Non-uniform polynomial time via non-wellfounded parsimonious proofs

Matteo Acclavio Department of Computer Science, University of Southern Denmark, macclavio@gmail.com
Gianluca Curzi Department of Computer Science, University of Birmingham g.curzi@bham.ac.uk
Giulio Guerrieri LIS, Aix Marseille Université giulio.guerrieri@lis-lab.fr

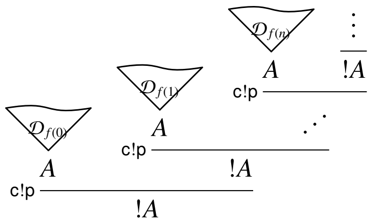
Linear logic (**LL**) [9] is a refinement of both classical and intuitionistic logic allowing control over computational resources. This is obtained by having a strong discipline on the use of resources in proofs thanks to the use of the *exponential* modality (denoted by !), which marks the distinction between those assumptions that can be used linearly, that is, exactly once, and those ones that are reusable at will. In the Curry-Howard interpretation, the exponential modality introduces non-linearity in functional programs: a proof of the linear implication $!A \multimap B$ is interpreted as a program returning an output of type B using an arbitrary (but finite) number of times an input of type A.

Thanks to its computational features, linear logic has established itself as an important tool for *Implicit Computational Complexity* (ICC), the branch of computational complexity studying languages or calculi able to capture the inherent principles of bounded computation without depending on specific machine models or explicit resource bounds. In particular, several variants of second-order linear logic called *light logics* have been proposed to capture complexity classes: examples are *soft linear logic* (**SLL**) [11] or *light linear logic* (**LLL**) [10] for **FP** (the class of polynomial time computable functions), and *elementary linear logic* (**ELL**) [5] for **FELEMENTARY** (the class of elementary time computable functions).

Continuing this tradition, in [14] Mazza and Terui introduced *parsimonious logic*, $\mathbf{nuPL}_{\forall\ell}$, a lambda calculus with polymorphic types inspired by linear logic that characterises the complexity class $\mathbf{P}/\mathsf{poly}$ (the class of problems decidable in non-uniform polynomial time)[1]In this system, the exponential modality satisfies the so-called Milner's law $!A \simeq A \otimes !A$. According to the Curry-Howard interpretation, this law allows us to interpret a formula $!A$ as the type of streams over data of type A. Therefore, the linear implications $A \otimes !A \multimap !A$ (*co-absorption*) and $!A \multimap A \otimes !A$ (*absorption*) can be respectively interpreted as the push and the pop operations on streams. In $\mathbf{nuPL}_{\forall\ell}$, non-uniformity is introduced by the typing rule $!\mathsf{I}$, which takes a finite set of proofs $\mathcal{D}_1, \dots, \mathcal{D}_n$ of A and a (possibly non-recursive) function $f : \mathbb{N} \to \{1, \dots, n\}$ as premises, and constructs a proof of $!A$ modelling the stream $\mathcal{D}_{f(0)} :: \mathcal{D}_{f(1)} :: \mathcal{D}_{f(2)} :: \dots$. Specifically, the typing rule $!\mathsf{I}$ allows the encoding of *advices* for Turing machines, the crucial step to show completeness for $\mathbf{P}/\mathsf{poly}$. On the other hand, polynomial step cut-elimination is guaranteed thanks to "parsimony", which invalids the implications $!A \multimap !!A$ (*digging*) and $!A \multimap !A \otimes !A$ (*contraction*).

[1]Formally, $\mathbf{P}/\mathsf{poly}$ can be defined as the class of problems decidable by families of circuits with polynomial size or, equivalently, as the class of problems decidable in polynomial time by a Turing machine with polynomial *advice*, that is, an extra input whose size depends on the length of the input, but not on the input itself.

Contribution In this talk we present an ongoing work exploring a different approach to $\mathbf{P}/\mathsf{poly}$ based on *Cyclic Implicit Complexity*, the study of ICC in the context of non-wellfounded proof theory [3, 4]. Specifically, the typing rule $!\mathsf{I}$ parametrised by a (possibly non-recursive) function $f : \mathbb{N} \to \{1, \ldots, n\}$ and defining the stream $\mathcal{D}_{f(0)} :: \mathcal{D}_{f(1)} :: \mathcal{D}_{f(2)} :: \ldots$ will be represented by a non-wellfounded proof of the following form:



essentially by "unpacking" $!\mathsf{I}$ into an infinite proof iterating a more primitive rule called *conditional promotion* ($\mathsf{c!p}$).

The resulting system of non-wellfounded proofs for parsimonious logic, called nuPLL_2^∞, introduces fallacious reasoning. Logical consistency is then recovered by adapting a standard global condition, called *progressiveness criterion*, which relies on threads of exponential formulas occurring in the infinite branches of a derivation tree. In particular, progressiveness forces a computational interpretation of the modalities $!$ and $?$ (i.e. the dual of $!$), in terms of greatest and least fixed points respectively. Note that definitions of exponentials based on fixed points have been proposed in [2] by defining $!A := \nu\alpha.(\mathbf{1} \,\&\, A \,\&\, (\alpha \otimes \alpha))$ and $?A := \mu\alpha.(\bot \oplus A \oplus (\alpha \parr \alpha))$, where ν and μ are the greatest and the least fixed point operator respectively. However, as shown in [6], such an encoding does not give rise to a Seely category, which is essential to model linear logic.

