LMSO: A Curry-Howard Approach to Church's Synthesis via Linear Logic*

Pierre Pradic[1][2] and Colin Riba[3]

[1] ENS de Lyon, Université de Lyon, LIP, Lyon, France
pierre.pradic@ens-lyon.fr
[2] University of Warsaw, Faculty of Mathematics, Informatics and Mechanics, Warsaw, Poland
[3] ENS de Lyon, Université de Lyon, LIP, Lyon, France
colin.riba@ens-lyon.fr

Abstract

We propose LMSO, a proof system inspired from Linear Logic, as a proof-theoretical framework to extract finite-state stream transducers from linear-constructive proofs of omega-regular specifications. We advocate LMSO as a stepping stone toward semi-automatic approaches to Church's synthesis combining computer assisted proofs with automatic decisions procedures. LMSO is correct in the sense that it comes with an automata-based realizability model in which proofs are interpreted as finite-state stream transducers. It is moreover complete, in the sense that every solvable instance of Church's synthesis problem leads to a linear-constructive proof of the formula specifying the synthesis problem.

1 Introduction

Church's synthesis [5] consists of the automatic extraction of stream transducers (or *Mealy machines*) from input-output specifications. Ideally, these specifications would be written in *Monadic Second-Order Logic* (MSO) on ω-words [18, 19]. MSO on ω-words is a decidable logic thanks to Büchi's Theorem [3], whose proof is originally based on an effective translation of MSO formulae to non-deterministic Büchi automata (NBAs). This logic subsumes non-trivial logics used in verification such as LTL (see e.g. [17, 1]). Church's synthesis for (subsystems of) LTL has also been substantially studied (see e.g. [8, 6, 2]).

Traditional theoretical solutions to Church's synthesis start from an ω-word automaton recognizing the specification (typically an NBA), and apply *McNaughton's Theorem* [9] to obtain an equivalent deterministic (say parity) automaton on ω-words. There are then essentially two methods (see e.g. [18, 19]). The first one turns the deterministic automaton into a game graph, in which the *Opponent* O ($\forall$bélard) plays input characters to which the *Proponent* P ($\exists$loïse) replies with output characters. Solutions to Church's synthesis are then given by the Büchi-Landweber Theorem [4], which says that in such games, either P or O has a finite-state winning strategy. The second one goes via infinite trees [13], noting that a causal function from say Σ to Γ can be represented by an infinite Γ-labeled Σ-ary tree.

In this work, extending [11], we advocate an approach to Church's synthesis in the framework of program extraction from proofs (in the sense of e.g. [16]). We propose a constructive deduction system for (an expressively equivalent variant of) MSO, based on a complete axiomatization of MSO on ω-words as a subsystem of second-order Peano arithmetic [15] (see also [14]). The formal proofs in this deduction system are interpreted in an automata-based

*This work was partially supported by the ANR-14-CE25-0007-RAPIDO and by the LABEX MILYON (ANR-10-LABX-0070) of Université de Lyon. An extended version of this abstract will appear in the proceedings of LICS'18 [12].

realizability semantics, along the lines of the Curry-Howard *proofs-as-programs* correspondence. Our system is correct, in the sense that from a proof of a ∀∃-specification one can extract a Mealy machine implementing the specification. It is moreover complete, in the sense that it proves all ∀∃-specifications which are realizable by Mealy machines.

The crux of our approach is that on the one hand the *correctness proof* of our realizability interpretation relies on McNaughton's Theorem, while on the other hand the *extraction* of realizers from formal proofs does not invoke it.

In the context of MSO, using a deduction system may avoid the systematic translation of formulae to automata, and may allow for human intervention and compositional reasoning. In a typical usage scenario, the user interactively performs some proofs steps and delegates the generated subgoals to automatized synthesis procedures. The partial proof tree built by the user is then translated to a combinator able to compose the transducers synthesized by the algorithms.

