Checking Multi-Agent Systems against Temporal-Epistemic Specifications

Ran Chen[1,2] , **Wenhui Zhang**[1,2]
[1]SKLCS, Institute of Software, Chinese Academy of Sciences
[2]University of Chinese Academy of Sciences

Abstract

This work is on methodologies for checking multi-agent systems against temporal-epistemic specifications. Since behaviors that comply with temporal-epistemic specifications in general (or theoretically) involve infinite sequences of actions of the involved agents, we must avoid checking such specifications based on such infinite sequences. This work at the theoretical side provides a bounded semantics for the temporal-epistemic specification formalism CTLK such that checking an infinite number of steps can be reduced to checking a finite number of steps, and at the practical side develops an approach based on the semantics and QBF-solving techniques for the verification purpose. The approach has been implemented and experimental data show that there exist verification problems that can be verified by this new approach more efficiently than BDD based symbolic model checking.

1 Introduction

Multi-agent systems are concurrent systems composed of multiple interacting agents within an environment [Shoham and Leyton-Brown, 2008; Wooldridge, 2009]. They can be used to solve problems that are difficult for an individual agent to solve, and can be deployed in safety, mission, or business critical scenarios, and therefore methodologies for checking the behavior of such systems are of great importance. Since knowledge is the main concept to model intelligence, reasoning about knowledge plays an important role in analyzing multi-agent systems. For such a purpose, many formalisms based on formal logics have been studied. Modal logic is generally used to specify agents. Many extensions of modal logic, such as combining epistemic logic with other modal logics, have been developed to specify multi-agent systems. Model checking against LDLK which is a logic combining epistemic logic with dynamic logic has been studied in [Kong and Lomuscio, 2018; Kong and Lomuscio, 2017a]. There is also work about model checking against ATLK [Lomuscio and Michaliszyn, 2016] which combines epistemic logic with strategy logic ATL [Belardinelli *et al.*, 2018]. Another variant is to combine epistemic logic with temporal logic [Kong and Lomuscio, 2017b; Penczek *et al.*, 2012; Meski *et al.*, 2014] and it has been widely used to model and reason about multi-agent systems. To verify multi-agent systems which use temporal epistemic logic as their specification language, a number of model checking methods are adapted to check the properties of such systems. Such methods include BDD-based model checking, bounded model checking, unbounded model checking, and many tools based on such methods have been developed over the years. In [Penczek and Lomuscio, 2003], bounded semantics has been developed and a bounded model checking approach has been proposed, and verification based on fixpoint computations has been proposed in [Kacprzak *et al.*, 2004b]. Verification based on BDDs was discussed in [Gammie and Meyden, 2004; Meyden and Su, 2004; Raimondi and Lomuscio, 2007], and the complexity issues discussed in [Lomuscio and Raimondi, 2006]. For dealing with the efficiency of verification, integration of abstraction [Dam *et al.*, 2009], symmetry reduction [Cohen *et al.*, 2009], and parallel executions [Kwiatkowska *et al.*, 2010] have been proposed.

This work focuses on the basic methodologies for checking multi-agent systems against temporal-epistemic specifications, in particular, CTLK, a computation tree logic of knowledge [Penczek and Lomuscio, 2003]. The behaviors that comply with temporal-epistemic specifications involve infinite sequences of actions such that we must avoid checking such specifications based on such infinite sequences. For the existential fragment of CTLK (denoted ECTLK), bounded semantics that tries to reduce the problem of checking the infinite behavior to checking a finite number of steps has already been suggested [Penczek and Lomuscio, 2003], however bounded semantics for the full CTLK has not been available.

For successfully defining such a bounded semantics, we make the use of a combination of paths on the state transition relations and knowledge-paths on epistemic relations, and the contributions of this work are as follows.

- A bounded semantics for CTLK such that for every specification in CTLK, we are able to use bounded semantics to check whether the behavior of a multi-agent system is consistent with CTLK specifications;
- A verification approach based on the bounded semantics

s, such that the problem of checking multi-agent systems against CTLK properties is reduced into the problem of checking the validity of QBF formulas;

- An implementation of the approach, such that the complementary nature of such an approach and BDD based symbolic model checking is demonstrated based on experimental analysis of test cases.

2 Preliminaries

Multi-agent systems may be represented by interpreted systems [Fagin *et al.*, 2004]. In such a representation, at any point of time, each of the agents is in some *state* referred to as the agent's *local* state. In addition, the *environment* is included and viewed as "everything else that is relevant". The *environment* can be omitted if necessary. Assume a set of agents $A = \{1, \dots, n\}$, a set of local states L_i for each agent $i \in A$, and a set L_e of local states for the environment. The set of possible global states for the system is then defined as $G \subseteq L_1 \times \cdots \times L_n \times L_e$, where each element $(l_1, \dots, l_n, l_e)$ of G represents a computational state for the system.

A transition occurs when some *actions* are performed, and those actions are performed according to some *protocol* which is encoded by a function $P_i : L_i \to 2^{Act_i}$ for each agent $i \in A$, and a function $P_e : L_e \to 2^{Act_e}$ for the environment. Then we can have the *transition function* $t : G \times Act \to G$, where $Act \subseteq Act_1 \times \cdots \times Act_n \times Act_e$ is the set of joint actions. A *joint protocol* P is a tuple $(P_1, \dots, P_n)$ consisting of protocols P_i for $i \in A$. P_e is not included in the joint protocol while Act_e is in the joint action. This is because when designing multi-agent systems, the environment is often viewed as an opposing player to the agents. There is also a set I of initial states. We assume a set AP of primitive propositions and a function L which assigns truth values to the primitive propositions on every state.

2.1 Kripke Structures

For representation of multi-agent systems facilitating automated analysis of temporal-epistemic properties of the sytems, we may use a Kripke structure, which can be constructed from interpreted systems [Lomuscio and Ryan, 1997].

Definition 2.1. *Let $A = \{1, \dots, n\}$ be a set of agents and AP be a set of propositions. A Kripke structure over A and AP is represented as $M = (S, T, I, \sim_1, \dots, \sim_n, L)$, where S is a finite set of the reachable global states in the interpreted system; $T \subseteq S \times S$ is a total transition relation (serial relation) on S which is determined by the transition function, joint actions and the joint protocol in the interpreted system; $I \subseteq S$ is a non-empty set of initial states; for each $i \in A$, $\sim_i \subseteq S \times S$ is an equivalence relation on S, that is, a set of pairs of elements of S which is defined by $s \sim_i s'$ iff $l_i(s) = l_i(s')$ where the function $l_i : S \to L_i$ returns the local state of agent i from a global state s; and $L : S \to 2^{AP}$ is a valuation function that maps each state to a subset of propositions of AP.*

A Kripke structure is also called a model.

We use $|M|$ to denote the size of the model, which is defined as the number of states in S. The relation $\sim_i$ is referred to as the accessibility relation for agent i. Notice that the set S is the set of reachable states of the intended interpreted system, which is consistent with the definition of the Kripke structures in [Penczek and Lomuscio, 2003]. This simplifies the model in the sense that we can focus the reasoning on the epistemic properties without arguing whether a state is reachable. A Kripke structure $M = (S, T, I, \sim_1, \dots, \sim_n, L)$ is usually viewed as a directed graph (S, T) with additional information on the set of agents. According to this point of view, state transition paths (or simply, paths) on such a graph are defined as follows.

A *path* in M is an infinite sequence of states $\pi = \pi_0\pi_1 \dots$ such that $(\pi_u, \pi_{u+1}) \in T$ for each $u \in \mathbf{N}$. A finite path π of M is a finite prefix of an infinite path of M. We use $\pi(s)$ to denote a path π with $\pi_0 = s$. A *computation* of M is an infinite path $\pi = \pi_0\pi_1 \dots$ of M such that $\pi_0 \in I$.

On the other hand, $M = (S, T, I, \sim_1, \dots, \sim_n, L)$ can also be viewed as an epistemic graph with its nodes being S, and with $\sim_i$ as edges labeled by agent i for $i \in A$. The epistemic graph describes the epistemic relation of the system. Following this point of view, we define common-knowledge-paths as follows.

A *common-knowledge-path*, CK-path for short, for a group Γ of agents is an infinite sequence of states $\xi^{C_\Gamma} = \xi_0\xi_1 \dots$ such that $(\xi_u, \xi_{u+1}) \in \bigcup_{i \in \Gamma} \sim_i$ for each $u \in \mathbf{N}$. We use $\xi^{C_\Gamma}(s)$ to denote a CK-path ξ^{C_Γ} with $\xi_0 = s$.

2.2 The Temporal-Epistemic Logic CTLK

Computation Tree Logic of Knowledge, CTLK for short [Penczek and Lomuscio, 2003], is a logic that uses CTL [Clarke and Emerson, 1981; Emerson and Clarke, 1982] as a basic temporal language and extends it with epistemic components.

Definition 2.2 (Syntax of CTLK)**.** *Let AP be a set of propositional variables and A a set of agents. Let p range over AP, i over A, Γ over subsets of A. The set of CTLK formulas ϕ (written in negation normal form) is defined by the following BNF syntax.*

$$\begin{array}{rcl} \phi & ::= & p \mid \neg p \mid \phi \wedge \phi \mid \phi \vee \phi \mid \\ & & AX\phi \mid A(\phi R \phi) \mid A(\phi U \phi) \mid \\ & & EX\phi \mid E(\phi R \phi) \mid E(\phi U \phi) \mid \\ & & K_i\phi \mid E_\Gamma\phi \mid C_\Gamma\phi \mid D_\Gamma\phi \mid \\ & & \overline{K}_i\phi \mid \overline{E}_\Gamma\phi \mid \overline{C}_\Gamma\phi \mid \overline{D}_\Gamma\phi \end{array}$$

In addition, we may use $EF\phi, EG\phi, AF\phi, AG\phi$ as abbreviations for $E(\top U \phi), E(\neg\top R\phi)$, $A(\top U \phi)$, $A(\neg\top R\phi)$ where $\top$ represents $p \vee \neg p$ (meaning $true$) for some $p \in AP$.

We may as well also use $\neg\phi$ as an abbreviation of the negation normal form formula equivalent to $\neg\phi$.

