction

Implicit Computational Complexity (ICC) and Linear Logic. Amongst the different approaches to ICC, a number of results have been obtained recently by considering syntactic restrictions of linear logic. Linear logic is obtained through a decomposition of the intuitionistic implication $\Rightarrow$ in two independent operations: the exponentiation $!A$ which allows duplication, and a linear implication, written $\multimap$, which produces from a proof of A a proof of B , *consuming A in the process*. The implication $A \Rightarrow B$ is then obtained as the formula $!A \multimap B$. The re-use of resources being associated to the exponential connective $!$, one may then modify the rules of this connective to obtain restrictions of linear logic which have a computational interest. This is how constrained linear logic systems, such as BLL [GSS92], SLL [Laf04] and LLL [Gir95] which characterize PTime, or ELL [DJ03] which characterizes Elementary time are obtained.

Geometry of Interaction. This program of research was initiated by Girard soon after his discovery of linear logic [Gir89b]. It defines and studies a notion of realizability models for linear logic, what we will call GoI models. This field of research already lead to applications in the field of computational complexity. For instance, it is by using the first GoI model [Gir89a] that Abadi, Gonthier and Lévy [GAL92] showed the optimality of Lamping's reduction in lambda-calculus [Lam90]. The same model was also applied in implicit computational complexity [BP01], and was the main inspiration behind dal Lago's context semantics [Lag09]. The more recent model [Gir11] lead to characterizations of logarithmic space computation [AS12, AS13].

*This work was partially supported by the ANR-10-BLAN-0213 LOGOI.

Interaction Graphs. The author introduced, in previous work[Sei12b, Sei12a, Sei12c], a family of models which generalizes and axiomatizes the notion a GoI model. This systematic approach, extended by considering a generalization of graphs named *graphings*, allows one to consider a very rich hierarchy of models parametrized by two monoids: the *weight monoid* and a monoid of measurable maps called a *microcosm*. These two parameters allow for fine distinctions on the computational principles allowed in the model. This can be illustrated by the obtention of models of linear logic where the type $!\mathbf{Nat}_2 \multimap \mathbf{Bool}$ corresponds respectively to regular languages and logarithmic space predicates.

2 Graphings

Definition 1. Let $(X, \mathcal{B}, \lambda)$ be a measured space. We denote by $\mathcal{M}(X)$ the set of non-singular transformations¹ $X \rightarrow X$. A *microcosm* of the measured space X is a subset $\mathfrak{m}$ of $\mathcal{M}(X)$ which is closed under composition and contains the identity.

In the following, we will consider a notion of graphing depending on a *weight-monoid* Ω , i.e. a monoid $(\Omega, \cdot, 1)$ which contains the possible weights of the edges.

Definition 2 (Graphings). Let $\mathfrak{m}$ be a microcosm of a measured space $(X, \mathcal{B}, \lambda)$ and V^F a measurable subset of X . A Ω -*weighted graphing* in $\mathfrak{m}$ of carrier V^F is a countable family $F = \{(\omega_e^F, \phi_e^F : S_e^F \rightarrow T_e^F)\}_{e \in E^F}$, where, for all $e \in E^F$ (the set of edges):

- ω_e^F is an element of Ω , the *weight* of the edge e ;
- $S_e^F \subset V^F$ is a measurable set, the *source* of the edge e ;
- $T_e^F = \phi_e^F(S_e^F) \subset V^F$ is a measurable set, the *target* of the edge e ;
- ϕ_e^F is the restriction of an element of $\mathfrak{m}$ to S_e^F , the *realization* of the edge e .

The author showed in earlier work [Sei12a] how one can construct models of multiplicative-additive linear logic where proofs are interpreted as graphs. This construction relied on a single property, called the *trefoil property*, which relates two simple notions:

- the *execution* $F :: G$ of two graphs, a graph defined as a set of paths;
- the *measurement* $\llbracket F, G \rrbracket_m$, a real number computed from a set of cycles.