We discuss an alternative technique to prove cut-elimination for nuPLL_2^∞ relying on infinitary rewriting techniques (see, e.g. [7]), but avoiding the use of the multicut rule, as opposed to [1]. Then, we show that nuPLL_2^∞ captures the class $\mathbf{FP}/\mathsf{poly}$ (the class of functions computable in non-uniform polynomial time). To this end, we establish a polynomial "modulus of continuity" for cut elimination (see e.g. [13]), from which we infer soundness for $\mathbf{FP}/\mathsf{poly}$. This is one of the major technical results of the paper. Completeness is established via an encoding of polynomial time Turing machines with (polynomial) advice by adapting standard methods from [12, 8] to the setting of non-uniform computation. Along the way, we show that cPLL_2^∞, i.e. the restriction of nuPLL_2^∞ to proofs having a regular tree structure (known as *circular* proofs), captures precisely $\mathbf{FP}$.

We conclude by introducing a relational semantics for nuPLL_2^∞ and by analysing its interplay with cut elimination.

References

[1] David Baelde, Amina Doumane, and Alexis Saurin. Infinitary proof theory: the multiplicative additive case. In Jean-Marc Talbot and Laurent Regnier, editors, *25th EACSL Annual Conference on Computer Science Logic, CSL 2016, August 29 - September 1, 2016, Marseille, France*, volume 62 of *LIPIcs*, pages 42:1–

42:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.CSL.2016.42.

[2] David Baelde and Dale Miller. Least and greatest fixed points in linear logic. In Nachum Dershowitz and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning, 14th International Conference, LPAR 2007, Yerevan, Armenia, October 15-19, 2007, Proceedings*, volume 4790 of *Lecture Notes in Computer Science*, pages 92–106. Springer, 2007. doi:10.1007/978-3-540-75560-9_9.

[3] Gianluca Curzi and Anupam Das. Cyclic implicit complexity. In Christel Baier and Dana Fisman, editors, *LICS '22: 37th Annual ACM/IEEE Symposium on Logic in Computer Science, Haifa, Israel, August 2 - 5, 2022*, pages 19:1–19:13. ACM, 2022. doi:10.1145/3531130.3533340.

[4] Gianluca Curzi and Anupam Das. Non-uniform complexity via non-wellfounded proofs. In Bartek Klin and Elaine Pimentel, editors, *31st EACSL Annual Conference on Computer Science Logic, CSL 2023, February 13-16, 2023, Warsaw, Poland*, volume 252 of *LIPIcs*, pages 16:1–16:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.CSL.2023.16.

[5] Vincent Danos and Jean-Baptiste Joinet. Linear logic and elementary time. *Inf. Comput.*, 183(1):123–137, 2003. doi:10.1016/S0890-5401(03)00010-5.

[6] Thomas Ehrhard and Farzad Jafar-Rahmani. On the denotational semantics of linear logic with least and greatest fixed points of formulas. *CoRR*, abs/1906.05593, 2019. URL: http://arxiv.org/abs/1906.05593, arXiv:1906.05593.

[7] Jérôme Fortier and Luigi Santocanale. Cuts for circular proofs: semantics and cut-elimination. In Simona Ronchi Della Rocca, editor, *Computer Science Logic 2013 (CSL 2013), CSL 2013, September 2-5, 2013, Torino, Italy*, volume 23 of *LIPIcs*, pages 248–262. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2013. doi:10.4230/LIPIcs.CSL.2013.248.

[8] Marco Gaboardi and Simona Ronchi Della Rocca. From light logics to type assignments: a case study. *Log. J. IGPL*, 17(5):499–530, 2009. doi:10.1093/jigpal/jzp019.

[9] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50(1):1–101, 1987. doi:10.1016/0304-3975(87)90045-4.

[10] Jean-Yves Girard. Light linear logic. In Daniel Leivant, editor, *Logical and Computational Complexity. Selected Papers. Logic and Computational Complexity, International Workshop LCC '94, Indianapolis, Indiana, USA, 13-16 October 1994*, volume 960 of *Lecture Notes in Computer Science*, pages 145–176. Springer, 1994. doi:10.1007/3-540-60178-3_83.

[11] Yves Lafont. Soft linear logic and polynomial time. *Theoretical Computer Science*, 318(1):163–180, 2004. Implicit Computational Complexity. doi:10.1016/j.tcs.2003.10.018.

[12] Harry G. Mairson and Kazushige Terui. On the computational complexity of cut-elimination in linear logic. In Carlo Blundo and Cosimo Laneve, editors, *Theoretical Computer Science, 8th Italian Conference, ICTCS 2003, Bertinoro, Italy, October 13-15, 2003, Proceedings*, volume 2841 of *Lecture Notes in Computer Science*, pages 23–36. Springer, 2003. doi:10.1007/978-3-540-45208-9_4.

[13] Damiano Mazza. Non-uniform polytime computation in the infinitary affine lambda-calculus. In Javier Esparza, Pierre Fraigniaud, Thore Husfeldt, and Elias Koutsoupias, editors, *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part II*, volume 8573 of *Lecture Notes in Computer Science*, pages 305–317. Springer, 2014. doi:10.1007/978-3-662-43951-7_26.

[14] Damiano Mazza. Simple parsimonious types and logarithmic space. In Stephan Kreutzer, editor, *24th EACSL Annual Conference on Computer Science Logic, CSL 2015, September 7-10, 2015, Berlin, Germany*, volume 41 of *LIPIcs*, pages 24–40. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015. doi:10.4230/LIPIcs.CSL.2015.24.