Definition 2.3 (Semantics of CTLK)**.** *Let M be a model, $s \in S$ a state, and φ and ψ formulas of CTLK. Let $\pi = \pi_0\pi_1 \dots$ denote an infinite path of M, and $\xi^{C_\Gamma} = \xi_0\xi_1 \dots$ denote an infinite CK-path of M. The relation $M, s \models \varphi$ denotes that φ is true at the state s in the model M. The relation $\models$ is defined in Table 1.*

This definition of the semantics of CTLK is slightly different from that defined in [Penczek and Lomuscio, 2003] for

Table 1: Semantics of CTLK

$M, s \models p$	iff $p \in L(s)$, for $p \in AP$
$M, s \models \neg\varphi$	iff $M, s \not\models \varphi$
$M, s \models \varphi \wedge \psi$	iff $(M, s \models \varphi)$ and $(M, s \models \psi)$
$M, s \models \varphi \vee \psi$	iff $(M, s \models \varphi)$ or $(M, s \models \psi)$
$M, s \models AX\varphi$	iff $\forall\pi(s).(M, \pi_1 \models \varphi)$
$M, s \models A(\varphi U \psi)$	iff $\forall\pi(s).(\exists u \geq 0.$ $(M, \pi_u \models \psi \wedge \forall w < u.(M, \pi_w \models \varphi)))$
$M, s \models A(\varphi R \psi)$	iff $\forall\pi(s).((\forall u \geq 0.$ $(M, \pi_u \models \psi \vee \exists w < u.(M, \pi_w \models \varphi)))$
$M, s \models EX\varphi$	iff $\exists\pi(s).(M, \pi_1 \models \varphi)$
$M, s \models E(\varphi U \psi)$	iff $\exists\pi(s).(\exists u \geq 0.$ $(M, \pi_u \models \psi \wedge \forall w < u.(M, \pi_w \models \varphi)))$
$M, s \models E(\varphi R \psi)$	iff $\exists\pi(s).((\forall u \geq 0.$ $(M, \pi_u \models \psi \vee \exists w < u.(M, \pi_w \models \varphi)))$
$M, s \models K_i\varphi$	iff $\forall s' \in S.((s, s') \in \sim_i \rightarrow (M, s' \models \varphi))$
$M, s \models D_\Gamma\varphi$	iff $\forall s' \in S.((s, s') \in \bigcap_{i\in\Gamma} \sim_i \rightarrow (M, s' \models \varphi))$
$M, s \models E_\Gamma\varphi$	iff $\forall s' \in S.((s, s') \in \bigcup_{i\in\Gamma} \sim_i \rightarrow (M, s' \models \varphi))$
$M, s \models C_\Gamma\varphi$	iff $\forall\xi^{C_\Gamma}(s).\forall u \geq 0.(M, \xi_u \models \varphi)$
$M, s \models \overline{K}_i\varphi$	iff $\exists s' \in S.((s, s') \in \sim_i \wedge (M, s' \models \varphi))$
$M, s \models \overline{D}_\Gamma\varphi$	iff $\exists s' \in S.((s, s') \in \bigcap_{i\in\Gamma} \sim_i \wedge (M, s' \models \varphi))$
$M, s \models \overline{E}_\Gamma\varphi$	iff $\exists s' \in S.((s, s') \in \bigcup_{i\in\Gamma} \sim_i \wedge (M, s' \models \varphi))$
$M, s \models \overline{C}_\Gamma\varphi$	iff $\exists\xi^{C_\Gamma}(s).\exists u \geq 0.(M, \xi_u \models \varphi)$

the epistemic operator $C_\Gamma\varphi$ and $\overline{C}_\Gamma\varphi$. In this definition, we have avoided the use of the transitive closure of the union of the accessibility relations of the agents of Γ. Instead, we use CK-paths. This follows from that we view the Kripke model as two types of graphs, and this viewpoint allows us to manipulate temporal operators and epistemic operators on different graphs and simplify the formulation of the bounded semantics and the verification of CTLK that we will demonstrate later on. The definition is consistent with that defined in [Fagin *et al.*, 2004] as follows.

$$M, s \models C_\Gamma\varphi \text{ iff } M, s \models E^k_\Gamma\varphi \text{ for } k = 1, 2, \ldots$$

The above definition has a graph-theoretical interpretation, that is, $M, s \models C_\Gamma\varphi$ iff $M, t \models \varphi$ for all the states t that exists a path from s to t whose edges are labeled by members of Γ. We refer to Lemma 2.2.1 in [Shoham and Leyton-Brown, 2008] for details. Thus in our case, the definitions of $M, s \models C_\Gamma\varphi$ and $M, s \models \overline{C}_\Gamma\varphi$ in Table 1 are consistent with the graph-theoretical interpretation.

Definition 2.4. *A CTLK formula φ is valid in M, denoted $M \models \varphi$, if φ is true at all initial states of M.*

3 Bounded Semantics

The semantics defined in Table 1 involves infinite paths. In this section, we define bounded semantics for CTLK such that we can check whether a property is satisfied by only looking at finite paths. This formalism can be seen as an extension of that of ECTLK [Penczek and Lomuscio, 2003] and also of that of CTL [Zhang, 2015]. For the epistemic operators, we define their bounded semantics on knowledge-paths, which are paths build on epistemic relation of the system. Thus, before defining the bounded semantics, we define a set of notions as follows.

Distributed-Knowledge-Paths Since distributed knowledge $D_\Gamma\varphi$ is defined upon a different epistemic relation, we define another type of knowledge-path on the epistemic graph, i.e., distributed-knowledge-path (DK-paths), defined as follows. A DK-path for a group Γ of agents is an infinite sequence of states $\xi^{D_\Gamma} = \xi_0\xi_1\ldots$ such that $(\xi_u, \xi_{u+1}) \in \bigcap_{i\in\Gamma} \sim_i$ for each $u \in \mathbf{N}$. We use $\xi^{D_\Gamma}(s)$ to denote a DK-path ξ^{D_Γ} with $\xi_0 = s$.

Knowledge-Paths For convenience, both CK-paths and DK-paths are referred to as knowledge-paths.

k-Paths and Paths with Repeating States (rs-Paths) A k-path is a finite path with length $k + 1$ [Biere *et al.*, 1999]. Likewise, we may use k-CK-path and k-DK-path to denote CK-path and DK-path with length $k + 1$.

To specify whether a path has a state that occurs more than once, we define rs-path as a path that has a state that occurs at least twice. Let $|\pi|$ be the length of π. Formally, we have the following definition of $rs(\pi)$.

$$rs(\pi) := \bigvee_{u=0}^{|\pi|-1} \bigvee_{w=u+1}^{|\pi|-1} \pi_u = \pi_w$$

If π is a prefix of π', then $rs(\pi) \rightarrow rs(\pi')$. In this paper, this notion applies to the usual paths as well as to the knowledge-paths.

On Defining Bounded Semantics The basic idea of bounded semantics is to identify a witness for a property with a given bound on the length of the paths. The main objective is then to define a bounded semantics suitable for arguing that if a property is satisfied on all paths with a given bound (with the bounded semantics), then it is satisfied on all infinite paths (with the original semantics). On the other hand (though it is easier to handle), we also need that if a property is satisfied on some path with a given bound, then it is satisfied on some infinite path.

Definition 3.1 (Bounded Semantics of CTLK)**.** *Let M be a Kripke model, $s \in S$ be a state, φ and ψ be CTLK formulas. $\models_k$ denotes the bounded semantics with respect to $\models$. Let $k \geq 1$. Let $\pi = \pi_0 \ldots \pi_k$ denote k-paths, $\xi^{D_\Gamma} = \xi_0 \ldots \xi_k$ and $\xi^{C_\Gamma} = \xi_0 \ldots \xi_k$ denote k-knowledge-paths in which $\Gamma \subseteq A$. The semantics relation $M, s \models_k \varphi$ is defined in Table 2.*

Notice that we defined epistemic operators on knowledge-paths rather than on normal transition paths as in [Penczek and Lomuscio, 2003]. In this way, the semantics of the operators namely K, $\overline{K}$, D, $\overline{D}$, E, $\overline{E}$ are similar to the semantics of temporal operator X which would simplify the verification because only the next states on the knowledge-paths need to be checked. The bounded semantics of $K_i\varphi$ and $\overline{K}_i\varphi$ is defined on CK-paths with Γ being exactly $\{i\}$. This can as well be done on DK-paths with Γ being $\{i\}$, since the two types of knowledge-paths coincide when there is only one agent in Γ. The relation between the bounded semantics and the standard semantics is to be established as follows.

Remark The proofs of the following lemmas are all based on structural induction, and since the bounded semantics can be seen as an extension of the one for CTL and the treatment of propositional formulas and temporal operators is similar to that in [Zhang, 2015], we only consider the knowledge operators in the proofs of the following lemmas.

Table 2: Bounded Semantics of CTLK

$M, s \models_k p$	iff $p \in L(s)$, for $p \in AP$
$M, s \models_k \neg p$	iff $p \notin L(s)$
$M, s \models_k \varphi \wedge \psi$	iff $(M, s \models_k \varphi)$ and $(M, s \models_k \psi)$
$M, s \models_k \varphi \vee \psi$	iff $(M, s \models_k \varphi)$ or $(M, s \models_k \psi)$
$M, s \models_k AX\varphi$	iff $\forall \pi(s).(M, \pi_1 \models_k \varphi)$
$M, s \models_k A(\varphi U \psi)$	iff $\forall \pi(s).(\exists u \leq k.$ $(M, \pi_u \models_k \psi \wedge \forall w < u.(M, \pi_w \models_k \varphi)))$
$M, s \models_k A(\varphi R \psi)$	iff $\forall \pi(s).((rs(\pi) \wedge \forall u \leq k.(M, \pi_u \models_k \psi)) \vee$ $\exists u \leq k.(M, \pi_u \models_k \varphi \wedge \forall w \leq u.(M, \pi_w \models_k \psi)))$
$M, s \models_k EX\varphi$	iff $\exists \pi(s).(M, \pi_1 \models_k \varphi)$
$M, s \models_k E(\varphi U \psi)$	iff $\exists \pi(s).(\exists u \leq k.$ $(M, \pi_u \models_k \psi \wedge \forall w < u.(M, \pi_w \models_k \varphi)))$
$M, s \models_k E(\varphi R \psi)$	iff $\exists \pi(s).((rs(\pi) \wedge \forall u \leq k.(M, \pi_u \models_k \psi)) \vee$ $\exists u \leq k.(M, \pi_u \models_k \varphi \wedge \forall w \leq u.(M, \pi_w \models_k \psi)))$
$M, s \models_k K_i\varphi$	iff $\forall \xi^{C_{\{i\}}}(s).(M, \xi_1 \models_k \varphi)$
$M, s \models_k D_\Gamma\varphi$	iff $\forall \xi^{D_\Gamma}(s).(M, \xi_1 \models_k \varphi)$
$M, s \models_k E_\Gamma\varphi$	iff $\forall \xi^{C_\Gamma}(s).(M, \xi_1 \models_k \varphi)$
$M, s \models_k C_\Gamma\varphi$	iff $\forall \xi^{C_\Gamma}(s).(rs(\xi) \wedge \forall u \leq k.(M, \xi_u \models_k \varphi))$
$M, s \models_k \overline{K}_i\varphi$	iff $\exists \xi^{C_{\{i\}}}(s).(M, \xi_1 \models_k \varphi)$
$M, s \models_k \overline{D}_\Gamma\varphi$	iff $\exists \xi^{D_\Gamma}(s).(M, \xi_1 \models_k \varphi)$
$M, s \models_k \overline{E}_\Gamma\varphi$	iff $\exists \xi^{C_\Gamma}(s).(M, \xi_1 \models_k \varphi)$
$M, s \models_k \overline{C}_\Gamma\varphi$	iff $\exists \xi^{C_\Gamma}(s).(\exists u \leq k.(M, \xi_u \models_k \varphi))$

Lemma 3.1. *Let $k \geq 1$. If $M, s \models_k \varphi$, then $M, s \models_{k+1} \varphi$.*

Proof by induction. The knowledge operators are dealt with as follows.