These constructions can be extended to the more general framework where proofs are interpreted as graphings. Indeed, the notions of paths and cycles in a graphings are quite natural, and from two graphings F, G in a microcosm $\mathfrak{m}$ one can define its execution $F :: G$ which is again a graphing in $\mathfrak{m}^2$. A more involved argument then shows that the trefoil property holds for a family of measurements $\llbracket \cdot, \cdot \rrbracket_m$, where $m : \Omega \rightarrow \mathbf{R}_{\geq 0} \cup \{\infty\}$ is any measurable map. These results are obtained as a generalization of constructions considered in the author's thesis³.

Theorem 3. *Let Ω be a monoid and $\mathfrak{m}$ a microcosm. The set of Ω -weighted graphings in $\mathfrak{m}$ yields a model, denoted by $\mathbb{M}[\Omega, \mathfrak{m}]$, of multiplicative-additive linear logic.*

¹A non-singular transformation $f : X \rightarrow X$ is a measurable map which preserves the sets of null measure, i.e. $\lambda(f(A)) = 0$ if and only if $\lambda(A) = 0$.

²As a consequence, a microcosm is a closed world for the execution of programs.

³In the cited work, the results were stated in the particular case of the microcosm of measure-preserving maps on the real line.

In some cases, one can define exponential modalities in the model. A *perennial structure* on a microcosm $m \in \mathcal{M}(X)$ consists in the following data:

- for all integer n , a measurable partition $(X_i^n)_{i=0}^n$ of X ;
- for all integer n , elements $\tau_{i,j} \in m$ ($0 \leq i, j \leq n$) such that $\tau_{i,j}(X_i^n) = X_j^n$;

Theorem 4. *Let Ω be a monoid and m a microcosm. A perennial structure on m defines a modality $!$ in the model. Moreover, the model validates the contraction principle $!A \multimap !A \otimes !A$.*

Given two microcosms $m \subset n$, it is clear that a graphing in m is in particular a graphing in n . This inclusion actually extends to an embedding of the model $\mathbb{M}[\Omega, m]$ into $\mathbb{M}[\Omega, n]$ which preserves all logical operations. In the same way, an inclusion of monoids $\Omega \subset \Gamma$ yields an embedding of the the model $\mathbb{M}[\Omega, m]$ into $\mathbb{M}[\Gamma, m]$. Moreover, given two microcosms m and n , one can define the *smallest common extension* $m + n$ as the compositional closure of the set $m \cup n$. The model $\mathbb{M}[\Omega, m + n]$ then contains both models $\mathbb{M}[\Omega, m]$ and $\mathbb{M}[\Omega, n]$ through the embedding just mentioned.

3 Models and Complexity Classes

It is important to remark that in general the model $\mathbb{M}[\Omega, m]$ does not validate the functorial promotion rule. This turns out to be interesting: we obtain in this way models of linear logic where the exponentials are weaker than what is obtained through syntactic consideration in systems like BLL, SLL, etc.

In the models of linear logic we described, one can easily define the type of binary lists $\mathbf{Nat}_2$. The definition of the representation of lists is explained thoroughly in previous papers [AS12, AS13]. Then, for each natural number n , there exists an element N_n in the type $!\mathbf{Nat}_2$ which represents it. We say that a given element F of type $!\mathbf{Nat}_2 \multimap \mathbf{Bool}$ *accepts* the integer n when $F :: N_n$ is equal to the distinguished element $\text{true} \in \mathbf{Bool}$. The *language* accepted by such an element F is then defined as $[F] = \{n \in \mathbf{N} \mid \phi :: N_n = \text{true}\}$

Definition 5. Let Ω be a monoid, m a microcosm and $\mathcal{L}$ a set of languages. We say the type $!\mathbf{Nat}_2 \multimap \mathbf{Bool}$ *characterizes the set $\mathcal{L}$* if the set $\{[F] \mid F \in !\mathbf{Nat}_2 \multimap \mathbf{Bool}\}$ is equal to $\mathcal{L}$.

