Theoretical Computer Science 612 (2016) 45–62

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Model checking computation tree logic over finite lattices

Haiyu Pan [a,b], Yongming Li [b], Yongzhi Cao [c,d,*], Zhanyou Ma [b]

[a] *College of Computer Science and Technology, Taizhou University, Taizhou 225300, China*
[b] *College of Computer Science, Shaanxi Normal University, Xi'an 710062, China*
[c] *Institute of Software, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China*
[d] *Key Laboratory of High Confidence Software Technologies (Peking University), Ministry of Education, China*

ARTICLE INFO

Article history:
Received 23 January 2014
Received in revised form 27 August 2015
Accepted 9 October 2015
Available online 20 October 2015
Communicated by P. Aziz Abdulla

Keywords:
Model checking
Computation tree logic
Lattice
Weighted automata

ABSTRACT

Multi-valued model-checking is an extension of classical model-checking used to verify the properties of systems with uncertain information content. It is used in systems where the semantic domain of both the models of the systems and the specification is a De Morgan algebra or more abstract structure such as semirings. A common feature of De Morgan algebras and semirings is that they satisfy the distributive law, which is crucial for all of the multi-valued model-checking algorithms developed so far. In this paper, we study computation tree logic with membership values in a finite lattice, which are called multi-valued computation tree logic (MCTL). We introduce three semantics for MCTL over the multi-valued Kripke structure: path semantics, fixpoint semantics, and algebraic semantics, and prove that they coincide if and only if the underlying lattice is distributive. Furthermore, we provide model-checking algorithms with respect to the three semantics. Our algorithms show that MCTL model-checking problems with respect to the fixpoint and algebraic semantics can be solved in polynomial time in the size of the state space of the multi-valued Kripke structure, the size of the lattice, and the length of the formula, while MCTL model-checking problem with respect to the path semantics is in PSPACE. We also provide a lower bound for the MCTL model-checking problem with respect to the path semantics.

1. Introduction

Model-checking [3,11] is a technique to automatically determine whether a system model satisfies a specification. It has been established as one of the most effective formal verification techniques for analyzing the correctness of software and hardware designs. The model-checking approach has been extended to handle real-time [2], probabilistic [3], quantum [15,35,36], fuzzy [24–27], and multi-valued systems during the last several decades. Multi-valued model-checking is a generalization of classical model-checking methods from a two-valued setting to an arbitrary distributive De Morgan algebra or a semiring. It is proving valuable as the basis for a variety of new verification methods, including abstract techniques [16], temporal logic query checking [17], and reasoning about conflicting or inconsistent viewpoints [19].

It is worth noting that De Morgan algebras and semirings in the multi-valued model-checking setting have a common feature: they all satisfy the distributive law. More precisely, the multiplication of a semiring distributes over addition, and the meet and join operators of the lattice distribute over each other. The distributive law is crucial in the theory

[*] Corresponding author at: Institute of Software, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China.
E-mail addresses: phyu76@126.com (H. Pan), liyongm@snnu.edu.cn (Y. Li), caoyz@pku.edu.cn (Y. Cao), mazhany@126.com (Z. Ma).

http://dx.doi.org/10.1016/j.tcs.2015.10.014

of multi-valued model-checking developed so far. Nevertheless, many lattices are not distributive [12]. The importance of non-distributive finite De Morgan algebras for multi-valued model-checking was stressed in [29]. General lattices have applications in many fields of computer science including automata with finite words and infinite words [13,14,23], lattice tree automata [22], and description logics [31]. The notion of orthomodular lattice-valued (quantum) automata was introduced as a mathematical model of a quantum computer [33,34], where orthomodular lattices do not necessarily satisfy the distributive law. Recently, Li [23] investigated a lattice-valued finite automaton with membership values in a general (not necessarily distributive) lattice. Droste and Vogler [13] defined weighted monadic second order logic over strong bimonoids [14] where the class of strong bimonoids contains all bounded lattices. It should be pointed out that temporal logics [3,11] such as linear temporal logic (LTL) and computation tree logic (CTL) can be regarded as fragments of monadic second order logic [4]. Lehmann and Peñaloza [22] analyzed the complexity of computing the behavior of lattice-valued automata on infinite trees, where the underlying lattice is not distributive as well.