- For $E_\Gamma\varphi$, suppose that $M, s \models_k E_\Gamma\varphi$ holds and $M, s \models_{k+1} E_\Gamma\varphi$ does not hold. Then there exists a $(k+1)$-knowledge-path $\xi^{C_\Gamma}(s)$ such that $M, \xi_1 \models_{k+1} \varphi$ doesn't hold. On the other hand, since $\xi_0^{C_\Gamma} \cdots \xi_k^{C_\Gamma}$ is a k-knowledge-path, and therefore according to $M, s \models_k E_\Gamma\varphi$, we have $M, \xi_1 \models_k \varphi$. By the induction hypothesis, we have $M, \xi_1 \models_{k+1} \varphi$ and therefore a contradiction.
- For $K_i\varphi$ and $D_\Gamma\varphi$, the proof is the similar to the case of $E_\Gamma\varphi$, except that Γ is $\{i\}$ in the case of $K_i\varphi$, and the knowledge-path is ξ^{D_Γ} in the case of $D_\Gamma\varphi$.
- For $C_\Gamma\varphi$, suppose that $M, s \models_k C_\Gamma\varphi$ holds and $M, s \models_{k+1} C_\Gamma\varphi$ does not hold. Then there is a $(k+1)$-knowledge-path $\xi^{C_\Gamma}(s)$ such that $rs(\xi) \wedge \forall u \leq (k+1).(M, \xi_u \models_{k+1} \varphi)$ does not hold.

  We consider two cases.

  (a) $rs(\xi)$ does not hold.

  Then $rs(\xi_0 \cdots \xi_k)$ does not hold. Since $M, s \models_k C_\Gamma\varphi$, for every k-CK-path ξ' starting at s, $rs(\xi')$ holds. This contradicts to that $rs(\xi)$ does not hold.

  (b) $rs(\xi)$ holds and there is a u such that $M, \xi_u \models_{k+1} \varphi$ does not hold.

  We consider two subcases.

  (b1) $u \leq k$.

  Then ξ_u is in $\xi_0 \cdots \xi_k$. Since $M, s \models_k C_\Gamma\varphi$, for every k-CK-path ξ' starting at s, $\forall v \leq k.(M, \xi_v \models_k \varphi)$ holds, and therefore $\forall v \leq k.(M, \xi_v \models_{k+1} \varphi)$ holds by the induction hypothesis. This contradicts to that $M, \xi_u \models_{k+1} \varphi$ does not hold.

  (b2) $u = k + 1$.

  Since $rs(\xi)$ holds, there are $x < y \leq k + 1$ such that $\xi_x = \xi_y$. In this case we can construct a k-CK-path ξ' such that $\xi_0 \cdots \xi_{x-1}\xi_y \cdots \xi_{k+1}$ is a prefix of ξ' and ξ_u appears in ξ'. Similar to the argument of the case $(b1)$, we have $\forall v \leq k.(M, \xi'_v \models_{k+1} \varphi)$ holds and therefore a contradiction to that $M, \xi_u \models_{k+1} \varphi$ does not hold.
- For $\overline{E}_\Gamma\varphi$, suppose that $M, s \models_k \overline{E}_\Gamma\varphi$ holds. Then there is a k-knowledge-path $\xi^{C_\Gamma}(s)$ and $M, \xi_1 \models_k \varphi$. By the induction hypothesis, we have $M, \xi_1 \models_{k+1} \varphi$. Since the epistemic relations $\sim_i$ for $i \in \Gamma$ are all equivalence relations, we can extend $\xi^{C_\Gamma}(s)$ to a $k+1$ knowledge-path $(\xi')^{C_\Gamma}(s)$ such that $M, \xi'_1 \models_{k+1} \varphi$. Therefore, we have $M, s \models_{k+1} \overline{E}_\Gamma\varphi$
- For $\overline{K}_i\varphi$ and $\overline{D}_\Gamma\varphi$, the proof is similar to the case of $E_\Gamma\varphi$, except that Γ is $\{i\}$ in the case of $\overline{K}_i\varphi$, and the knowledge-path is ξ^{D_Γ} in the case of $\overline{D}_\Gamma\varphi$.
- For $\overline{C}_\Gamma\varphi$, suppose that $M, s \models_k \overline{C}_\Gamma\varphi$ holds. Then there is a k-knowledge-path $\xi^{C_\Gamma}(s) = \xi_0 \ldots \xi_k$ and a state ξ_u with $u \leq k$ such that $M, \xi_u \models_k \varphi$. By the induction hypothesis, we have $M, \xi_u \models_{k+1} \varphi$. By extending ξ to a $(k+1)$-knowledge-path ξ', we have $\exists u \leq k + 1.(M, (\xi')_u \models_{k+1} \varphi)$. Therefore we have $M, s \models_{k+1} \overline{C}_\Gamma\varphi$.

□

For an intuitive understanding of the fact that $M, s \models_k \varphi$ implies $M, s \models_{k+1} \varphi$, we may take the case $\varphi = C_\Gamma\varphi_0$ as an example, since C_Γ is the most complicated knowledge operator.

Following from $M, s \models_k C_\Gamma\varphi_0$, we have that (c1) φ_0 holds on every position on every k-CK-path ξ^{C_Γ} (starting at s) and that (c2) every such ξ^{C_Γ} satisfies $rs(\xi^{C_\Gamma})$, i.e., there are at least two states on the path that are the same.

By c2, every state on any position j on a $(k+1)$-CK-path (or an infinite CK-path) $s_0s_1s_2 \cdots s_j \cdots$ (where $s_0 = s$) is on some position on some k-CK-path starting at s. The reason is as follows. (a) if j is less than or equal to k, then $s_0s_1s_2 \cdots s_j$ is a prefix of some k-CK-path starting at s; (b) otherwise, the path $s_0s_1s_2 \cdots s_j$ must have at least two states that are the same, and then it can be compressed to a shorter path by removing the circuiting part such that (or repeatedly until) all states in the shorter path are different, and the path (still with s_j being the last state) becomes necessarily a prefix of some k-CK-path starting at s.

By c1, every state on every k-CK-path starting at s satisfies φ_0.

Therefore every state on any position j on a $(k+1)$-CK-path starting at s satisfies φ_0. Therefore $M, s \models_k C_\Gamma\varphi_0$ implies $M, s \models_{k+1} C_\Gamma\varphi_0$.

Similar arguments also apply to temporal operators such as AR and the derived operator AG.

Example Assume that we have a Kripke structure of a MAS with agents 1 and 2 shown in Figure 1 where the transition relation T is omitted, since the transition relation is not relevant to this particular example with the given property.

Consider the problem $M, s_0 \models C_{\{1,2\}}p$. To prove this, we first take $k = 1$. Remember that the bounded semantics of

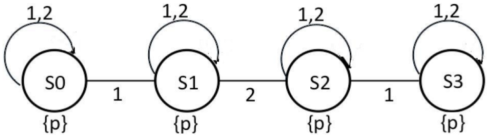

Figure 1: A Kripke structure $\{S, T, I, \sim_1, \sim_2, L\}$ with reachable states S being $\{s_0, s_1, s_2, s_3\}$, and equivelance relations shown with the labeling lines.

common knowledge requires all the k-knowledge-paths to be rs-paths. But 1-knowledge-path $\{s_0, s_1\}$ is not an rs-path. Thus we increase k. For the same reason, we keep increasing k until $k = 4$. Since all the 4-knowledge-paths are rs-paths, the property is verified. For comparison, we also consider the negated property $\overline{C}_{\{1,2\}}\neg p$, and we have the following table.

Property	$k = 1$	$k = 2$	$k = 3$	$k \geq 4$
$\overline{C}_{\{1,2\}}p$	*F*	*F*	*F*	*T*
$\overline{C}_{\{1,2\}}\neg p$	*F*	*F*	*F*	*F*

On the other hand, if we change the label on s_2 to $\{\}$, i.e., p is false on s_2, we have the following table.

Property	$k = 1$	$k = 2$	$k = 3$	$k \geq 4$
$\overline{C}_{\{1,2\}}p$	*F*	*F*	*F*	*F*
$\overline{C}_{\{1,2\}}\neg p$	*F*	*T*	*T*	*T*

Lemma 3.1 is a statement of the incremental nature of the bounded semantics. With this lemma in hand, we are ready to prove the following lemma relating satisfiability under $\models_k$ to satisfiability under the standard semantics $\models$.

Lemma 3.2. *If $M, s \models_k \varphi$ for some $k \geq 1$, then $M, s \models \varphi$.*

Proof by Induction. According to Lemma 3.1, if $M, s \models_k \varphi$ for some $k \geq 1$, then $M, s \models_{k'} \varphi$ holds for a larger k'. Let k' be the larger number of $|M|$ and k. We have $M, s \models_{k'} \varphi$.

- For $E_\Gamma\varphi$, suppose that $M, s \models E_\Gamma\varphi$ doesn't hold while $M, s \models_{k'} E_\Gamma\varphi$ holds. Then there exists s' that $(s, s') \in \bigcup_{i\in\Gamma} \sim_i$ and $M, s' \models \varphi$ doesn't hold. On the other hand, since the relations $\sim_i$ for $i \in \Gamma$ are all equivalence relations, we can construct a k'-knowledge-path $\xi^{C_\Gamma}(s) = \xi_0\xi_1\ldots\xi_{k'}$ with $\xi_1 = s'$. Since $M, s \models_{k'} E_\Gamma\varphi$ holds and $\xi^{C_\Gamma}(s)$ is a k'-CK-path starting at s, we have $M, s' \models_{k'} \varphi$. By the induction hypothesis, we have $M, s' \models \varphi$ and therefore there is a contradiction.
- For $K_i\varphi$ and $D_\Gamma\varphi$, the proof is similar to the case of $E_\Gamma\varphi$, except that Γ is $\{i\}$ in the case of $K_i\varphi$, and the knowledge-path is ξ^{D_Γ} and hence the epistemic relation is $\bigcap_{i\in\Gamma} \sim_i$ in the case of $D_\Gamma\varphi$.
- For $C_\Gamma\varphi$, suppose that $M, s \models C_\Gamma\varphi$ doesn't hold while $M, s \models_{k'} C_\Gamma\varphi$ holds. Then there exists a knowledge-path $\xi^{C_\Gamma}(s) = s\ldots s'$ where $M, s' \models \varphi$ doesn't hold. We consider two cases.

  (1) The length of $\xi^{C_\Gamma}(s)$ is less than or equal to k'.

  Let the k'-knowledge-path $(\xi')^{C_\Gamma} = s\ldots s'\ldots\xi'_{k'}$ be an extension of ξ^{C_Γ}. Since $M, s \models_{k'} C_\Gamma\varphi$ holds and $(\xi')^{C_\Gamma}$ is a k'-CK-path starting at s, we have $M, s' \models_{k'} \varphi$. By the induction hypothesis we have $M, s' \models \varphi$ and therefore there is a contradiction.

  (2) The length of $\xi^{C_\Gamma}(s)$ is greater than k'.

  Then there are duplicated states on the knowledge-path. Thus, we can contract the knowledge-path until the length of the knowledge-path is less than or equal to k'. Similar to case (1), there is a contradiction also in this case.
- For $\overline{E}_\Gamma\varphi$, suppose that $M, s \models_{k'} \overline{E}_\Gamma\varphi$. Then there exists a k'-knowledge-path $\xi^{C_\Gamma}(s)$ that $M, \xi_1 \models_{k'} \varphi$. And then we have $M, \xi_1 \models \varphi$. From the definition of the common-knowledge-path, we have $(s, \xi_1) \in \bigcup_{i\in\Gamma} \sim_i$. Hence we have $M, s \models \overline{E}_\Gamma$.
- For $\overline{K}_i\varphi$ and $\overline{D}_\Gamma\varphi$, the proof is similar to the case of $\overline{E}_\Gamma\varphi$, except that Γ is $\{i\}$ in the case of $\overline{K}_i\varphi$, and the knowledge-path is ξ^{D_Γ} and hence the epistemic relation is $\bigcap_{i\in\Gamma} \sim_i$ in the case of $\overline{D}_\Gamma\varphi$.
- For $\overline{C}_\Gamma\varphi$, suppose that $M, s \models_{k'} \overline{C}_\Gamma\varphi$. Then there exists a k'-knowledge-path $\xi^{C_\Gamma}(s)$ that $\exists u \leq k.(M, \xi_u \models_{k'} \varphi)$. By the induction hypothesis, $(M, \xi_u \models \varphi)$ holds on the k'-knowledge-path. By extending ξ to an infinite knowledge-path ξ', we have $\exists u \geq 0.(M, (\xi')_u \models_{k'} \varphi)$. Therefore we have $M, s \models \overline{C}_\Gamma\varphi$.