We now consider the measured space $\mathbf{N} \times [0, 1]^{\mathbf{N}}$ endowed with the product of the counting measure on $\mathbf{N}$ and the Lebesgue measure on the Hilbert cube $[0, 1]^{\mathbf{N}}$. We fix ρ any measurable bijection from $[0, 1]^2$ to $[0, 1]$ and we write $\bar{\rho}_i : [0, 1] \rightarrow [0, 1]$ ($i = 1, 2$) the projections of its inverse. We define the following microcosms:

- t is the monoid of translations $T_k : (n, x) \mapsto (n + k, x)$;
- p is the monoid of finite permutations $T_\sigma : (n, (x_i)_{i \in \mathbf{N}}) \mapsto (n, (x_{\sigma(i)})_{i \in \mathbf{N}})$
- c is the monoid generated by the two measurable transformations ψ and ψ^* defined as follows:
 - $\psi : (n, (x_0, x_1, \dots, x_n, \dots)) \mapsto (n, (\rho(x_0, x_1), x_2, \dots, x_n, \dots))$
 - $\psi^* : (n, (x_0, x_1, \dots, x_n, \dots)) \mapsto (n, (\bar{\rho}_1(x_0), \bar{\rho}_2(x_0), x_1, \dots, x_n, \dots))$

The intuition is that a microcosm m represents the set of computational principles available to write programs in the model. The operation $+$ thus extends the set of principles at disposal, increasing expressivity. As a consequence, the set of languages characterized by the type $!\mathbf{Nat}_2 \multimap \mathbf{Bool}$ becomes larger and larger as we consider extensions of the microcosms. As an example, the microcosm t

corresponds to allowing oneself to compute with automata. Expanding this microcosm by p corresponds to the addition of multiple heads to the automata. This is illustrated by the following theorems.

Theorem 6. *In the model $\mathbb{M}[\{1\}, t]$, the type $!Nat_2 \multimap Bool$ characterizes the class of Regular languages.*

Theorem 7. *In the model $\mathbb{M}[\{1\}, t + p]$, the type $!Nat_2 \multimap Bool$ characterizes the class of LogSpace languages.*

The extension of $t + p$ along the microcosm c then allows one to construct stacks of pointers by using the maps ψ (push) and ψ^* (pop). This increases once again the expressivity of the model, as the number of pointers available is no longer a constant parameter of the machine.

One can show a soundness result for Elementary Linear Logic (ELL) with additive connectives in the model $\{\{1\}, t + p + c\}$. On the other hand, a recent result of Baillot [Bai11], state that $!Nat_2 \multimap !!B$ characterizes PTime predicates in Elementary Affine Linear Logic (EAL). This result, together with the fact that we have a map $!Bool \multimap Bool$ in our model, should lead to a proof of the following result:

Conjecture 1. *In the model $\mathbb{M}[\{1\}, t + p + c]$, the type $!Nat_2 \multimap Bool$ characterizes the class of PTime languages.*

Moreover, all the preceding results have non-deterministic analogues. Indeed, instead of considering elements of the type $!Nat_2 \multimap Bool$, one can consider elements of the type $!Nat_2 \multimap NB$, where NB is a specific type definable in the models, somehow a non-deterministic version of the booleans. The analogues of Theorems 6 and 7 then hold and a proof of the conjecture would imply that its non-deterministic version would also hold.

4 Conclusion and Perspectives

We exposed a new, semantic-based, approach to implicit computational complexity. We illustrated it with two examples, obtaining characterizations of the regular languages on one hand and of logarithmic space predicates on the other. This approach is very rich, and leads to a number of questions that will be the object of future work.