In view of the attractiveness of multi-value model-checking approaches and the usefulness of general lattices, we would like to consider multi-valued model-checking problems that take values from an arbitrary finite lattice in this paper. To this end, we introduce a multi-valued temporal logic that takes truth values from finite lattices $\mathcal{L}$ as a specification language for multi-valued Kripke structures (MKSs). The syntax of such a temporal logic, which will be referred to as multi-valued computation tree logic (MCTL), is the same as an existential normal form of classical CTL [3,11]. It has three kinds of operators: propositional logical operators such as conjunction, disjunction, and negation, path operators such as “next” (X), “until” (U), and “always” (G), and existential path quantifier $\exists$.

We discuss three semantics based on MKSs: path semantics, fixpoint semantics, and algebraic semantics. The three semantics interpret the MCTL formulas as mappings from the set of states of MKSs to the domain of $\mathcal{L}$. For the three semantics, the conjunction, disjunction, and negation logical operators are interpreted as the meet, join, and complementation operators in $\mathcal{L}$, respectively. For the path semantics, the truth values of the path operators U and G are the join and meet of truth values over a given path, respectively; the existential path quantifier $\exists$ determines the least upper bound for values of all paths from a given state. It turns out that the path semantics of MCTL is a lattice-valued generalization of the classical interpretations of formulas in CTL. There are many quantitative extensions of CTL and their corresponding path semantics are proposed in the literature (see, for example, [1–3,28,25–27]).

The fixpoint semantics is given by extending the classical fixpoint semantics of CTL [3,11], which is very useful for symbolic model checking, to the lattice-valued setting. Both the syntax and the fixpoint semantics of our logic are similar to $\mathcal{X}$CTL proposed by Chechik et al. in [8,9] except that our structure of truth values is more general. Based on $\mathcal{X}$CTL, they implemented the first multi-valued symbolic model-checker $\mathcal{X}$Chek [9]. In addition, some researchers often implicitly or explicitly considered the fixpoint semantics of their logics [1,3,11,28,24–27], when studying the quantitative versions of CTL model-checking problems with respect to the path semantics. The advantage is that one can use fixpoint iteration algorithms to solve model-checking problems, when the two semantics coincide.

The algebraic semantics of MCTL represents the meaning of MCTL formulas by using matrices instead of specific paths. It is motivated by the initial algebraic semantics of strong bimonoids-weighted automata [10,14] and the semantics of valued CTL based on semirings-weighted automata [7]. The algebraic semantics advantages over the path semantics and fixpointed semantics is that its model-checking problem can be solved by using the well-known algorithms for matrices (see [3,24,25,27], for example).

A main result of the paper is that the three semantics coincide if and only if the lattice is distributive. Hence, in terms of non-distributive lattices, the three semantic interpretations yield very different model-checking results and require different algorithmic techniques. We present model-checking algorithms with respect to the three semantics of MCTL over MKSs. Like CTL, the algorithms work recursively on the sub-formulas of a given formula, and thus we need only to consider the cases for $\exists U$ and $\exists G$. For the fixpoint and algebraic semantics, we present the model-checking algorithms based on iterations for $\exists U$ and $\exists G$. The resulting algorithms are polynomial in the size of the state space of the multi-valued Kripke structure, and linear in both the size of the lattice and the length of the formula. For the path semantics, we first show that its model-checking problem is decidable. One cannot use a brute-force algorithm to solve the MCTL model-checking problem with respect to the path semantics, since such an algorithm requires exponential time. Motivated by the reduction method of multi-valued model-checking problems and the tableaux algorithm of description logic over lattices [31], we provide an algorithm via reduction to classical model-checking algorithms. The resulting algorithm gives an upper bound for MCTL model-checking problem with respect to the path semantics: The MCTL model-checking problem with respect to the path semantics is in PSPACE. We also provide a lower bound for the problem by exploiting reduction from the behavior verification problem of lattice-valued tree automata in [22]: The MCTL model-checking problem with respect to the path semantics is in both NP-hard and co-NP-hard.

Related work: In the literature, there are many results on multi-valued model-checking techniques. To the best of our knowledge, our paper is the first to study the multi-valued model-checking problems in an arbitrary finite lattice-valued setting.

Model checking multi-valued versions of the classical logics LTL, CTL, CTL*, and μ-calculus have already been extensively studied. The multi-valued model-checking algorithms can be grouped into two classes. The first class includes all the algorithms that reduce the multi-valued problem to a set of two-valued or three-valued model-checking problems [5,18,21]. For example, Bruns and Godefroid [5] introduced an approach to derive a classical Kripke structure for each join-irreducible