□

Lemma 3.3. *If $M, s \models \varphi$, then $M, s \models_k \varphi$ for some $k \geq 1$.*

Proof by induction. Let $k = |M|$. The knowledge operators are dealt with as follows.

- For $E_\Gamma\varphi$, suppose that $M, s \models_k E_\Gamma\varphi$ doesn't hold while $M, s \models E_\Gamma\varphi$ holds. Then there exists a k-knowledge-path $\xi^{C_\Gamma}(s)$ on which $M, \xi_1 \models_k \varphi$ doesn't hold. By the induction hypothesis, we have that $M, \xi_1 \models \varphi$ doesn't hold. By the definition of CK-paths, we also have $(s, \xi_1) \in \bigcup_{i\in\Gamma} \sim_i$, which contradicts $M, s \models E_\Gamma\varphi$.
- For $K_i\varphi$ and $D_\Gamma\varphi$, the proof is similar to the case of $E_\Gamma\varphi$, except that Γ is $\{i\}$ in the case of $K_i\varphi$, and the knowledge-path is ξ^{D_Γ} and hence the epistemic relation is $\bigcap_{i\in\Gamma} \sim_i$ in the case of $D_\Gamma\varphi$.
- For $C_\Gamma\varphi$, suppose that $M, s \models_k C_\Gamma\varphi$ doesn't hold while $M, s \models C_\Gamma\varphi$ holds. Then there exists a k-knowledge-path $\xi^{C_\Gamma}(s)$ that $rs(\xi)$ holds since $k = |M|$, and then there exists some $u \leq k$ that $M, \xi_u \models_k \varphi$ doesn't hold. By the induction hypothesis, $M, \xi_u \models \varphi$ doesn't hold, which contradicts $M, s \models C_\Gamma\varphi$.
- For $\overline{E}_\Gamma\varphi$, suppose that $M, s \models \overline{E}_\Gamma\varphi$. Then there exists s' that $(s, s') \in \bigcup_{i\in\Gamma} \sim_i$ and $M, s' \models \varphi$. By the induction hypothesis, we have $M, s' \models_k \varphi$. Since the relations $\sim_i$ for $i \in \Gamma$ are all equivalence relations, we can construct a k-knowledge-path $\xi^{C_\Gamma}(s) = \xi_0\xi_1\ldots\xi_k$ with $\xi_1 = s'$. Then we have $M, s \models_k \overline{E}_\Gamma\varphi$.
- For $\overline{K}_i\varphi$ and $\overline{D}_\Gamma\varphi$, the proof is similar to the case of $\overline{E}_\Gamma\varphi$, except that Γ is $\{i\}$ in the case of $\overline{K}_i\varphi$, and the epistemic relation is $\bigcap_{i\in\Gamma} \sim_i$ and hence the knowledge-path is ξ^{D_Γ} in the case of $\overline{D}_\Gamma\varphi$.

- For $\overline{C}_\Gamma\varphi$, suppose that $M, s \models \overline{C}_\Gamma\varphi$. Then there exists a knowledge-path $\xi^{C_\Gamma}(s) = s \ldots s'$ that $M, s' \models \varphi$. By the induction hypothesis, we have $M, s' \models_k \varphi$. We consider two cases.

  (1) The length of $\xi^{C_\Gamma}(s)$ is less than or equal to k.

  Then we can construct a k-knowledge-path $(\xi')^{C_\Gamma}(s) = s \ldots s' \ldots \xi'_k$ with its prefix being exactly $\xi^{C_\Gamma}(s)$. Therefore we have $M, s \models_k \overline{C}_\Gamma$.

  (2) The length of $\xi^{C_\Gamma}(s)$ is greater than k.

  Then there are duplicated states on the knowledge-path. Thus, we can contract the knowledge-path until the length of the knowledge-path is less than or equal to k. Similar to case (1), we have $M, s \models_k \overline{C}_\Gamma\varphi$ also in this case.

□

Theorem 3.1. *$M, s \models \varphi$ iff $M, s \models_k \varphi$ for some $k \geq 1$.*

Proof. Lemma 3.2 is a statement of a kind of soundness of the bounded semantics and Lemma 3.3 is a statement of completeness. This theorem is a combination of the two facts and follows from Lemmas 3.2 and Lemma 3.3. □

Definition 3.2. *Let $k \geq 1$. $M \models_k \varphi$, if $M, \iota \models_k \varphi$ for all $\iota \in I$.*

Corollary 3.1. *$M \models \varphi$ iff $M \models_k \varphi$ for some $k \geq 1$.*

We present the following two simple examples for demonstrating the application of Theorem 3.1.

Example 1 Consider the model M with $S = \{s, t, u\}$ being the set of states; $T = \{(s, t), (t, u), (u, u)\}$; $L(s) = L(t) = \{p\}$ and $L(u) = \{\}$, i.e., s and t satisfy p while u satisfies $\neg p$.

Let $\phi = E(pU\neg p)$. We have $M, s \models \phi$. This is demonstrated as follows.

For $k = 1$, both of $M, s \models_1 \phi$ and $M, s \models_1 \neg\phi$ are evaluated to false, and no conclusion can be achieved in this step.

For $k = 2$, $M, s \models_2 \phi$ is evaluated to true, while $M, s \models_2 \neg\phi$ is evaluated to false, and we conclude that $M, s \models \phi$ holds (the one that evaluates to true counts).

Example 2 Consider the model M with $S = \{s_0, s_1, s_2\}$ being the set of states; $T = \{(s_0, s_0), (s_0, s_1), (s_1, s_2), (s_2, s_2)\}$; $L(s_0) = L(s_1) = \{p\}$ and $L(s_2) = \{\}$, i.e., s_0 and s_1 satisfy p while s_2 satisfies $\neg p$.

(i) Let $\phi = AGp$. We have $M, s_0 \not\models \phi$.

This is demonstrated as follows.

For $k = 1$, $M, s_0 \models_1 \phi$ does not hold, due to that some k-path $\pi(s_0)$ does not satisfy $rs(\pi)$.

For $k \geq 2$, $M, s_0 \models_k \phi$ does not hold, due to that s_2 appears on some such k-path starting at s_0.

The two facts alone do not certify that $M, s_0 \not\models \phi$ holds. For showing $M, s_0 \not\models \phi$, we have to prove $M, s_0 \models \neg\phi$, i.e., to prove that there is a $k \geq 1$ such that $M, s_0 \models_k \neg\phi$ holds. Since it is easily seen that $M, s_0 \models_2 \neg\phi$ holds, and then it follows that we have $M, s_0 \models \neg\phi$, i.e., $M, s_0 \not\models \phi$.

(ii) For epistemic properties, we consider two agents $\{a, b\}$, and add to the previous structure the equivalence relation that partitions $\{s_0, s_1, s_2\}$ into two groups $\{s_0\}$ and $\{s_1, s_2\}$ for agent a, and that partitions $\{s_0, s_1, s_2\}$ into $\{s_0, s_1\}$ and $\{s_2\}$ for agent b.

(ii.a) Let $\phi' = C_{\{a,b\}}p$. We have $M, s_0 \not\models \phi'$.

This is demonstrated as follows.

Since the equivalence relations are used as accessibility relations, every state has at least one successor (a self-loop).

For $k = 1$, $M, s_0 \models_1 \phi'$ is false, since we have a knowledge path (s_0, s_1) that does not satisfy the rs-requirement.

For $k \geq 2$, $M, s_0 \models_k \phi'$ is false, since we have a knowledge path $(s_0, s_1, s_2, ...)$ that not all states on the path satisfy p.

Similarly, the two facts alone do not certify that $M, s_0 \not\models \phi'$ holds. For showing $M, s_0 \not\models \phi'$, we have to prove $M, s_0 \models \neg\phi'$, i.e., to prove that there is a $k \geq 1$ such that $M, s_0 \models_k \neg\phi'$ holds.

For $k = 1$, $M, s_0 \models_1 \neg\phi'$ does not hold, since all knowledge paths with 2 states are not able to reach s_2.

For $k = 2$, $M, s_0 \models_2 \neg\phi'$ holds with the witness path being (s_0, s_1, s_2).

Therefore $M, s_0 \models \neg\phi'$, i.e., $M, s_0 \not\models \phi'$.

(ii.b) On the other hand, suppose that $\phi'' = E_{\{a,b\}}p$, then we have $M, s_0 \models \phi''$.

For $k = 1$, we have $M, s_0 \models_1 \phi''$, since we only have two k-CK paths (s_0, s_0) and (s_0, s_1) starting from s_0 for the set $\{a, b\}$ of agents, and both of $M, s_0 \models_1 p$ and $M, s_1 \models_1 p$ hold. Then by the theorem, this is sufficient for proving $M, s_0 \models \phi''$.

Remark Notice that although we have $M, s \not\models \varphi$ iff $M, s \models \neg\varphi$, we do not have $M, s \not\models_k \varphi$ iff $M, s \models_k \neg\varphi$. According to the bounded semantics, both of $M, s \models_k \varphi$ and $M, s \models_k \neg\varphi$ may be false. On the other hand, at most one of them can be true.

4 Checking Temporal-Epistemic Properties

By Theorem 3.1 (and Corollary 3.1), we are able to checking temporal-epistemic properties with a bounded correctness checking approach. Let M be a Kripke structure and φ a CTLK formula in NNF. The algorithmic formulation of the verification approach is presented as a pseudo-algorithm as follows, where $\neg\varphi$ denote the NNF formula equivalent to $\neg\varphi$.

```
1: k := 1;
2: while true do
3:     if $M \models_k \varphi$, report $M \models \varphi$ and return;
4:     if $\exists\iota \in I.(M, \iota \models_k \neg\varphi)$, report $M \not\models \varphi$ and return;
5:     k := k+1;
6: end while
```

The correctness of line 3 of the pseudo-algorithm follows from Corollary 3.1. The correctness of line 4 follows from the following equivalences.

$M \not\models \varphi$
iff $\exists\iota \in I.(M, \iota \models \neg\varphi)$ (the semantics of $\models$)
iff $\exists\iota \in I.(M, \iota \models_k \neg\varphi)$ for some $k \geq 1$ (Theorem 3.1)

The termination of the verification approach follows from the fact that we have either $M \models \varphi$ or $M \not\models \varphi$, and in the former case, there is a $k \geq 1$ such that $M \models_k \varphi$ (by Corollary 3.1) and in the latter case, there is an $\iota \in I$ such that $M, \iota \models_k \neg\varphi$ and then there is a $k \geq 1$ such that $M, s \models_k \neg\varphi$ (by Theorem 3.1).

Remark Notice that in the algorithmic formulation, the input is M and φ, the variable k is an internal variable, and the output is a report on whether $M \models \varphi$ holds.

Transformation of Problems In the algorithmic formulation, we need subroutines for the evaluation of $M, s \models_k \varphi$. It would not be efficient to enumerate all k-paths starting at s and then check the satisfiability according to the bounded semantics.

