Knowledge-Based Systems 50 (2013) 279–295

Contents lists available at SciVerse ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Model checking epistemic–probabilistic logic using probabilistic interpreted systems ☆

Wei Wan, Jamal Bentahar *,[1], Abdessamad Ben Hamza

Concordia Institute for Information Systems Engineering, Concordia University, Montreal, Canada

ARTICLE INFO

Article history:
Received 24 June 2012
Received in revised form 22 June 2013
Accepted 24 June 2013
Available online 5 July 2013

Keywords:
Model Checking
Verification
Interpreted systems
Multi-agent systems
Markov chains
Markov decision processes
Probabilistic and epistemic logic

ABSTRACT

Model checking is a formal technique widely used to verify security and communication protocols in epistemic multi-agent systems against given properties. Qualitative properties such as safety and liveliness have been widely analyzed in the literature. However, systems have also quantitative and uncertain (i.e., probabilistic) properties such as degree of reliability and reachability, which still need further attention from the model checking perspective. In this paper, we analyze such properties and present a new method for probabilistic model checking of epistemic multi-agent systems specified by a new probabilistic–epistemic logic PCTLK. We model multi-agent systems as distributed knowledge bases using probabilistic interpreted systems and define transformations from those interpreted systems into discrete-time Markov chains and from PCTLK formulae to PCTL formulae, an existing extension of CTL with probabilities. By so doing, we are able to convert the PCTLK model checking problem into the PCTL one. Thus, we make use of PRISM, the model checker of PCTL without adding new computation cost. A concrete case study has been implemented to show the applicability of the proposed technique along with performance analysis and comparison with MCK, an epistemic–probabilistic model checker, and MCMAS, a model checker for multi-agent systems, in terms of execution time and state space scalability.

1. Introduction

Model checking is a formal, fully automatic, well-designed technique to verify whether or not system design models satisfy given requirements [30]. In recent years, this technique has been applied to a wide range of systems and applications including process-based systems [40], multi-agent applications [35,39], agent communication [5], and service composition [6,36]. In conventional model checking, such as the technique used in [6], verification only focuses on the absolute accuracy of properties in the model being constructed, which means whether the checked properties are true or false. However, actual scenarios are rarely absolutely reliable but most often probabilistic and systems are subject to stochastic phenomena. For instance, in distributed systems, situations such as "the message will be delivered successfully with probability of 95%" and "the channel is 75% error free" are common. In multi-agent settings, it is also desirable to express properties such as "an agent knows that items could be lost with a chance of 30%". Considering quantitative aspects when modeling the system allows the assessment of the likelihood of different events. In fact, an appropriate reaction to an event depends on the confidence one would have about the occurrence of that event. For instance, if the agent knows that the message will be successfully delivered with a probability 0.8, then she should consider other ways such as sending duplicate copies. Accounting for stochastic phenomena in epistemic systems, which are the main focus of this paper, and verifying their correctness are important aspects in concrete applications [3,10,24,22,49].

There are two main frameworks for representing and reasoning about epistemic systems: Partially Observable Markov Decision Processes (POMDPs) and interpreted systems. On the one hand, POMDPs, which are a generalization of Markov Decision Processes (MDPs), have been used to model the uncertainty of knowledge and behavior for stochastic agents since the 1990s [8,18,26,27]. Recently, POMDPs have been used extensively in machine learning [1,42,12], agent decision making [38], and robotic applications [28,43]. In the POMDPs-based framework, agents only observe the underlying states partially and maintain a probability

☆ This manuscript is an extension of a previous version accepted for publication at SoMeT_2012 [50]. The new version extends the previous one by: (1) extending the discussion of related work (Section 2); (2) refining and extending the theoretical results (Section 4); (3) extending the model checking technique (Section 5); (4) adding a complete case study along with its implementation, the performance analysis, and comparison with two existing model-checkers' performances (Section 6); and (5) adding new examples and illustrations in different sections.

* Corresponding author.

E-mail addresses: w_wan@encs.concordia.ca (W. Wan), bentahar@encs.concordia.ca (J. Bentahar), hamza@encs.concordia.ca (A. Ben Hamza).

[1] This work has been performed while Jamal Bentahar is visiting Khalifa University of Science, Technology, and Research, UAE.

0950-7051/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.knosys.2013.06.017

distribution over the set of possible states, called belief states, which are computed based on a set of observations. On the other hand, the interpreted systems formalism [16] that formalizes agent models has proven its value in representing, modeling and verifying epistemic systems [46]. Using interpreted systems, epistemic logics to reason about knowledge and time are thoroughly investigated and extensively used in specifying and verifying multi-agent systems [34,35,37,39,46]. Epistemic modalities have been developed to represent not only an individual agent's knowledge, but also a group's knowledge such as common knowledge in different models of time including linear and branching [45]. They have also been investigated within a first order logic where quantified interpreted systems are introduced to define a semantics to reason about knowledge and time in a first order setting [3]. Soundness and completeness issues are discussed in this paper. However, the quantifications in this work are first order quantifications, and not uncertainty quantifications as we propose in our work. In fact, using interpreted systems to specify agents' uncertainty of knowledge is still in the early stages and verifying agents' probabilistic (i.e., uncertain) knowledge using interpreted systems is still a fertile research topic. In this paper, we aim to use knowledge representation techniques through the definition of a new logic and interpreted systems to express not only qualitative, but also quantitative and uncertain knowledge and investigate the model checking of the defined logic.

