Developing Sorting Algorithms using Proof–Based Synthesis

Tudor Jebelean

RISC, JKU Linz, Austria

Tudor.Jebelean@JKU.AT

An alternative to the classical approach to certified programming (algorithm design followed by formal verification) is the development in parallel of the following formal items:

- the *object theory* relevant for the objects manipulated by the algorithm,
- the *specification* of the problem to be solved, and
- the *proof* that a solution to the problem exist, from which the *algorithm* can be extracted.

We describe a case study of automatic assistance to this process in the case of sorting by the automated reasoning environment Theorema (`www.theorema.org`), which allows to define and organize the logical formulae expressing the mathematical theory, the specification, and the algorithm, as well as automatic proofs of necessary properties. In particular, the system allows to prove the formalization of the *synthesis statement* "for any list, there exists a sorted version of it" and to extract automatically the algorithm from it. The algorithm is expressed as set of conditional equalities and it is executable by the system.

We construct an object theory of lists, consisting in basic axioms and proven properties, in a process which may be seen as *theory exploration*. In parallel, we formalize the specification of the sorting problem and we develop the proof of the synthesis statement. By user choice of the appropriate knowledge given to the prover, as well as of various proof strategies and induction principlee, this results in four different known sorting algorithms: selection-sort, insertion-sort, merge-sort, and quick-sort, plus one which is a new variation of merge-sort.

The theory is constructed in first order logic, and most of the properties are equivalent to Horn clauses, thus in principle most parts of the proofs could be carried out by SLD resolution, however this leads to very large proofs. Therefore we followed the Theorema tradition of generating proofs in natural style, by using novel proof techniques for lists. This leads to much shorter and human readable proofs. Thus several interesting proof techniques for lists have been revealed during the construction of an appropriate prover. For instance we discovered specific inference rules and strategies for reasoning with the equivalence relation over lists (induced by the predicate "have the same elements") and with various ordering relations on lists (induced by the ordering among elements). We also use a novel treatment of the failed proof branches on goals containing no lists, in order to improve the proof and find case distinctions in the algorithms. Also, we experimented with the use of various induction principles as expressions of various algorithms structures.

Acknowledgements. This is joint work with Isabela Dramnesc, and extends previous work done by Bruno Buchberger.

References

[1] I. Dramnesc, T. Jebelean. *Theory Exploration in Theorema: Case Study on Lists.* In: Proceedings of the 7th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI 2012), pp. 421-426.

[2] B. Buchberger, A. Craciun, T. Jebelean, L. Kovacs, T. Kutsia, K. Nakagawa, F. Piroi, N. Popov, J. Robu, M. Rosenkranz, and W. Windsteiger. Theorema: Towards Computer-Aided Mathematical Theory Exploration. *Journal of Applied Logic*, 4(4):470–504, 2006.