Similar to the bounded model checking approaches presented in e.g., [Biere *et al.*, 1999; Penczek and Lomuscio, 2003], we have to transform the problem of verifying $M, s \models_k \varphi$ into the checking of the satisfiability of logical formulas. In this case, it is natural to use quantified Boolean formulas instead of propositional formulas, since we have alternations of path quantifiers in the temporal formulas.

By adapting the ideas presented in [Clarke *et al.*, 1999] on symbolic models, we can encode the Kripke structure $(S, T, I, \sim_1, \ldots, \sim_n, L)$ by a symbolic model $(V, \rho_T, \rho_I, \rho_1, \ldots, \rho_n, \rho_L)$ with $V = \{v_1, ..., v_m\}$ being a set of m Boolean variables with $|S| \leq 2^m$ such that we have the following.

- A state $s \in S$ maps to a Boolean formula $f_s(v_1, ..., v_m)$, such that there is a unique assignment of the Boolean variables $v_1, ..., v_m$ that satisfies $f_s(v_1, ..., v_m)$, and the assignment represents the state s;
- A set $X \subseteq S$ maps to a Boolean formula $f_X(v_1, ..., v_m)$;
- A pair (s, s') maps to a Boolean formula
$$g_{(s,s')}(v_1, ..., v_m, v'_1, ..., v'_m)$$
such that there is a unique assignment of the Boolean variables that satisfies $g_{(s,s')}(v_1, ..., v_m, v'_1, ..., v'_m)$, and the part of the assignment of $v_1, ..., v_m$ represents the state s, and the assignment of the primed variables $v'_1, ..., v'_m$ represents the successor state s'.
- A set Y of pairs maps to $g_Y(v_1, ..., v_m, v'_1, ..., v'_m)$.

Accordingly, we have the following.

- The transition relation T is represented by $\rho_T = g_T(v_1, ..., v_m, v'_1, ..., v'_m)$;
- The set I of initial states is represented by $\rho_I = f_I(v_1, ..., v_m)$;
- For each $i \in \{1, ..., n\}$, the equivalence $\sim_i$ is represented by $\rho_i = g_{\sim_i}(v_1, ..., v_m, v'_1, ..., v'_m)$;
- The labeling function $L : S \rightarrow 2^{AP}$ is represented by ρ_L defined by: $\rho_L(p) = f_X(v_1, ..., v_m)$, where $X = \{s \mid p \in L(s)\}$, for each $p \in AP$.

The construction of f_s, f_X, g_s, g_X follows from the standard encoding techniques [Clarke *et al.*, 1999] for construction of symbolic models, and the details are omitted.

In the following, we assume that we have the symbolic model $(V, \rho_T, \rho_I, \rho_1, \ldots, \rho_n, \rho_L)$.

Then we can encode that φ is satisfied on a state represented by $v_1, ..., v_m$ under the bounded semantics $\models_k$ by a QBF-formula $h_{\varphi,k}(v_1, ..., v_m)$ such that the following holds.

$M, s \models_k \varphi$ iff
$\forall v_1 \cdots v_m.(f_s(v_1, ..., v_m) \rightarrow h_{\varphi,k}(v_1, ..., v_m))$ is valid.

The encoding function $h_{\varphi,k}(v_1, ..., v_m)$ is defined as follows.

Suppose that φ is given as the property to be verified.

Let $a \in \varphi$ denote that a appears in φ.

Let $\gamma_D = \{\Gamma \mid S_\Gamma \in \varphi, S \in \{D, \overline{D}\}\}$ and $\gamma_C = \{\Gamma \mid S_\Gamma \in \varphi, S \in \{K, \overline{K}, E, \overline{E}, C, \overline{C}\}\}$. Then we have the following definitions.

- $\rho_{D_\Gamma} = \bigwedge_{i \in \Gamma} \rho_i$ for $\Gamma \in \gamma_D$.
- $\rho_{C_\Gamma} = \bigvee_{i \in \Gamma} \rho_i$ for $\Gamma \in \gamma_C$.

ρ_{D_Γ} represents the conjunction of the equivalence relations in the group of agents Γ. ρ_{C_Γ} represents the disjunction of the equivalence relations in Γ.

Let $k \geq 1$. Let $\overrightarrow{u} = u_0, \ldots, u_k$ be a finite sequence of state variables, where $u_i = (u_{i,l}, ..., u_{i,m})$ such that an assignment to u_i represents a state of the system.

When q is a formula over $\{v_1, ..., v_m\}$, we use $q(u_i)$ to denote the formula with $\{v_1, ..., v_m\}$ replaced by $(u_{i,1}, ..., u_{i,m})$, and when q is a formula over $\{v_1, ..., v_m, v'_1, ..., v'_m\}$, we use $q(u_i, u_{i+1})$ to denote the formula with $\{v_1, ..., v_m, v'_1, ..., v'_m\}$ replaced by $(u_{i,1}, ..., u_{i,m}, u_{i+1,1}, ..., u_{i+1,m})$.

We use $P_k(\overrightarrow{u})$ to represent a k-path. For a group of agents Γ, we use $P_k^{C_\Gamma}(\overrightarrow{u})$ and $P_k^{D_\Gamma}(\overrightarrow{u})$ to represent a k-CK-path and a k-DK-path respectively.

Formally, we have $P_k(\overrightarrow{u}) = \bigwedge_{j=0}^{k-1} \rho_T(u_j, u_{j+1})$, $P_k^{C_\Gamma}(\overrightarrow{u}) = \bigwedge_{j=0}^{k-1} \rho_{C_\Gamma}(u_j, u_{j+1})$, and $P_k^{D_\Gamma}(\overrightarrow{u}) = \bigwedge_{j=0}^{k-1} \rho_{D_\Gamma}(u_j, u_{j+1})$.

Let $u_x = u_y$ denote $(u_{x,1} \leftrightarrow u_{y,1}) \wedge \cdots \wedge (u_{x,m} \leftrightarrow u_{y,m})$. That a k-path represented by $\overrightarrow{u}$ is an rs-path, denoted $rs_k(\overrightarrow{u})$, is defined by: $rs_k(\overrightarrow{u}) := \bigvee_{x=0}^{k-1} \bigvee_{y=x+1}^{k} u_x = u_y$.

Definition 4.1 (Encoding of CTLK Formulas). *Let $k \geq 1$. Let $w = (\{w_1, ..., w_m\})$ be a state variable and φ be a $CTLK$ formula. The encoding $[[\varphi, w]]_k$ is defined as in Table 3.*

Let $h_{\varphi,k}(w_1, ..., w_m) = [[\varphi, w]]_k$. Following from the encoding, we have the following theorem and the corollary.

Theorem 4.1. *Let $k \geq 1$. $M, s \models_k \varphi$ iff $\forall v_1 \cdots v_m.(f_s(v_1, ..., v_m) \rightarrow h_{\varphi,k}(v_1, ..., v_m))$ is valid.*

Corollary 4.1. *The following hold.*

- *$M \models_k \varphi$ iff*
$\forall v_1 \cdots v_m.(\rho_I(v_1, ..., v_m) \rightarrow h_{\varphi,k}(v_1, ..., v_m)))$;
- *$\exists \iota \in I.(M, \iota \models_k \neg\varphi)$ iff*
$\exists v_1 \cdots v_m.(\rho_I(v_1, ..., v_m) \wedge h_{\neg\varphi,k}(v_1, ..., v_m)))$.

Then accordingly, checking whether a temporal-epistemic formula is satisfied on a Kripke structure produced from an interpreted system can be turned into checking the validity of QBF-formulas (via the bounded semantics and the encoding), by the following verification procedure (which is a modification of the one presented at the beginning of this section).

Table 3: Encoding Scheme

$[[p, w]]_k$	$= \rho_L(p)(w)$
$[[\neg p, w]]_k$	$= \neg\rho_L(p)(w)$
$[[\varphi \wedge \psi, w]]_k$	$= [[\varphi, w]]_k \wedge [[\psi, w]]_k$
$[[\varphi \vee \psi, w]]_k$	$= [[\varphi, w]]_k \vee [[\psi, w]]_k$
$[[AX\varphi, w]]_k$	$= \forall\overrightarrow{u}.(P_k(\overrightarrow{u}) \wedge w = u_0 \rightarrow [[\varphi, u_1]]_k)$
$[[A(\varphi U\psi), w]]_k$	$= \forall\overrightarrow{u}.(P_k(\overrightarrow{u}) \wedge w = u_0 \rightarrow$ $(\bigvee_{j=0}^{k}([[\psi, u_j]]_k \wedge \bigwedge_{t=0}^{j-1}[[\varphi, u_t]]_k))$
$[[A(\varphi R\psi), w]]_k$	$= \forall\overrightarrow{u}.(P_k(\overrightarrow{u}) \wedge w = u_0 \rightarrow$ $((rs_k(\overrightarrow{u}) \wedge \bigwedge_{j=0}^{k}[[\psi, u_j]]_k \vee$ $(\bigvee_{j=0}^{k}([[\varphi, u_j]]_k \wedge \bigwedge_{t=0}^{j-1}[[\psi, u_t]]_k))))$
$[[EX\varphi, w]]_k$	$= \exists\overrightarrow{u}.(P_k(\overrightarrow{u}) \wedge w = u_0 \wedge [[\varphi, u_1]]_k)$
$[[E(\varphi U\psi), w]]_k$	$= \exists\overrightarrow{u}.(P_k(\overrightarrow{u}) \wedge w = u_0 \wedge$ $(\bigvee_{j=0}^{k}([[\psi, u_j]]_k \wedge \bigwedge_{t=0}^{j-1}[[\varphi, u_t]]_k))$
$[[E(\varphi R\psi), w]]_k$	$= \exists\overrightarrow{u}.(P_k(\overrightarrow{u}) \wedge w = u_0 \wedge$ $((rs_k(\overrightarrow{u}) \wedge \bigwedge_{j=0}^{k}[[\psi, u_j]]_k \vee$ $(\bigvee_{j=0}^{k}([[\varphi, u_j]]_k \wedge \bigwedge_{t=0}^{j-1}[[\psi, u_t]]_k))))$
$[[K_i\varphi, w]]_k$	$= \forall\overrightarrow{u}.(P_k^{C_{\{i\}}}(\overrightarrow{u}) \wedge w = u_0 \rightarrow [[\varphi, u_1]]_k)$
$[[D_\Gamma\varphi, w]]_k$	$= \forall\overrightarrow{u}.(P_k^{D_\Gamma}(\overrightarrow{u}) \wedge w = u_0 \rightarrow [[\varphi, u_1]]_k)$
$[[E_\Gamma\varphi, w]]_k$	$= \forall\overrightarrow{u}.(P_k^{C_\Gamma}(\overrightarrow{u}) \wedge w = u_0 \rightarrow [[\varphi, u_1]]_k)$
$[[C_\Gamma\varphi, w]]_k$	$= \forall\overrightarrow{u}.(P_k^{C_\Gamma}(\overrightarrow{u}) \wedge w = u_0 \rightarrow$ $(rs_k(\overrightarrow{u}) \wedge \bigwedge_{j=0}^{k}[[\varphi, u_j]]_k))$
$[[\overline{K}_i\varphi, w]]_k$	$= \exists\overrightarrow{u}.(P_k^{C_{\{i\}}}(\overrightarrow{u}) \wedge w = u_0 \wedge [[\varphi, u_1]]_k)$
$[[\overline{D}_\Gamma\varphi, w]]_k$	$= \exists\overrightarrow{u}.(P_k^{D_\Gamma}(\overrightarrow{u}) \wedge w = u_0 \wedge [[\varphi, u_1]]_k)$
$[[\overline{E}_\Gamma\varphi, w]]_k$	$= \exists\overrightarrow{u}.(P_k^{C_\Gamma}(\overrightarrow{u}) \wedge w = u_0 \wedge [[\varphi, u_1]]_k)$
$[[\overline{C}_\Gamma\varphi, w]]_k$	$= \exists\overrightarrow{u}.(P_k^{C_\Gamma}(\overrightarrow{u}) \wedge w = u_0 \wedge \bigvee_{j=0}^{k}[[\varphi, u_j]]_k))$

```
k := 1;
while true do
    if $\forall v_1 \cdots v_m.(\rho_I(v_1, ..., v_m) \rightarrow h_{\varphi,k}(v_1, ..., v_m)))$,
    report $M \models \varphi$ and return;
    if $\exists v_1 \cdots v_m.(\rho_I(v_1, ..., v_m) \wedge h_{\neg\varphi,k}(v_1, ..., v_m)))$,
    report $M \not\models \varphi$ and return;
    k := k+1;
end while
```

Corollary 4.1 together with the correctness of the verification approach presented at the beginning of this section guarantees that this verification procedure always terminates and returns with the correct answer.

