Non-uniform polynomial time and non-wellfounded parsimonious proofs

Matteo Acclavio
University of Southern Denmark
macclavio@gmail.com

Gianluca Curzi
University of Gothenburg
gianluca.curzi@gu.se

Giulio Guerrieri
Aix Marseille Université
giulio.guerrieri@gmail.com

Non-wellfounded proof theory studies proofs as possibly infinite (but finitely branching) trees, where logical consistency is maintained by means of global conditions called *progressing* (or *validity*) *criteria* (see, e.g., [13, 6]). In this setting, the so-called *regular* (also called *cyclic*) proofs receive a special attention, in that they admit a finite description given in terms of (possibly cyclic) directed graphs.

In joint work with Das [3, 4], the second author presented non-wellfounded proof systems characterising the complexity classes **P** (the problems decidable in polynomial time), **ELEMENTARY** (the problems decidable in elementary time), and $\mathbf{P}/\mathsf{poly}$ (the problems decidable in *non-uniform* polynomial time) [1]. These systems recast in a non-wellfounded proof-theoretic setting the principle of safe recursion introduced by the celebrated work of Bellantoni and Cook [2], a cornerstone of *Implicit Computational Complexity* (ICC). Intuitively, ICC studies characterisations of complexity classes without reference to a specific underlying machine model or to explicit bounds on computational resources, unlike conventional complexity theory.

In this talk we follow an alternative route to **P** and $\mathbf{P}/\mathsf{poly}$ based on *linear logic* [8]. Roughly, linear logic (LL) is a refinement of both classical and intuitionistic logic that allows a fine-grained control over computational resources, implemented via the so-called *exponential modalities* (denoted by ! and ?). Linear logic has inspired a variety of methods for taming complexity in the style of ICC. The key idea is to weaken the inference rules for the exponential modality ! to induce a bound on cut-elimination, hence reducing the computational strength of the system (see, e.g., [8, 9, 5]).

Continuing this tradition, in a series of papers [11, 12] Mazza introduced *parsimonious logic*, a variant of linear logic where the exponential modality ! satisfies Milner's law (i.e., $!A \multimapboth A \otimes !A$) and invalidates the implications $!A \multimap !!A$ (*digging*) and $!A \multimap !A \otimes !A$ (*contraction*). Computationally, this allows us to interpret the proof of a formula $!A$ as a *stream* over (a finite set of) proofs of A, i.e., as a greatest fixed point.

In [12], Mazza and Terui studied the system $\mathsf{nuPL}_{\forall\ell}$, a second-order extension of parsimonious logic that implements a proof-theoretic counterpart of the notion of *non-uniformity* from computational complexity, and proved that the system characterises $\mathbf{P}/\mathsf{poly}$. As a straightforward consequence of this result, the fully uniform version of this system, called $\mathsf{PL}_{\forall\ell}$, captures the class **P**.

In this talk we investigate non-wellfounded versions of the systems $\mathsf{PL}_{\forall\ell}$ and $\mathsf{nuPL}_{\forall\ell}$, that we denote rPLL_2^∞ and wrPLL_2^∞, where proof-theoretical non-uniformity is modelled by *weak regularity*, a relaxation of the regularity condition for non-wellfounded proofs. The proof-theory of the propositional formulation of rPLL_2^∞ and wrPLL_2^∞ has already been studied in a previous work by the present authors [1], where a continuous cut-elimination result for these systems is established.

As our main contribution, we prove that rPLL_2^∞ and wrPLL_2^∞ duly characterise the complexity classes **P** and $\mathbf{P}/\mathsf{poly}$, respectively. First, we show a *polynomial modulus of continuity* for cut-elimination, from

[1] More precisely, $\mathbf{P}/\mathsf{poly}$ is the class of functions computable in polynomial time by Turing machines with access to *polynomial advice* or, equivalently, decidable by non-uniform families of polynomial-size circuits.

Submitted to:
SOS 2007

which we infer that wrPLL_2^∞ is sound for $\mathbf{P}/\mathsf{poly}$ (and that rPLL_2^∞ is sound for $\mathbf{P}$). Completeness requires a series of intermediate steps. We first introduce the type system nuPTA_2, which implements some restricted form of oracle-based computation. Then we show an encoding of polynomial time Turing machines with (polynomial) advice in nuPTA_2, essentially by adapting standard methods from [10, 7] to the setting of non-uniform computation. This allows us to prove that nuPTA_2 is complete for $\mathbf{P}/\mathsf{poly}$. Thirdly, we define a translation from nuPTA_2 to $\mathsf{nuPL}_{\forall\ell}$. Finally, we show that computation over strings in $\mathsf{nuPL}_{\forall\ell}$ can be simulated within wrPLL_2^∞. We then apply a similar completeness argument for rPLL_2^∞.

