Inductive Termination Proofs by Transition Predicate Abstraction and their relationship to the Size-Change Abstraction

Florian Zuleger

Technische Universität Wien

Abstract

The last decade has seen renewed interest in automated techniques for proving the termination of programs. A popular termination criterion is based on the covering of the transitive hull of the transition relation of a program by a finite number of well-founded relations. In an automated analysis, this termination criterion is usually established by an inductive proof using transition predicate abstraction. Such termination proofs have the structure of a finite automaton. These automata, which we call transition automata, are the central object of study in this paper. Our main results are as follows: (1) A previous criterion for termination analysis with transition automata is not complete; we provide a complete criterion. (2) We show how to bound the height of the transition relation of the program using the termination proof by transition predicate abstraction. This result has applications in the automated complexity analysis of programs. (3) We show that every termination proof by transition predicate abstraction gives rise to a termination proof by the size-change abstraction; this connection is crucial to obtain results (1) and (2) from previous results on the size-change abstraction. Further, our result establishes that transition predicate abstraction and size-change abstraction have the same expressivity for automated termination proofs.

1 Introduction

The last decade has seen a renewed interest in automated techniques for proving the termination of programs. In particular the Terminator termination analyzer [2] has received widespread attention, being able to analyse device drivers with several thousand lines of code, hinting at potential termination bugs. The soundness proof of the analysis in [2] made use of a termination criterion suggested by Rybalchenko and Podelski [5] (for a discussion of earlier work that implicitly used the same principle we refer the reader to [1]): In order to show the well-foundedness of a relation R, it is sufficient to find a finite number of well-founded relations $R_1, \ldots, R_k$ with

$$R^+ \subseteq R_1 \cup \cdots \cup R_k \ (*) \ ,$$

where R^+ denotes the transitive hull of R.

The essential question in using this termination criterion is how to establish the condition (*). The difficulty lies in reasoning about the transitive hull R^+ which usually requires induction. Indeed, it is already suggested in [5] to establish (*) by an inductive argument. The idea of an inductive argument was developed further in [6], where the use of *transition predicate abstraction* (TPA) is suggested. Transition predicates are predicates over primed and unprimed variables, where unprimed variables refer to the current state and primed variables to the next state; in this way transition predicates allow to describe relations. The idea of inductive termination proofs by TPA has been successfully implemented in the cited termination analyzer Terminator. The starting point of our research is the structure of the inductive termination proofs by transition predicate abstraction. These proofs have the

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

structure of a finite automaton, which we call *transition automata* for future reference. We show the structure of the transition automata carries substantially more information than has been exploited in previous work. In particular we show that the soundness of inductive termination proofs by TPA can be shown solely by *automata-theoretic techniques* without referring to the condition (*). This is interesting because the proof of (*) given in [6] makes of Ramsey's theorem. Ramsey's theorem does not use the inductive structure of the termination proof — captured by the transition automaton — , but relies on general combinatoric principles. It is precisely this structure which allows us to obtain stronger results, which we discuss in the following:

Result 1: Following [6] we examine a first criterion for termination proofs based on transition automata. We show that the universality of the transition automaton implies the termination of the program under analysis. Surprisingly, this termination argument turns out to be incomplete. We define a weaker criterion and show its completeness.

Result 2: An interesting line of work [1, 7] has studied how termination proofs based on the termination criterion (*) can be used to infer bounds on height of the relation R in terms of the height of the relations $R_1, \ldots, R_k$. These works are purely based on the criterion (*) and do not use the structure offered by the transition automata. Analysing transition automata allows us to obtain precise estimates for transition relations whose height is bounded by a natural number.

Result 3: Termination proofs by transition predicate abstraction bear a remarkable resemblance to termination proofs by the *size-change abstraction* (SCA), which has been introduced by Ben-Amram, Lee and Jones in [4]. This similarity has been the subject of previous research [3], which contains first results. In this paper we establish the fundamental result the every termination proof by TPA gives rise to a termination proof by SCA. This is particularly striking because the termination criterion (*) has been suggested in [5] as a generalization of termination proofs by the size-change termination. However, our results establish that TPA and SCA have the same expressivity for automated termination proofs. It is precisely this relationship between SCA and TPA which allows us to establish Results 1 and 2.

References

1 Andreas Blass and Yuri Gurevich. Program termination and well partial orderings. *ACM Trans. Comput. Log.*, 9(3):18:1–18:26, 2008.

2 Byron Cook, Andreas Podelski, and Andrey Rybalchenko. Termination proofs for systems code. In *PLDI*, pages 415–426, 2006.

3 Matthias Heizmann, Neil D. Jones, and Andreas Podelski. Size-change termination and transition invariants. In *Static Analysis - 17th International Symposium, SAS 2010, Perpignan, France, September 14-16, 2010. Proceedings*, pages 22–50, 2010.

4 Chin Soon Lee, Neil D. Jones, and Amir M. Ben-Amram. The size-change principle for program termination. In *POPL*, pages 81–92, 2001.

5 Andreas Podelski and Andrey Rybalchenko. Transition invariants. In *LICS*, pages 32–41, 2004.

6 Andreas Podelski and Andrey Rybalchenko. Transition predicate abstraction and fair termination. *ACM Trans. Program. Lang. Syst.*, 29(3):15, 2007.

7 Silvia Steila and Keita Yokoyama. Reverse mathematical bounds for the termination theorem. *Ann. Pure Appl. Logic*, 167(12):1213–1241, 2016.