5 Examples and Experimental Comparison

In this section, we first present two examples for demonstrating the use of the bounded semantics as a verification method, and then present an experimental comparison of an implementation of this approach and an implementation of BDD based symbolic model checking approach.

5.1 An Example with the Train Controller System

In this subsection, we consider the train controller system [Hoek and Wooldridge, 2002; Kacprzak *et al.*, 2004a]. The system consists of two trains and a controller. It assumed that the two trains from opposite directions have to pass through a tunnel which has only one track inside. In this case, two trains cannot pass through the tunnel at the same time, and the purpose of the controller is to make sure that there is at most one train inside the tunnel at a time. There are traffic lights on both sides of the tunnel which can be either red or green. When the trains approach or leave the tunnel, they will notify the controller, and the controller can control the color of the traffic lights accordingly.

Let t_1, t_2 and c_0 be three agents representing respectively the two trains and the controller. The system can be expressed with an interpreted system as follows.

- The local states of each of the agents:
  - $L_{t_1} = \{aw, wt, tunl\}$
  - $L_{c_0} = \{r, g\}$
  - $L_{t_2} = \{aw, wt, tunl\}$
- The set of global states, $S = L_{t_1} \times L_{c_0} \times L_{t_2}$.
- The set of the initial states of the system, $I = \{(aw, g, aw)\}$.
- The actions of each of the agents:
  - $Act_{t_1} = \{none, aprch, request, leave\}$
  - $Act_{c_0} = \{none, tr, tg\}$
  - $Act_{t_2} = \{none, aprch, request, leave\}$
- The protocols for each of the agents:
  - $p_{t_1}(aw) = \{null, aprch\}$,
    $p_{t_1}(wt) = \{null, request\}$,
    $p_{t_1}(tunl) = \{null, leave\}$
  - $p_{t_2}(aw) = \{null, aprch\}$,
    $p_{t_2}(wt) = \{null, request\}$,
    $p_{t_2}(tunl) = \{null, leave\}$
  - $p_{c_0}(red) = \{null, tg\}$,
    $p_{c_0}(green) = \{null, tr\}$

Then we have the transition relation of the reachable states of the system shown in Figure 2.

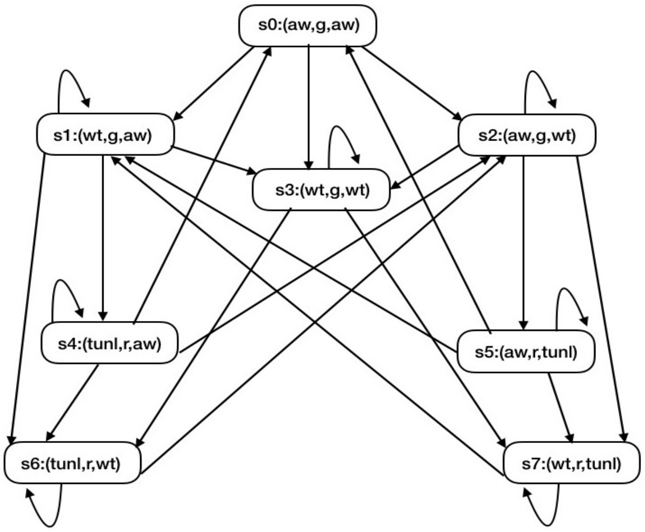

Figure 2: the train controller system

For the agent c_0, we have $\sim_{c_0}$ as follows: $s \sim_{c_0} s'$ iff $s, s' \in \{s_0, s_1, s_2, s_3\}$ or $s, s' \in \{s_4, s_5, s_6, s_7\}$, when s, s' are restricted to those of the reachable states.

For the agent t_1, we have $\sim_{t_1}$ as follows: $s \sim_{t_1} s'$ iff $s, s' \in \{s_0, s_2, s_5\}$ or $s, s' \in \{s_1, s_3, s_7\}$ or $s, s' \in \{s_4, s_6\}$ when s, s' are restricted to those of the reachable states.

For the agent t_2, we have $\sim_c$ as follows: $s \sim_{t_2} s'$ iff $s, s' \in \{s_0, s_1, s_4\}$ or $s, s' \in \{s_2, s_3, s_6\}$ or $s, s' \in \{s_5, s_7\}$, when s, s' are restricted to those of the reachable states.

Let $AP = \{t1tunnel, t2tunnel\}$ be a set of proposition symbols representing the basic properties of the states. Let $L : S \to 2^{AP}$ be defined as follows.

- $t1tunnel \in L(s)$ iff $l_{t_1}(s) = tunl$
- $t2tunnel \in L(s)$ iff $l_{t_2}(s) = tunl$

Then the system satisfies the the following properties.

- $(a)\ \psi_1 = AG(t1tunnel \to K_{t_1}(\neg t2tunnel))$
- $(b)\ \psi_2 = EF(K_{t_1}(t1tunnel \wedge \overline{K}_{t_2}(\neg t1tunnel)))$

In the following, we demonstrate how ψ_1 and ψ_2 are verified by using the bounded semantics.

(a) Verification of ψ_1.

Let $\varphi = t1tunnel \to K_{t_1}(\neg t2tunnel)$.

We start with $k = 1$, and for $k = 1, 2, .., 6$, we neither have $M \models_k \psi_1$, nor $\exists \iota \in I.(M, \iota \models_k \neg\psi_1)$, and we have to consider the case where $k = 7$.

For $k = 7$, we have to check all the rs-paths starting from the initial state s_0. It is easily seen that all the rs-paths cover all eight reachable states from s_0 to s_7. Then we verify whether φ holds at all these states.

It is easily seen that φ is true on the states on which $t1tunnel$ does not hold. The states on which $t1tunnel$ hold are s_4 and s_6, and for these two states, it is easily seen that $\neg t2tunnel$ holds on all the next states on the knowledge-paths starting from s_4 and s_6. Hence $M \models_k \psi_1$, and therefore $M \models \psi_1$.

(b) Verification of ψ_2.

Let $\varphi = K_{t_1}(t1tunnel \wedge \overline{K}_{t_2}(\neg t1tunnel))$.

We start with $k = 1$, and consider the k-paths starting from the initial state s_0. The k-paths are s_0s_1, s_0s_2 and s_0s_3.

Then we need $(s_0 \models \varphi) \vee (s_1 \models \varphi)$ or $(s_0 \models \varphi) \vee (s_2 \models \varphi)$ or $(s_0 \models \varphi) \vee (s_3 \models \varphi)$.

It's easy to see that $K_{t_1}(t1tunnel)$ doesn't hold on any of the four states s_0, s_1, s_2 or s_3. Thus, $k = 1$ is not enough to prove $EF\varphi$.

Then we take $k = 2$ and now the k-paths are $s_0s_1s_1$, $s_0s_1s_3$, $s_0s_1s_4$, $s_0s_1s_6$, $s_0s_2s_2$, $s_0s_2s_3$, $s_0s_2s_5$, $s_0s_2s_7$, $s_0s_3s_3$, $s_0s_3s_6$ and $s_0s_3s_7$.

Then we find that $s_0s_1s_4$ is a path where φ holds on s_4. For a proof of this, we have to look at all the k-CK-paths for the agent t_1 starting from s_4, which are $s_4s_4s_4$, $s_4s_4s_6$, $s_4s_6s_4$and $s_4s_6s_6$, and check whether $s_4 \models t1tunnel \wedge \overline{K}_{t_2}(\neg t1tunnel)$ and $s_6 \models t1tunnel \wedge \overline{K}_{t_2}(\neg t1tunnel)$ hold.

It is easily verified that $t1tunnel$ holds on s_4 and s_6.

For proving that $\overline{K}_{t_2}(\neg t1tunnel)$ also holds on s_4 and s_6, we have to consider all the k-CK-paths for the agent t_2 starting from s_4 and s_6, and check $\neg t1tunnel$ holds on at least one state on at least one of these paths. $s_4s_1s_0$ and $s_6s_3s_2$ can be two witnesses. Then this can be successfully verified, and therefore the property is true.

5.2 The Dining Cryptographers

In this subsection, we consider the dining cryptographers [Raimondi and Lomuscio, 2007; Chaum, 1988]. There are three cryptographers having a dinner in a round table and the bill is supposed to be paid anonymously by one of the cryptographers or NSA (the agency). A protocol is executed to figure out whether NSA is paying or one of the three cryptographers is paying. Each cryptographer flips a coin between him and the cryptographer on his right hand side such that only these two cryptographers can see the outcome, i.e., each cryptographer can see the coins on his left hand side and that on his right hand side. Then the cryptographers report whether the two coins he sees fell on the same side or not. The rule is that a cryptographer only reports the truth if he is not the payer (otherwise he reports the opposite of the fact he sees). An odd number of differences of the reported facts indicates that a cryptographer is paying, an even number indicates that the payer is NSA. For the formal description of the interpreted system, we take the three cryptographers as three agents c_1, c_2 and c_3.

- The local state of each agent $c_i \in \{c_1, c_2, c_3\}$ is $L_{c_i} = (Payer, SeeDifferent, NumberOfOdd)$ where $Payer \in \{y, n\}$ indicates whether or not the agent is the payer, $SeeDifferent \in \{none, y, n\}$ indicates whether the two coins seen by the agent are different and $NumberOfOdd \in \{none, odd, even\}$ indicates the number of differences of the reported facts is even or odd, where $none$ is the initial value.
- The actions of each agent $c_i \in \{c_1, c_2, c_3\}$ is as follows.
  - $Act_{c_i} = \{checkCoins, sayDiff, saySame\}$
- The protocol of each agent $c_i \in \{c_1, c_2, c_3\}$ is as follows.
  - $P_{c_i}(n, none, none) = P_{c_i}(y, none, none) = \{checkCoins\}$
  - $P_{c_i}(n, y, none) = P_{c_i}(y, n, none) = \{sayDiff\}$
  - $P_{c_i}(n, n, none) = P_{c_i}(y, y, none) = \{saySame\}$
- Then the set of global states is $S = L_{c_1} \times L_{c_2} \times L_{c_3}$
- There are four initial states : either one of the cryptographers is paying or NSA is paying (i.e., none of the cryptographers is paying), and both $SeeDifferent = none$ and $NumberOfOdd = none$ at the beginning for all the three agents.

