A semantic conjecture on second-order MLL and its complexity consequences (work in progress)

Le Thanh Dung NGUYEN
Département d'informatique
École normale supérieure
Paris Sciences et Lettres
Paris, France
le.thanh.dung.nguyen@ens.fr

Thomas SEILLER
CNRS
LIPN – UMR 7030
Université Paris 13, Sorbonne Paris Cité
Paris, France
seiller@lipn.fr

Let us represent booleans as the type $\mathsf{Bool} = \forall X.X \otimes X \multimap X \otimes X$, and bitstrings using the stratified Church encoding $\mathsf{Str} = \forall X \mathsf{Str}[X]$, with $\mathsf{Str}[X] = !(X \multimap X) \multimap !(X \multimap X) \multimap !(X \multimap X)$. Then we have the following characterization of the exponential time hierarchy:

Theorem 1 (Baillot [1])**.** *A language can be recognized by a proof of* $!\mathsf{Str} \multimap !^{k+2}\mathsf{Bool}$ *in Intuitionistic Elementary Affine Logic (IEAL)* with recursive types *if and only if it belongs to* k-EXPTIME*. In particular, the languages decided by predicates of type* $!\mathsf{Str} \multimap !!\mathsf{Bool}$ *are exactly those in* PTIME*.*

Baillot's proof of extensional completeness crucially relies on type fixpoints. A natural question is whether they are actually necessary for this result, and if so, what lower complexity class is characterized if we remove them. We give a conditional answer, assuming a *semantic* conjecture.

Conjecture 2. *There exists a denotational model of* second-order *Intuitionistic Multiplicative*[1] *Affine Logic which interprets types as* finite sets*, and distinguishes between the booleans* `t` *and* `f` *in* Bool*.*

We shall not be precise as to the definition of a denotational model: we expect any concrete model to satisfy the required properties for the proof of our result. Our inspiration is the following theorem.

Theorem 3 (Hillebrand & Kanellakis [3])**.** *A language can be decided by a simply typed* λ*-term of type* $((A \to A) \to (A \to A) \to (A \to A)) \to (o \to o \to o)$ *(A can be chosen depending on the language, and o is a base type) if and only if it is* regular*.*

This can be proved rather quickly by interpreting the simply typed λ-calculus in the cartesian closed category of *finite* sets (o must have an interpretation of cardinality ≥ 2 to distinguish the booleans). This argument, which is an instance of *semantic evaluation* (cf. [4]), adapts easily to show:

Theorem 4. *Languages decided by proofs of* $!\mathsf{Str}[A] \multimap 1 \oplus 1$ *in propositional linear logic are regular.*

Indeed, propositional linear logic admits models with finite sets: for instance, finite coherence spaces, or the Scott model of prime algebraic lattices [4, §3]. (In fact[2], transition functions of finite automata recognizing those languages may be directly read from the interpretation of such a proof in the Scott model; this may be seen as a particularly simple version of Grellois & Melliès's work on higher-order model checking and semantics of linear logic, cf. [2, §9.1].) Analogously, our conditional result is:

Theorem 5. *If the above conjecture is true, then the languages decided by predicates in IEAL of type* $!\mathsf{Str} \multimap !!\mathsf{Bool}$ *are exactly the regular languages.*

[1] Actually, multiplicative connectives, weakening and impredicative quantifiation are enough to define the additives.

[2] Thanks to Pierre Pradic for communicating this remark to the first author.

Submitted to:
Linearity-TLLA 2018

© Le Thanh Dung Nguyen & Thomas Seiller
This work is licensed under the
Creative Commons Attribution License.

Proof sketch. We illustrate the idea on the easier case of the type $\mathsf{Str} \multimap !\mathsf{Bool}$. The only thing a proof of this formula can do, given a Str as input, is to instantiate it as $\mathsf{Str}[A]$ for some chosen type A, feed it some $f_0 : A \multimap A$ and $f_1 : A \multimap A$, retrieve some $g = f_{i_1} \circ \ldots \circ f_{i_n}$ depending on the input word, and apply a fixed program to this g to obtain a boolean. We may assume that A contains no exponential: it would be of no use because of the stratification at work in IEAL.