First, we will search for other characterizations of complexity classes. This can be obtained by studying extensions of the microcosms, as illustrated above, but also by modifying the weight monoid. As an example, one can consider the model $\mathbb{M}[G, t]$ where G is the free group over one element to allow oneself the use of a (unary) counter. Other extensions of this model with larger groups should lead to a characterization of context-free languages. Another question concerns the characterization of classes of functions and not only predicates, which should be possible following first results (in a syntactical approach) by Baillot [Bai11].

Another important aspect concerns the construction of types in the model. Types are defined through a notion of test. In the particular case of the model $\mathbb{M}[\{1\}, t + p]$, one can show that the number of tests defining the type $!Nat_2 \multimap Bool$ is finite. This definition of types by a set of tests is a very important issue when considering the tests defining the classes themselves. Indeed, this would

imply a definition of complexity classes by means of an existential formula, making proofs of non-membership easier. For this reason, we will work on defining and studying types corresponding to the classes characterized by elements of $\mathbf{!Nat}_2 \multimap \mathbf{Bool}$.

References

- [AS12] Clément Aubert and Thomas Seiller. Characterizing co-nl by a group action. *CoRR*, abs/1209.3422, 2012.
- [AS13] Clément Aubert and Thomas Seiller. Logarithmic space and permutations. *CoRR*, abs/1301.3189, 2013.
- [Bai11] Patrick Baillot. Elementary linear logic revisited for polynomial time and an exponential time hierarchy. In Hongseok Yang, editor, *APLAS*, volume 7078 of *Lecture Notes in Computer Science*, pages 337–352. Springer, 2011.
- [BP01] Patrick Baillot and Marco Pedicini. Elementary complexity and geometry of interaction. *Fundamenta Informaticae*, 45(1-2):1–31, 2001.
- [DJ03] Vincent Danos and Jean-Baptiste Joinet. Linear logic & elementary time. *Information and Computation*, 183(1):123–137, 2003.
- [GAL92] Georges Gonthier, Martín Abadi, and Jean-Jacques Lévy. The geometry of optimal lambda reduction. In Ravi Sethi, editor, *POPL*, pages 15–26. ACM Press, 1992.
- [Gir89a] Jean-Yves Girard. Geometry of interaction I: Interpretation of system F. In *In Proc. Logic Colloquium 88*, 1989.
- [Gir89b] Jean-Yves Girard. Towards a geometry of interaction. In *Proceedings of the AMS Conference on Categories, Logic and Computer Science*, 1989.
- [Gir95] Jean-Yves Girard. Light linear logic. In *Selected Papers from the International Workshop on Logical and Computational Complexity*, LCC ’94, pages 145–176, London, UK, UK, 1995. Springer-Verlag.
- [Gir11] Jean-Yves Girard. Geometry of interaction V: Logic in the hyperfinite factor. *Theoretical Computer Science*, 412:1860–1883, 2011.
- [GSS92] Jean-Yves Girard, Andre Scedrov, and Philip J. Scott. Bounded linear logic: a modular approach to polynomial-time computability. *Theor. Comput. Sci.*, 97(1):1–66, April 1992.
- [Laf04] Yves Lafont. Soft linear logic and polynomial time. *Theor. Comput. Sci.*, 318(1-2):163–180, June 2004.
- [Lag09] Ugo Dal Lago. The geometry of linear higher-order recursion. *ACM Trans. Comput. Logic*, 10(2):8:1–8:38, March 2009.
- [Lam90] John Lamping. An algorithm for optimal lambda calculus reduction. In Frances E. Allen, editor, *POPL*, pages 16–30. ACM Press, 1990.
- [Sei12a] Thomas Seiller. Interaction graphs: Additives. *CoRR*, abs/1205.6557, 2012.
- [Sei12b] Thomas Seiller. Interaction graphs: Multiplicatives. *Annals of Pure and Applied Logic*, 163:1808–1837, December 2012.
- [Sei12c] Thomas Seiller. *Logique dans le facteur hyperfini : géométrie de l’interaction et complexité*. PhD thesis, Université Aix-Marseille, 2012.