To summarize, there are two ways of representing and reasoning about stochastic epistemic systems: POMDPs and extension of interpreted systems with probabilistic and uncertain behavior. The first option has been widely studied. However, the second option is yet to be investigated. The purpose of this paper is to examine this option, not only from the knowledge representation perspective, but also from the verification and model checking points of view. This choice is motivated by the fact that interpreted systems provide a natural and elegant way of capturing the philosophical foundations of knowledge using possible and accessible worlds, agent local states, and system global states. Simply put, in this formalism, an agent's knowledge is captured by the information stored in all local equivalent states of the current state, which means states that the agent cannot distinguish. Thus, an agent knows φ in a given state iff φ is true in all the equivalent local states of that state (those states are said to be possible or accessible). Such a rich interpretation is not captured by POMDPs.

There are two questions that must be answered in order to check uncertain, quantitative-epistemic properties: how to specify measurable epistemic properties and how to represent models capturing measurable epistemic features. Uncertain knowledge can be represented using probabilities and the multi-agent system can be modeled as a probabilistic Kripke-like model. In fact, the multi-agent system is a distributed probabilistic knowledge-based system where components are autonomous and selfish. In this paper, we integrate Markov chains structure into interpreted systems to express probabilistic multi-agent systems. To specify the quantitative properties of these systems, we build on and extend our previous work [49,50], in which a probabilistic- epistemic logic was proposed via the combination of temporal and epistemic logics at the probabilistic level, by adding the degree of epistemic properties.

The contributions of this paper are twofold. First, we define probabilistic–epistemic logic PCTLK. PCTLK not only allows probabilities of paths (i.e. runs), but also represents quantified and uncertain knowledge. Discrete-Time Markov Chains (DTMCs) integrated into interpreted systems are used to model multi-agent systems. DTMCs are widely used to model systems with probability information and are formal models of PCTL [2], the probabilistic extension of computation tree logic CTL. The second contribution is the reduction of PCTLK model checking to PCTL model checking. This reduction is achieved by transforming the models of PCTLK into MDPs, which are then transformed to DTMCs using the notion of *scheduler* [32]. We show that a PCTLK formula is satisfied in a model of PCTLK iff a corresponding PCTL formula is satisfied in a DTMC model of PCTL. By doing so, formulae of PCTLK can be simply checked using PRISM [31], the model checker of PCTL.

This paper is organized as follows. Section 2 discusses and compares relevant related work on modeling and specifying knowledge and probability. In Section 3, we present the models and introduce probabilistic interpreted systems. We define a new logic PCTLK in Section 4 and state its syntax and semantics. In Section 5, we explain how model checking PCTLK can be reduced to model checking PCTL. We implement our approach with PRISM [31] and apply it to a case study in Section 6. In the same section, we experimentally compare our work with other related approaches and show that our approach outperforms the others in terms of both execution time and space. Finally, we summarize the paper and suggest further work in Section 7.

2. Related work

As an automatic verification technique at design time, model checking has been used to verify different desirable properties, such as deadlock freedom, safety, and reachability. Recently, this technique has been used to verify, in a static way, if composite Web services design models satisfy such properties, which allows us to check the soundness and completeness of the models [6]. Unlike our work that aims at proposing a new probabilistic–epistemic logic and a new model checking technique for the underlying models, the authors in [6] simply expressed the desired properties in the existing CTL and LTL logics with no knowledge operator, and used the classic (non-probabilistic) symbolic model checking technique to perform the verification.

Model checking epistemic logic from agent programming perspective has been investigated by Dennis et al. in [11]. The authors proposed a framework for verifying agent-based solutions. There are two components in their framework: the agent infrastructure layer, which is a set of Java classes designed to interpret belief, desire, and intension agent programming languages, and the agent Java pathfinder, which is an extended Java Pathfinder (JPF) model checker for agent programs. This framework emphasizes the verification of agents' beliefs, plans, and goals. However, uncertainty has not been considered in this work.

Dealing with uncertainty within distributed knowledge bases has been recently addressed by some researchers. Lawry and Tang in [33] use valuation pairs, which represent absolutely true and not absolutely false as a model of truth-gaps for propositional logic sentences. Instead of two proposition values of either absolutely true or absolutely false, valuation pairs set three-value propositions: true, borderline, and false. A sentence having a value, which is neither absolutely true nor absolutely false is borderline. This allows agents to consider uncertain and vague propositions. However, this logic is limited as it cannot express probability values over propositions as we propose in our logic. Moreover, practical model checking of this three-value logic is a complex procedure because the borderline value cannot be mapped to true so that the corresponding model states can be returned by the model checking algorithm. Such an algorithm is yet to be proposed. A related work has been investigated by Khan and Banerjee in [29] by proposing a logic for multiple-source approximation systems where the agent knowledge base is distributed. The authors used the theory of rough sets to define an approximation space, in which a domain of discourse and an equivalence relation on this domain are paired. Based on this approximation space, the lower and upper approximation can be computed. To express the properties related