The reachable states of the system is shown in Figure 3, where the second and the third trees are similar to the first one.

The Kripke structure consists of four trees with four initial states as their roots. The first three trees represent the cases in which c_i (with $i = 1, 2, 3$) is the payer, while the last tree represents the case in which NSA is the payer. Each state has 9 attributes, and the initial state s_1 represents $(y, none, none, n, none, none, n, none, none)$, where the values of the first three attributes represent the local state of c_1, i.e., $L_{c_1}(s_1) = (y, none, none)$, and at state s_1, it can only perform the action $checkCoins$ and then leads to the four successor-states which represent additionally

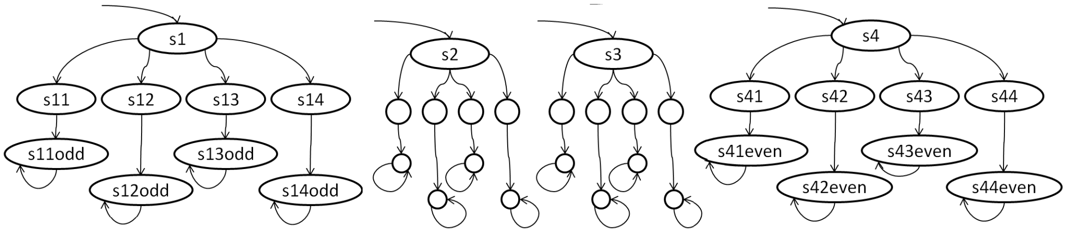

Figure 3: The dining cryptographers

the possibility of the outcome of the coins seen by the agents (i.e., leading to a change of the value of the attribute $SeeDifferent$). Since three coins can only be three in the same side or two in the same side, the outcome would be two of the agents see different sides or none of them see different sides which are the four cases. Then the system takes actions $sayDiff$ or $saySame$ according to the protocols from these four cases to the leaves of the trees. The other trees follow the same approach. Only the leaves of the last tree have $NumberOfOdd = even$ while the leaves of other trees have $NumberofOdd = odd$. Note that we add a self loop to every leaf because the transition relation in a Kripke model is required to be total.

Let nd denote the number of differences of the reported facts from the dinning cryptographers. The equivalence relation of $\sim_{c_1}$ is represented by a partition of the states as follows (it is similar for the other two agents, which are omitted for brevity).

- $\{s_1\}$ where c_1 is the payer
- $\{s_2, s_3, s_4\}$ where c_1 is not the payer
- $\{s_{11}, s_{12}\}$ where c_1 is the payer and he sees different side of coins
- $\{s_{13}, s_{14}\}$ where c_1 is the payer and he sees same side of coins
- $\{s_{21}, s_{22}, s_{31}, s_{32}, s_{41}, s_{42}\}$ where c_1 is not the payer and he sees different side of coins
- $\{s_{23}, s_{24}, s_{33}, s_{34}, s_{43}, s_{44}\}$ where c_1 is not the payer and he sees same side of coins
- $\{s_{11odd}, s_{12odd}\}$ where nd is odd, with other properties inherited from $\{s_{11}, s_{12}\}$
- $\{s_{13odd}, s_{14odd}\}$ where nd is odd, with other properties inherited from $\{s_{13}, s_{14}\}$
- $\{s_{21odd}, s_{22odd}, s_{31odd}, s_{32odd}\}$ where nd is odd, other properties inherited from $\{s_{21}, s_{22}, s_{31}, s_{32}\}$
- $\{s_{23odd}, s_{24odd}, s_{33odd}, s_{34odd}\}$ where nd is odd, other properties inherited from $\{s_{23}, s_{24}, s_{33}, s_{34}\}$
- $\{s_{41even}, s_{42even}\}$ where nd is even, other properties inherited from $\{s_{41}, s_{42}\}$
- $\{s_{43even}, s_{44even}\}$ where nd is even, other properties inherited from $\{s_{43}, s_{44}\}$

Let $AP = \{pay_{nsa}, pay_{c_1}, pay_{c_2}\}$ be a set of proposition symbols representing the basic properties of the states. Let $L : S \to 2^{AP}$ be defined as follows (where the place holder _ represents any value of the attribute).

- $pay_{nsa} \in L(s)$ iff $L_{c_1}(s)$, $L_{c_2}(s)$ and $L_{c_3}(s)$ are all of the form $(n, \_, \_)$,
- for each $i \in \{1, 2, 3\}$, $pay_{c_i} \in L(s)$ iff $L_{c_i}(s)$ is of the form $(y, \_, \_)$.

Then we consider the following properties.

- $(a)\ \psi_1 = AF(K_{c_1}(pay_{nsa}) \vee K_{c_1}(\neg pay_{nsa}))$
- $(b)\ \psi_2 = AG\neg K_{c_1}(pay_2)$
- $(c)\ \psi_3 = AG\neg K_{c_1}(pay_{nsa})$
- $(d)\ \psi_4 = AG\neg C_{\{c_1,c_2,c_3\}}(pay_{nsa})$

The first property states the fact that this protocol can help c_1 finding out whether NSA is paying. The second property states that for all reachable states, agent c_1 does not know that agent c_2 is paying, or equivalently, there does not exists any reachable state where agent c_1 knows that agent c_2 is paying. The third property states that there does not exists any reachable state where agent c_1 knows that NSA is paying. This property does not hold, since it is not necessary to keep anonymity when NSA is paying the bill. The last property does not hold either, and the negation of the last property states further that there is a state where the agents will have a common knowledge on that NSA is paying. We demonstrate how these properties are verified or falsified by using the bounded semantics.

$(a)\ \psi_1 = AF(K_{c_1}(pay_{nsa}) \vee K_{c_1}(\neg pay_{nsa}))$

Proof. Let $\varphi = K_{c_1}(pay_{nsa})$ and $\phi = K_{c_1}(\neg pay_{nsa})$.

We start with $k = 1$. We have to consider the 16 k-paths (cf. Figure 3) starting from the initial states, and prove that there exists a state on every k-paths such that $\varphi \vee \phi$ holds.

Then for the k-path s_2s_{21}, we find that $s_2 \not\models \varphi \vee \phi$ and $s_{21} \not\models \varphi \vee \phi$. This is because proving $s_2 \models \varphi \vee \phi$ and $s_{21} \models \varphi \vee \phi$ requires considering all k-CK-paths starting from s_2 and s_{21}. For the CK-path s_2s_4, $s_2 \not\models \varphi \vee \phi$ because $s_2 \models \neg pay_{nsa}$ while $s_4 \models pay_{nsa}$. For the CK-path $s_{21}s_{41}$, we have $s_{21} \not\models \varphi \vee \phi$ because $s_{21} \models \neg pay_{nsa}$ while $s_{41} \models pay_{nsa}$. Hence, $k = 1$ is not sufficient for proving the property.

Then we increase k to 2, and once again we need to consider the 16 k-paths starting from the initial states and to prove on each k-path there is at least one state that $\varphi \vee \phi$ is satisfiable. For every such 2-paths, the last state is the leaf of the trees and $\varphi \vee \phi$ is satisfied on all these leaves. This is because the property is satisfied on all states on all the k-CK-paths starting from the leaves (cf. the equivalence relation of c_1 presented above). Those leaves marked with odd have $\neg pay_{nsa}$ satisfied and those that marked with $even$ have pay_{nsa} satisfied. Therefore this property holds.

(b) $\psi_2 = AG\neg K_{c_1}(pay_2)$

Proof. We have to prove that $AG\overline{K}_{c_1}(\neg pay_2)$ is true. This can be done with $k = 3$. For the first, all the k-paths starting from the initial states with $k = 3$ are rs-path. Then we consider all the states on those k-paths, and then the k-CK-paths starting from these states and try to find one state on the CK-paths with $\neg pay_2$ satisfied. Since every equivalence set (cf. the equivalence relation of c_1 presented above) has at least one state satisfying $\neg pay_2$, the property $AG\overline{K}_{c_1}(\neg pay_2)$ is true.

(c) $\psi_3 = AG\neg K_{c_1}(pay_{nsa})$

Proof. This property is false. For this, we have to prove that $EFK_{c_1}(pay_{nsa})$ holds on some initial state. This can be verified with $k = 2$. The k-path $s_4 s_{41} s_{41even}$ is a sufficient one for witnessing $EFK_{c_1}(pay_{nsa})$, since $s_{41even} \models K_{c_1}(pay_{nsa})$, which is explained as follows. The k-CK-paths starting from s_{41even} are $s_{41even} s_{41even} s_{41even}$, $s_{41even} s_{41even} s_{42even}$, $s_{41even} s_{42even} s_{42even}$ and $s_{41even} s_{42even} s_{41even}$. Since $s_{41even} \models pay_{nsa}$ and $s_{42even} \models pay_{nsa}$, we have the property. Therefore the property ψ_3 does not hold.

(d) $\psi_4 = AG\neg C_{\{c_1,c_2,c_3\}}(pay_{nsa})$

Proof. This property is false. For this, we have to prove that $EFC_{\{c_1,c_2,c_3\}}(pay_{nsa})$ holds on some initial state. This can be verified with $k = 4$. The k-path $s_4 s_{41} s_{41even} s_{41even} s_{41even}$ would be a sufficient one for witnessing $EFC(pay_{nsa})$, if we had that $C(pay_{nsa})$ holds on s_{41even}. To prove $s_{41even} \models C(pay_{nsa})$, We need to consider all CK-paths (where $\Gamma = \{c_1, c_2, c_3\}$) starting from s_{41even}. For the agent c_1, $\{s_{41even}, s_{42even}\}$ and $\{s_{43even}, s_{44even}\}$ are the two relevant equivalence sets, and for the other two agents, the relevant equivalence sets are $\{s_{41even}, s_{43even}\}$, $\{s_{42even}, s_{44even}\}$, $\{s_{41even}, s_{44even}\}$ and $\{s_{42even}, s_{43even}\}$. All these relations form the edges of the CK-paths. There are no edges going out from the set $\{s_{41even}, s_{42even}, s_{43even}, s_{44even}\}$. Since $k = 4$, all the k-CK-paths starting from s_{41even} are rs-path, and since we have $s_{41even} \models pay_{nsa}$, $s_{42even} \models pay_{nsa}$, $s_{43even} \models pay_{nsa}$ and $s_{44even} \models pay_{nsa}$, all the states on such CK-paths satisfy pay_{nsa}. Therefore the property ψ_4 does not hold.

5.3 Experimental Comparison

The procedure has been implemented based on the model checking tool VERDS [Zhang, 2013; Zhang, 2014], and the implementation is denoted qMAS[1].

The input language of qMAS is kind of a high level description which is consistent with interpreted systems. It includes components such as agents, environment, initial states, properties to verify, et cetera. Both agents and environment have components like local variables, actions, and protocols. qMAS computes reachable states and constructs a Kripke structure from the input language to prepare for the verification procedure. Then the model and the CTLK properties are encoded into QBF-formulas. Following the algorithm, qMAS uses a QBF solver to determine the validity of the QBF-formulas, thus report an answer for whether the properties are satisfied on the model.

The purpose of this subsection is to show the complementary nature of such an approach and BDD based symbolic model checking based on experimental analysis of test cases. We provide experimental data for comparison of qMAS with McMAS[2]on a set of test cases.

Complementary nature means that the two approaches have advantages over each other on different sets of test cases. Since we know that McMAS performs well on many problem instances and qMAS may perform well when the transition relation of the systems is sufficiently complicated and the verification procedure terminates (and returns an answer) when k is relatively small, we only provide evidence on the existence of problem instances that can be handled by qMAS in a more efficient way.