On a technical side, we stress that wrPLL_2^∞ and rPLL_2^∞ are free of the so-called *co-absorption rule* from parsimonious logic, which expresses computationally the “push" operation on streams. Therefore, as a byproduct of our results, we show that co-absorption is not essential for establishing the characterisation theorems in [12].

References

[1] Matteo Acclavio, Gianluca Curzi & Giulio Guerrieri (2023): *Infinitary cut-elimination via finite approximations*. arXiv:2308.07789.

[2] Stephen Bellantoni & Stephen Cook (1992): *A New Recursion-Theoretic Characterization of the Polytime Functions (Extended Abstract)*. In: *Proceedings of the Twenty-Fourth Annual ACM Symposium on Theory of Computing*, STOC ’92, Association for Computing Machinery, New York, NY, USA, p. 283–293, doi:10.1145/129712.129740. Available at `https://doi.org/10.1145/129712.129740`.

[3] Gianluca Curzi & Anupam Das (2022): *Cyclic Implicit Complexity*. In: *Proceedings of the 37th Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS ’22, Association for Computing Machinery, New York, NY, USA, doi:10.1145/3531130.3533340. Available at `https://doi.org/10.1145/3531130.3533340`.

[4] Gianluca Curzi & Anupam Das (2023): *Non-Uniform Complexity via Non-Wellfounded Proofs*. In Bartek Klin & Elaine Pimentel, editors: *31st EACSL Annual Conference on Computer Science Logic, CSL 2023, February 13-16, 2023, Warsaw, Poland*, *LIPIcs* 252, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, pp. 16:1–16:18, doi:10.4230/LIPIcs.CSL.2023.16. Available at `https://doi.org/10.4230/LIPIcs.CSL.2023.16`.

[5] Vincent Danos & Jean-Baptiste Joinet (2003): *and elementary time*. *Inf. Comput.* 183(1), pp. 123–137, doi:10.1016/S0890-5401(03)00010-5. Available at `https://doi.org/10.1016/S0890-5401(03)00010-5`.

[6] Christian Dax, Martin Hofmann & Martin Lange (2006): *A proof system for the linear time μ-calculus*. In: *International Conference on Foundations of Software Technology and Theoretical Computer Science*, Springer, pp. 273–284.

[7] Marco Gaboardi & Simona Ronchi Della Rocca (2009): *From light logics to type assignments: A case study*. *Logic Journal of the IGPL* 17, doi:10.1093/jigpal/jzp019.

[8] Jean-Yves Girard (1998): *Light Linear Logic*. *Information and Computation* 143(2), pp. 175–204, doi:10.1006/inco.1998.2700. Available at `https://www.sciencedirect.com/science/article/pii/S0890540198927006`.

[9] Yves Lafont (2004): *Soft linear logic and polynomial time*. *Theor. Comput. Sci.* 318(1-2), pp. 163–180, doi:10.1016/j.tcs.2003.10.018. Available at `https://doi.org/10.1016/j.tcs.2003.10.018`.

[10] Harry G. Mairson & Kazushige Terui (2003): *On the Computational Complexity of Cut-Elimination in Linear Logic*. In: *Italian Conference on Theoretical Computer Science*.

[11] Damiano Mazza (2015): *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, *LIPIcs* 41, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, pp. 24–40, doi:10.4230/LIPIcs.CSL.2015.24. Available at `https://doi.org/10.4230/LIPIcs.CSL.2015.24`.

[12] Damiano Mazza & Kazushige Terui (2015): *Parsimonious Types and Non-uniform Computation*. In Magnús M. Halldórsson, Kazuo Iwama, Naoki Kobayashi & Bettina Speckmann, editors: *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part II*, *Lecture Notes in Computer Science* 9135, Springer, pp. 350–361, doi:10.1007/978-3-662-47666-6_28. Available at `https://doi.org/10.1007/978-3-662-47666-6_28`.

[13] Damian Niwiński & Igor Walukiewicz (1996): *Games for the μ-calculus*. *Theoretical Computer Science* 163(1-2), pp. 99–116.