Recursion Schemes and the WMSO+U Logic*

Paweł Parys

In the proposed talk I plan to present the following result:

Theorem 1 *It is decidable whether the tree generated by a given higher-order recursion scheme satisfies a given* WMSO+U *sentence.*

This contribution is not yet published, the paper is in preparation.

Background

Higher-order recursion schemes (*schemes* in short) are used to faithfully represent the control flow of programs in languages with higher-order functions. This formalism is equivalent via direct translations to simply-typed λY-calculus. Collapsible pushdown systems and ordered tree-pushdown systems are other equivalent formalisms. Schemes cover some other models such as indexed grammars and ordered multi-pushdown automata.

In our setting, a scheme is a finite description of an infinite tree. A useful property of schemes is that the *MSO-model-checking problem* for schemes is decidable. This means that given a scheme $\mathcal{G}$ and an MSO sentence φ, it can be algorithmically decided whether the tree generated by $\mathcal{G}$ satisfies φ. This result has several different proofs, and also some extensions like global model checking, logical reflection, effective selection, existence of λ-calculus model. When the property of trees is given as an automaton, not as a formula, the model-checking problem can be solved efficiently, in the sense that there exist implementations working in a reasonable running time (most tools cover only a fragment of MSO, though).

Recently, an interest arisen in model-checking trees generated by schemes against properties not expressible in the MSO logic. These are properties expressing boundedness and unboundedness of some quantities. More precisely, it was shown that the *diagonal problem* for schemes is decidable. This problem asks whether for every $n \in \mathbb{N}$ there exists a path in the tree generated by the given scheme such that every letter from a given set A appears on this path at least n times. This result turns out to be interesting, because it entails other decidability results for recursion schemes, concerning in particular computability of the downward closure of recognized languages, and the problem of separability by piecewise testable languages.

In our work we show a result of a more general style. Instead of considering a particular property, like the diagonal problem, we consider a logic capable to express properties talking about boundedness. More precisely, we choose the WMSO+U logic. This logic extends WMSO (a fragment of MSO in which one can quantify only over finite sets) by the unbounding quantifier, U. A formula using this quantifier, $\mathsf{U}X.\,\varphi$, says that φ holds for arbitrarily large finite sets X. The WMSO+U logic was widely considered in the context of infinite words and infinite trees.

We remark that the model-checking problem for the full MSO logic (equipped with quantification over infinite sets) combined with the U quantifier is undecidable already over the the infinite word without labels, so even more over all fancy trees that can be generated by higher-order recursion schemes. For this reason it is necessary to restrict the quantification to finite sets.

Idea of the Proof

In our solution, we depend on several earlier results. First, we translate WMSO+U formulae to an equivalent model of nested automata, where every level of the automaton corresponds to a

*Work supported by the National Science Centre, Poland (grant no. 2016/22/E/ST6/00041).

single quantifier of the logic. The translation is rather straightforward, and uses the notion of logical types, aka. composition method. Second, we use the logical-reflection property of schemes. It says that given a scheme $\mathcal{G}$ and an MSO sentence φ one can construct a scheme $\mathcal{G}_\varphi$ generating the same tree as $\mathcal{G}$, where in every node it is additionally written whether φ is satisfied in the subtree starting in this node. Third, from our previous work on the diagonal problem, we deduce an analogous property for the diagonal problem, which we call *diagonal reflection*: given a scheme $\mathcal{G}$ we can construct a scheme $\mathcal{G}_{diag}$ generating the same tree as $\mathcal{G}$, where every node is additionally annotated by the solution of the diagonal problem in the subtree starting in this node. We believe that the diagonal-reflection property is a contribution of independent interest. Finally, we use the fact that schemes can be composed with finite tree transducers transforming the generated trees; this follows directly from the equivalence between schemes and collapsible pushdown systems.