Available online at www.sciencedirect.com
SCIENCE DIRECT®
Theoretical Computer Science 343 (2005) 332–369
Theoretical Computer Science
www.elsevier.com/locate/tcs

Parameterised boolean equation systems

Jan Friso Groote[a,*], Tim A.C. Willemse[a,b]

[a] *Department of Mathematics and Computer Science, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands*

[b] *Faculty of Science, Mathematics and Computing Science, University of Nijmegen, P.O. Box 9010, 6500 GL Nijmegen, The Netherlands*

Abstract

Boolean equation system are a useful tool for verifying formulas from modal μ-calculus on transition systems (see [Mader, Lecture Notes in Computer Science, Vol. 1019, 1995, pp. 72–88] for an excellent treatment). We are interested in an extension of boolean equation systems with data. This allows to formulate and prove a substantially wider range of properties on much larger and even infinite state systems. In previous works [Groote and Mateescu, Lecture Notes in Computer Science, Vol. 1548, 1999, pp. 74–90; Groote and Willemse, Sci. Comput. Program., 2005] it has been outlined how to transform a modal formula and a process, both containing data, to a so-called parameterised boolean equation system, or equation system for short. In this article we focus on techniques to solve such equation systems.

We introduce a new equivalence between equation systems, because existing equivalences are not compositional. We present techniques similar to Gauß elimination as outlined in [Mader, Lecture Notes in Computer Science, Vol. 1019, 1995, pp. 72–88] that allow to solve each equation system provided a single equation can be solved. We give several techniques for solving single equations, such as approximation (known), patterns (new) and invariants (new). Finally, we provide several small but illustrative examples of verifications of modal μ-calculus formulas on concrete processes to show the use of the techniques.

Keywords: First order modal μ-calculus; Parameterised boolean equation systems; Model checking; Infinite state systems

* Corresponding author.
E-mail address: J.F.Groote@tue.nl (J.F. Groote).

0304-3975/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2005.06.016

1. Introduction

Boolean Equation Systems (BESs) [20,21,25] are systems of the form $(\sigma_1 X_1 = f_1) \ldots (\sigma_N X_N = f_N)$, where σ_i is either a least fixpoint symbol μ or a greatest fixpoint symbol ν and f_i is a propositional formula. These systems can be seen as generalisations of nested and alternating fixpoint expressions, interpreted over a Boolean lattice.

BESs have been studied in detail by Vergauwen and Lewi [25], and Mader [20,21] in the context of model checking modal μ-calculus formulae. In [21], Mader shows that the model checking problem can be solved by solving BESs. Furthermore, she provides a complete proof system for solving BESs by means of algebraic manipulations.

Parameterised boolean equation systems (PBESs) (also known as *First-Order* Boolean Equation Systems) [11,15,26] are sequences of equations of the form $\sigma X(d_1{:}D_1, \ldots, d_n{:}D_n) = \varphi$, where σ is either a least or a greatest fixpoint symbol, d_i is a data variable of sort D_i and φ is a predicate formula. The sort $D_1 \times \cdots \times D_n$ is referred to as the *parameter-space* of a parameterised boolean equation.

PBESs form an extension of plain BESs. Groote and Mateescu [11] introduced these PBESs as an intermediate formalism for model checking processes with (arbitrary) data. Extending on the results of Mader [20,21], they showed that their model checking problem could be translated to the problem of solving PBESs. In [11], they provided four proof rules for approximating the solution of single parameterised equations: two for the least fixpoint and two for the greatest fixpoint. Furthermore, as a proof of concept, we showed in [15,26] that PBESs can be solved automatically by means of a technique that combines the essentials of Gauß-elimination [20,21], and approximation (see e.g. [10]).

While the automated approach has proved successful for several practical applications, it also illustrates the undecidability of model checking when no restrictions on the involved data-types are made, by occasionally requiring transfinite approximations of fixpoint expressions (i.e., in such cases, approximation procedures do not terminate). The emphasis on automation set a scene where possible remedies for such situations were hard to find.

Inspired by this latter observation, we take a different approach altogether in this paper, and focus on algebraic techniques that help in solving PBESs by hand. While this may seem a step back to some, being able to solve PBESs by hand provides a better understanding of the techniques that are involved. We intentionally proved many properties about systems by hand, some of which can be found in the second part of this paper, with as primary goal to build up experience and skill. As expected this led to effective techniques to manually solve parameterised boolean equation systems which are reported in the first part of this paper. Although it is not the focus of this paper, we expect that these techniques will also have a positive impact on the mechanised and automatic verification of modal formulas on processes in a setting with data.

The approach we describe in this paper is similar in spirit to the algebraic approach for solving BESs, taken by Mader [21]. We separate the problems of solving PBESs as a whole, and parameterised boolean equations in isolation. Central to our approach is the notion of a *system equivalence* that allows us to reason compositionally about PBESs. While in [21], also a system equivalence is introduced for BESs, it turns out that this equivalence is not compositional. We illustrate this fact by a simple example in Section 3. Together with system