Theoretical Computer Science ••• (••••) •••–•••

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

From hidden to visible: A unified framework for transforming behavioral theories into rewrite theories

Min Zhang[a,b,*], Kazuhiro Ogata[c]

[a] *Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai, China*
[b] *MoE International Joint Lab of Trustworthy Software, East China Normal University, Shanghai, China*
[c] *School of Information Science, Japan Advanced Institute of Science and Technology, Nomi, Japan*

ARTICLE INFO

Article history:
Received 22 January 2016
Received in revised form 8 January 2018
Accepted 8 January 2018
Available online xxxx
Communicated by M. Wirsing

Keywords:
Behavioral theory
Rewrite theory
Transformation
Linear-time property

ABSTRACT

Algebraic formalization and verification are effective and practical ways of modeling and verifying software systems by both model checking and theorem proving techniques. In algebraic approaches, a system can be modeled either in a *hidden* way as a behavioral theory or in a *visible* way as a rewrite theory. Several approaches have been proposed to transform behavioral theories into rewrite theories for integrating model checking and theorem proving in verification. In this paper, we propose a framework for transforming behavioral theories into rewrite theories, which unifies four existing related transformation approaches. In this framework, each existing transformation approach can be viewed as a process of transforming behavioral theories first into a special class of behavioral theories and finally into rewrite theories. From this perspective, these transformation approaches differ from each other only in the transformation from ordinary behavioral theories into the classified ones, and their transformations from the classified ones into rewrite theories are essentially the same. We prove that the transformation framework preserves linear-time properties. The preservation of linear-time properties guarantees that a counterexample found by model checking a linear-time property with a generated rewrite theory is also a counterexample in the original behavioral theory, as required by integrated verification.

1. Introduction

In formal methods, algebraic specification and verification become effective and practical ways of specifying and verifying software systems with the birth of the famous algebraic specification language OBJ originated by ADJ group [22]. A number of algebraic specification languages have been designed and developed. Most of them provide tool support for various formal analysis approaches, such as the OBJ language family [25] including two representatives, i.e., Maude [9] and CafeOBJ [12], ELAN [6], CASL (Common Algebraic Specification Language) [2], ASL+DSL [46] and even some domain-specific modeling languages based on algebraic formalization [43]. In algebraic formalisms, the static feature such as data and states of systems is formalized as abstract data types, and the dynamic feature such as behaviors of systems is formalized in two different ways by equations and rewrite rules, respectively. Their corresponding theories are called behavioral theories and rewrite theories, which are two main variants of specifying computer systems in the algebraic way.

[*] Corresponding author at: Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai, China.
E-mail addresses: zhangmin@sei.ecnu.edu.cn (M. Zhang), ogata@jaist.ac.jp (K. Ogata).

https://doi.org/10.1016/j.tcs.2018.01.006

A behavioral theory characterizes the way in which systems behave by distinguishing the sorts used for data values from those that are used for states, and defining the behavioral notion of satisfaction based on the idea of indistinguishability of states that are observationally the same [24,12]. In a behavioral specification, states are represented in a *hidden* way in that values in states are hidden and only observed indirectly by performing "experiments", i.e., using a special set of operators to return visible data values from the states [24]. Behavioral specifications are mainly used for verifying invariant properties by interactive theorem proving and suited to infinite-state systems. Invariant properties are the most common and useful safety properties which state that something bad should never happen. However, verification by interactive theorem proving usually needs much intellectual effort to conjuncture lemmas during theorem proving or find counterexamples manually based on the verification result [17,10].

In rewrite theories, objects in a computer system are usually formalized in a *visible* way as a *soup* of components based on an associative-commutative (AC) *soupification* operation [31,9]. Such components constitute a *configuration*, representing a state of the system. Rewrite theories can be used for model checking of not only invariant properties but also other temporal properties such as liveness properties. By model checking, counterexamples can be automatically found if a property fails to be satisfied by a system [13]. Unlike those well-known model checkers, such as SPIN [26] and NuSMV [7], which require the systems being model checked to have finite state space, model checking with rewrite theories can be also applied to those systems that have a finite space of reachable states [13]. Even for the systems that have infinite space of reachable states, they can be partially model checked by simply setting up a bound. However, model-checking technique also has its weakness, i.e., the state explosion problem. Counterexamples may not be found when the state space is too large to search exhaustively in a reasonable time.

Integration of theorem proving and model checking is a prevalent methodology of making the best use of the two techniques and meanwhile avoiding their weaknesses. For instance, the communication protocol of Mondex electronic purses is specified by stepwise refinement using RAISE formal specification language, and formal verification is made by translating to PVS [44] for theorem proving and to SAL [11] for model checking, respectively [20]. The PVS proof checker integrates the mu-calculus model checker by transforming fragments of PVS in high-order logic into propositional mu-calculus [40]. As for algebraic formal methods, some approaches have been proposed for the collaborative verification using theorem proving and model checking, such as an *induction-guided falsification* approach [38], a light-weight integration approach [28] and a *generate&check* method [16]. Maude also provides both ITP theorem prover [10] and LTL model checker [13]. The integrated verifications require a system to be specified both in behavioral theory and rewrite theory. Consequently, the transformation between the two theories becomes necessary and useful. Several approaches have been proposed to transform from the behavioral theories specified in CafeOBJ into the rewrite theories specified in Maude, such as those in the work [28,15,32] and the one in our earlier work [48,49]. These approaches differ from each other either in the way how the hidden states are transformed into configurations or in the scope of the behavioral theories that can be translated. Moreover, some approaches are proposed from the pragmatic point of view and hence are *ad hoc* to some specific class of behavioral theories. There is little attention paid to the problems that to which extent these existing approaches can be applied to the transformation of an arbitrarily given behavioral theory, and whether or not there exists a unified transformation approach applicable to all the behavioral theories.

In this work, we propose a unified framework for transforming behavioral theories into rewrite theories. The framework is called *unified* in that four other existing related transformation approaches can be viewed as special cases under it. In the framework, we firstly identify a class of behavioral theories which we call *0-1 bounded behavioral theories*, and then propose a transformation approach from the 0-1 bounded behavioral theories into rewrite theories. Intuitively, a behavioral theory is 0-1 bounded if by each transition there is at most one value changed among those that are observed by the same observer. Each existing transformation approach can be viewed as a process of transforming firstly from a behavioral theory to a 0-1 bounded one and finally from the intermediate 0-1 bounded behavioral theory to a rewrite theory. From this perspective, we find that these approaches differ from each other only in the transformation from ordinary behavioral theories to 0-1 bounded ones, and the transformations from the 0-1 bounded ones to rewrite theories are essentially the same. We also prove that our transformation framework preserves linear-time properties by showing the trace equivalence of the original 0-1 bounded behavioral theories and the corresponding generated rewrite theories. The preservation of linear-time properties guarantees that a counterexample found by model checking a linear-time property with a generated rewrite theory is also a counterexample in its original behavioral theory. Based on this property, we can perform light-weight integrated verification by doing theorem proving with original behavioral theory and doing model checking with the rewrite theory generated from it.

Organization of the paper: Section 2 introduces some preliminaries of behavioral theories and rewrite theories. We present our transformation framework in Section 3, and show how the existing transformation approaches are unified under the framework in Section 4. Section 5 presents the proof of the correctness of the transformation framework by showing its preservation of linear-time properties. Section 6 discusses some related work, and Section 7 concludes the paper.

2. Preliminaries

In this section, we present some background knowledge that are necessary to understand the proposed transformation framework.