Suppose our semantics sends A to a set $\mathscr{A}$ and the f_i to endofunctions on $\mathscr{A}$, preserving composition. This defines a monoid morphism $\varphi : \{0,1\}^* \to \mathrm{End}(\mathscr{A})$, sending each input word to the denotation of the corresponding g. Let $P \subseteq \mathrm{End}(\mathscr{A})$ be the set of those g for which the boolean output is $\mathtt{t}$. Our language can thus be defined as $\varphi^{-1}(P)$; it is regular since φ has a finite codomain. □

If this turns out to be true, it would be to our knowledge the first implicit complexity characterization of regular languages by a type system with unrestricted quantification. We also expect our semantic conjecture to entail results on all the types $!\mathsf{Str} \multimap !^{k+2}\mathsf{Bool}$ in IEAL.

To conclude this note, we justify our belief in this conjecture – at least in the case of second-order Multiplicative Linear Logic (MLL^2), which would already be enough to establish that in Elementary Linear Logic without additives, proofs of $!\mathsf{Str} \multimap !!\mathsf{Bool}$ decide regular languages. Actually, we consider a more precise conjecture to be highly plausible.

Definition 6. Let A be a MLL^2 formula and π, π' be proofs of A. We write $\pi \sim_A \pi'$ if and only if, for any proof ρ of $A \vdash B$ where B is some *propositional* MLL formula, $\mathbf{cut}(\pi,\rho)$ and $\mathbf{cut}(\pi',\rho)$ are equivalent.

Conjecture 7. *For any* MLL^2 *formula A, there are finitely many equivalence classes for* $\sim_A$.

Thus, our denotational semantics in finite sets would simply be an observational quotient of the syntactic model. This has no chance to work in System F, for example, since monomorphic types can have infinitely many non-equivalent inhabitants. But thanks to linearity, in MLL^2, propositional formulae – and more generally, formulae with only universal quantifiers – have a finite number of proofs. The difficulty is to handle the existential quantifier, which may be used to hide witnesses of arbitrary size. But any witness introduced by a proof π must be treated *generically* by the test ρ. This should translate into a bound on the amount of information on π needed to compute $\mathbf{cut}(\pi,\rho)$, depending only on A. Typically, think of $\exists X.X$: all of its proofs are equivalent since there is no way to examine the inside of the X (indeed, there is no MLL^2 proof of $\exists X.X \vdash B$ for propositional B).

More precisely, the proof strategy we are currently investigating is to show that $\mathbf{cut}(\pi,\rho)$ can be normalized in such a way that if a $\otimes/⅋$ or $\forall/\exists$ cut-elimination step involves both a link from π and a link from ρ, then these links correspond to connectives appearing in the respective types $A, A^\perp$. Morally, ρ cannot know about the connectives in the existential witnesses of π, and vice versa. If this holds, then the normalization could be faithfully summarized by a "dialogue" between π and ρ whose messages are positions of axiom links. Hence the bounded information exchange.

References

[1] Patrick Baillot (2015): *On the expressivity of elementary linear logic: Characterizing Ptime and an exponential time hierarchy*. Information and Computation 241, pp. 3–31.

[2] Charles Grellois (2016): *Semantics of linear logic and higher-order model-checking*. PhD Thesis.

[3] Gerd G. Hillebrand & Paris C. Kanellakis (1996): *On the Expressive Power of Simply Typed and Let-Polymorphic Lambda Calculi*. In: *LICS'96*.

[4] Kazushige Terui (2012): *Semantic Evaluation, Intersection Types and Complexity of Simply Typed Lambda Calculus*. In: *RTA'12*.