Models We use the transition relations in the train controller system [Hoek and Wooldridge, 2002; Kacprzak *et al.*, 2004a], and that in the dining cryptographers [Raimondi and Lomuscio, 2007; Chaum, 1988] as the basic transition relations, and in order to increase the complexity, we modify the transition relations and add a set of n Boolean variables ($n = 400, 500$ are used for obtaining the following experimental data) initialized to $false$ and assign a random value in each step of the transitions. These two types of models are referred to as type $(1, n)$ model (based on train controller) and type $(2, n)$ model (based on dining cryptographers), respectively.

Property Specifications For models of type $(1, n)$, we check the following properties.

$$\begin{array}{rcl}
\varphi_1 & = & AG(\overline{K}_{t_1}(t2tunnel)) \\
\varphi_2 & = & AG(\overline{K}_{t_1}(t1tunnel \wedge \overline{K}_{t_2}(\neg t1tunnel))) \\
\varphi_3 & = & AF(\overline{K}_{t_1}(t1tunnel)) \\
\varphi_4 & = & AF(\overline{K}_{t_1}(t2tunnel))
\end{array}$$

For models of type $(2, n)$, we check the following properties.

$$\begin{array}{rcl}
\psi_1 & = & AGEX(C_{\{c1,c2\}}(pay_{nsa})) \\
\psi_2 & = & AGEX(K_{c1}(\neg pay_{nsa})) \\
\psi_3 & = & AFEX(\overline{C}_{\{c1,c2\}}(pay_{nsa})) \\
\psi_4 & = & AFEX(K_{c1}(\neg pay_{nsa}) \vee K_{c1}(pay_{nsa}))
\end{array}$$

Experimental Data Since we use random assignments of values to a set of Boolean variables, for each type of the m odels, we use 20 instances (with different values for the Boolean variables) of each type and calculate an average of the verification times. The average times (in seconds) for respectively McMAS and qMAS are presented as follows. The experimental data are obtained by running the t-

[1] http://lcs.ios.ac.cn/∼zwh/qmas/

[2] https://vas.doc.ic.ac.uk/software/mcmas/

wo tools on a 64bit Linux platform on an Intel Xeon CPU E7450@2.40GHz.

Type	Prop.	T/F	McMAS	qMAS
(1,400)	φ_1	false	46.4	27.8
(1,400)	φ_2	false	46.7	18.7
(1,400)	φ_3	false	47.5	21.3
(1,400)	φ_4	true	47.1	14.3
(1,500)	φ_1	false	66.1	49.1
(1,500)	φ_2	false	65.1	31.6
(1,500)	φ_3	false	65.8	37.3
(1,500)	φ_4	true	66.0	25.5
(2,400)	ψ_1	false	135.9	88.0
(2,400)	ψ_2	false	133.1	87.9
(2,400)	ψ_3	true	138.4	59.1
(2,400)	ψ_4	true	138.5	95.3
(2,500)	ψ_1	false	247.2	146.6
(2,500)	ψ_2	false	248.5	138.2
(2,500)	ψ_3	true	228.1	92.5
(2,500)	ψ_4	true	248.2	147.7

Clearly, the experimental data show that qMAS has advantage over McMAS on the test cases, and the package containing the tool qMAS and the two sets of test cases are available[3].

Notice that the data presented in this table do not provide a comprehensive comparison of McMAS and qMAS, rather, they show the existence of problem instances that can be handled by qMAS (with the bounded semantics approach) in a more efficient way.

6 Concluding Remarks

This work has provided a definition of the bounded semantics of CTLK. This semantics can be viewed as an extension of that of ECTLK [Penczek and Lomuscio, 2003]. The existential fragment handles the epistemic operators $\overline{K}_i\varphi, \overline{E}_\Gamma\varphi, \overline{D}_\Gamma\varphi, \overline{C}_\Gamma\varphi$ which are the dual operators of the commonly-used operators $K_i\varphi$, $E_\Gamma\varphi$, $D_\Gamma\varphi$, $C_\Gamma\varphi$, and an important part of CTLK was left out (e.g., the properties considered in the test cases cannot be handled by the bounded semantics of ECTLK). It is therefore of interest to have a bounded semantics that handle these commonly-used operators with the possibility for the alternation of these and the dual operators (in NNF formulas) as well. The use of knowledge-paths (viewing the Kripke structure as two types of graphs) has simplified the semantics of the operators (although the two definitions of the semantics are equivalent) and had made it more manageable for the definition and reasoning of the bounded semantics, thus we have bounded semantics for the full CTLK. Following from this definition of the bounded semantics, we have provided an approach for checking multi-agent systems against CTLK properties such that the correctness and termination are guaranteed, and an automated verification approach is presented. The approach has been implemented and we have experimental data showing the existence of verification problems that can be verified by this new approach more efficiently than BDD based symbolic model checking.

[3] http://lcs.ios.ac.cn/∼zwh/software/qmas1.0.tar.gz

References

[Belardinelli *et al.*, 2018] Francesco Belardinelli, Alessio Lomuscio, Aniello Murano, and Sasha Rubin. Alternating-time temporal logic on finite traces. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden.*, pages 77–83, 2018.

[Biere *et al.*, 1999] Armin Biere, Alessandro Cimatti, Edmund M. Clarke, and Yunshan Zhu. Symbolic model checking without bdds. In *International Conference on TOOLS and Algorithms for Construction and Analysis of Systems*, pages 193–207, 1999.

[Chaum, 1988] David Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, 1(1):65–75, 1988.

[Clarke and Emerson, 1981] Edmund M. Clarke and E. Allen Emerson. Design and synthesis of synchronization skeletons using branching time temporal logic. In *The Workshop on Logic of Programs*, pages 52–71, 1981.

[Clarke *et al.*, 1999] Edmund M. Clarke, Orna Grumberg, and Doron A. Peled. *Model checking*. MIT Press,, 1999.

[Cohen *et al.*, 2009] Mika Cohen, Mads Dam, Alessio Lomuscio, and Hongyang Qu. A symmetry reduction technique for model checking temporal-epistemic logic. In *International Jont Conference on Artifical Intelligence*, pages 721–726, 2009.

[Dam *et al.*, 2009] Mads Dam, Alessio Lomuscio, and Francesco Russo. Abstraction in model checking multiagent systems. In *International Conference on Autonomous Agents and Multiagent Systems*, pages 945–952, 2009.

[Emerson and Clarke, 1982] E. Allen Emerson and Edmund M. Clarke. Using branching time temporal logic to synthesize synchronization skeletons. *Science of Computer Programming*, 2(3):241–266, 1982.

[Fagin *et al.*, 2004] Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. *Reasoning About Knowledge*. MIT Press,, 2004.

[Gammie and Meyden, 2004] Peter Gammie and Ron Van Der Meyden. Mck: Model checking the logic of knowledge. In *Computer Aided Verification, International Conference, CAV 2004, Boston, Ma, USA, July 13-17, 2004, Proceedings*, pages 479–483, 2004.

[Hoek and Wooldridge, 2002] Wiebe Van Der Hoek and Michael Wooldridge. Tractable multiagent planning for epistemic goals. In *International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1167–1174, 2002.

[Kacprzak *et al.*, 2004a] M Kacprzak, A. Lomuscio, and W. Penczek. Bounded versus unbounded model checking for interpreted systems. *Fundamenta Informaticae*, 2004.

[Kacprzak *et al.*, 2004b] Magdalena Kacprzak, Alessio Lomuscio, and Wojciech Penczek. From bounded to unbounded model checking for temporal epistemic logic. *Fundamenta Informaticae*, 63(2):221–240, 2004.

[Kong and Lomuscio, 2017a] Jeremy Kong and Alessio Lomuscio. Model checking multi-agent systems against LDLK specifications. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 1138–1144, 2017.

[Kong and Lomuscio, 2017b] Jeremy Kong and Alessio Lomuscio. Symbolic model checking multi-agent systems against ctl*k specifications. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2017, São Paulo, Brazil, May 8-12, 2017*, pages 114–122, 2017.

[Kong and Lomuscio, 2018] Jeremy Kong and Alessio Lomuscio. Model checking multi-agent systems against LDLK specifications on finite traces. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2018, Stockholm, Sweden, July 10-15, 2018*, pages 166–174, 2018.

[Kwiatkowska *et al.*, 2010] Marta Kwiatkowska, Alessio Lomuscio, and Hongyang Qu. Parallel model checking for temporal epistemic logic. In *Conference on ECAI 2010: European Conference on Artificial Intelligence*, 2010.

[Lomuscio and Michaliszyn, 2016] Alessio Lomuscio and Jakub Michaliszyn. Verification of multi-agent systems via predicate abstraction against ATLK specifications. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems, Singapore, May 9-13, 2016*, pages 662–670, 2016.

[Lomuscio and Raimondi, 2006] Alessio Lomuscio and Franco Raimondi. The complexity of model checking concurrent programs against ctlk specifications. In *International Workshop on Declarative Agent Languages and Technologies*, pages 548–550, 2006.

[Lomuscio and Ryan, 1997] Alessio Lomuscio and Mark Ryan. On the relation between interpreted systems and kripke models. In *Australian Workshop on Distributed Artificial Intelligence*, pages 46–59, 1997.

[Meski *et al.*, 2014] Artur Meski, Wojciech Penczek, Maciej Szreter, Bozena Wozna-Szczesniak, and Andrzej Zbrzezny. Bdd-versus sat-based bounded model checking for the existential fragment of linear temporal logic with knowledge: algorithms and their performance. *Autonomous Agents and Multi-Agent Systems*, 28(4):558–604, 2014.

[Meyden and Su, 2004] Ron Van Der Meyden and Kaile Su. Symbolic model checking the knowledge of the dining cryptographers. In *Computer Security Foundations Workshop, 2004. Proceedings. IEEE*, pages 280–291, 2004.

[Penczek and Lomuscio, 2003] Wojciech Penczek and Alessio Lomuscio. Verifying epistemic properties of multi-agent systems via bounded model checking. In *The Second International Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2003, July 14-18, 2003, Melbourne, Victoria, Australia, Proceedings*, pages 209–216, 2003.

[Penczek *et al.*, 2012] Wojciech Penczek, Bozena Wozna-Szczesniak, and Andrzej Zbrzezny. Towards sat-based BMC for LTLK over interleaved interpreted systems. *Fundam. Inform.*, 119(3-4):373–392, 2012.

[Raimondi and Lomuscio, 2007] Franco Raimondi and Alessio Lomuscio. Automatic verification of multi-agent systems by model checking via ordered binary decision diagrams. *Journal of Applied Logic*, 5(2):235–251, 2007.

[Shoham and Leyton-Brown, 2008] Yoav Shoham and Kevin Leyton-Brown. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press, 2008.

[Wooldridge, 2009] Michael J Wooldridge. An introduction to multi-agent systems. *Wiley & Sons*, 4(2):125–128, 2009.

[Zhang, 2013] Wenhui Zhang. Verds: Verification of hierarchical discrete systems by symbolic techniques. *Manuscript, available at webpage http://lcs.ios.ac.cn/∼zwh/verds/*, 2013.

[Zhang, 2014] Wenhui Zhang. Qbf encoding of temporal properties and qbf-based verification. In *IJCAR 2014 (LNAI 8562)*, pages 224–239, 2014.

[Zhang, 2015] Wenhui Zhang. Bounded semantics. *Theoretical Computer Sci.*, 564:1–29, 2015.