The deduction system SMSO proposed in [11] was based on intuitionistic logic. While SMSO is correct and complete for Church's synthesis, it suffers from a very limited set of primitive connectives ($\wedge$, $\neg$, $\exists$) so that formal proofs may be cumbersome without resorting to a negative translation from the complete axiomatization of MSO in classical logic. We propose a deduction system LMSO inspired from (intuitionistic) *Linear Logic* [7] (see also [10]). LMSO has a rich set of connectives (with primitive $\otimes, ⅋, \multimap, !, ?, \exists, \forall$), with a straightforward interpretation as usual automata constructions.[1] The system LMSO is moreover based on an extension MSO^+ of MSO with primitive function symbols for Mealy machines, allowing for a more efficient extraction of realizers from proofs.

The work discussed in this abstract is covered by [12]. The presentation at the workshop will emphazise proof-theoretical aspects relating deduction in the logical system LMSO with the realizability model, and in particular properties of LMSO provided by the model.

References

[1] C. Baier and J.-P. Katoen. *Principles of Model Checking.* MIT Press, 2008.

[2] R. Bloem, B. Jobstmann, N. Piterman, A. Pnueli, and Y. Sa'ar. Synthesis of reactive (1) designs. *Journal of Computer and System Sciences*, 78(3):911–938, 2012.

[3] J. R. Büchi. On a Decision Methond in Restricted Second-Order Arithmetic. In E. Nagel et al., editor, *Logic, Methodology and Philosophy of Science (Proc. 1960 Intern. Congr.)*, pages 1–11. Stanford Univ. Press, 1962.

[4] J. R. Büchi and L. H. Landweber. Solving Sequential Conditions by Finite-State Strategies. *Trans. Amer. Math. Soc.*, 138:367–378, 1969.

[5] A. Church. Applications of recursive arithmetic to the problem of circuit synthesis. In *Summaries of the SISL*, volume 1, pages 3–50. Cornell Univ., 1957.

[6] E. Filiot, N. Jin, and J.-F. Raskin. Antichains and compositional algorithms for LTL synthesis. *Form. Method. Syst. Des.*, 39(3):261–296, Dec 2011.

[7] J.-Y. Girard. Linear Logic. *Theor. Comput. Sci.*, 50:1–102, 1987.

[8] O. Kupferman, N. Piterman, and Y. Vardi, M. Safraless Compositional Synthesis. In T. Ball and R. B. Jones, editors, *Proceedings of CAV'06*, pages 31–44. Springer, 2006.

[9] R. McNaughton. Testing and generating infinite sequences by a finite automaton. *Inf. Control*, 9(5):521 – 530, 1966.

[10] P.-A. Melliès. Categorical semantics of linear logic. In *Interactive models of computation and program behaviour*, volume 27 of *Panoramas et Synthèses*. SMF, 2009.

[1] The usual additive connectives $\oplus, \&$ of Linear Logic have also natural interpretations in automata.

[11] P. Pradic and C. Riba. A Curry-Howard Approach to Church's Synthesis. In *Proceedings ot FSCD'17*, volume 84 of *LIPIcs*, pages 30:1–30:16. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

[12] P. Pradic and C. Riba. LMSO: A Curry-Howard Approach to Church's Synthesis via Linear Logic. In *LICS*. ACM, 2018.

[13] M. O. Rabin. *Automata on Infinite Objects and Church's Problem.* Amer. Math. Soc., 1972.

[14] C. Riba. A model theoretic proof of completeness of an axiomatization of monadic second-order logic on infinite words. In *Proceedings of IFIP-TCS'12*, 2012.

[15] D. Siefkes. *Decidable Theories I : Büchi's Monadic Second Order Successor Arithmetic*, volume 120 of *LNM*. Springer, 1970.

[16] M. H. Sørensen and P. Urzyczyn. *Lectures on the Curry-Howard Isomorphism*, volume 149 of *Studies in Logic and the Foundations of Mathematics*. Elsevier Science Inc., 2006.

[17] W. Thomas. Languages, Automata, and Logic. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume III, pages 389–455. Springer, 1997.

[18] W. Thomas. Solution of Church's Problem: A tutorial. *New Perspectives on Games and Interaction*, 5:23, 2008.

[19] W. Thomas. Facets of Synthesis: Revisiting Church's Problem. In L. de Alfaro, editor, *Proceedings of FOSSACS'09*, pages 1–14. Springer, 2009.