Bounded Model Checking for the Existential Part of Real-Time CTL and Knowledge

Bożena Woźna-Szcześniak

Bounded Model Checking for the Existential Part of Real-Time CTL and Knowledge

Bożena Woźna-Szcześniak

IMCS, Jan Długosz University. Al. Armii Krajowej 13/15, 42-200 Częstochowa, Poland.
b.wozna@ajd.czest.pl, bwozna@gmail.com

Abstract. A considerably large class of multi-agent systems operate in distributed and real-time environments, and often their correctness specifications require us to express time-critical properties that depend on performed actions of the system. In the paper, we focus on the formal verification of such systems by means of the bounded model checking (BMC) method, where specifications are expressed in the existential fragment of the Real-Time Computation Tree Logic augmented to include standard epistemic operators.

1 Introduction

Model checking [5] is an automatic and usually quite fast verification technique that can be applied to various hardware and software designs where specifications are given by formulae of modal logics. In model checking we represent a program as a labelled transition system (model), and describe a specification as a modal formula in order to check automatically whether the formula holds in the model.

In the last decade, the computer scientists have made tremendous progress in developing new model checking approaches. One of the most successful is called bounded model checking (see, among others, [12,11,14,16,1,2,6]), and it has been introduced as a technique complementary to the BDD-based symbolic model checking method [4]. The BMC method is an efficient SAT-based technique, especially designed for finding bugs in systems and producing counterexamples that can be used to point out the source of errors.

The main idea of BMC relies on looking for witnesses of an existential specification (or equivalently, searching for counterexamples of an universal specification) on suitable subsets of the full model of the system under consideration. Once a submodel is selected, the formula to be checked as well as the considered submodel are encoded as propositional formulae. Next, the propositional satisfiability of the conjunction of the two formulae mentioned above is solved by means of SAT solvers. If the satisfiability test is positive on the submodel, this means that the specification holds on the whole model; this is because an existential syntax is checked. If not, a larger submodel is selected and the whole procedure is run again.

The study of multi-agent systems (MAS) focuses on systems in which many intelligent agents (i.e., autonomous entities, such as software programs or robots) interact with each other. Their interactions can be either cooperative or selfish, that is, the agents can share a common goal or they can pursue their own interests. Also, each agent may

have a deadline or other stated timing constraints to achieve an intended target. Reasoning about knowledge of such agents has always been a core concern in artificial intelligence, and thus many logical formalisms and verification techniques have been proposed and refined over the years, among others, [8,15,9,13].

The aim of this paper is to develop a novel, SAT-based, verification techniques for logic-based specifications of the MAS, in which agents have time-limits or other explicit timing constraints to accomplish intended goals. In particular, we define a bounded model checking method in which a specification is expressed in the existential fragment of the Real-Time CTL augmented to include standard epistemic operators (RTECTLK). RTECTLK is an epistemic real-time computation tree logic that is the fusion [3] of the two underlying languages: an existential fragment of real-time CTL (RTECTL) [7] and $S5_n$ for the knowledge operators [8]. RTECTL is a propositional branching-time temporal logic with bounded operators. It was introduced to permit specification and reasoning about distributed and real-time systems at the twin levels of abstraction: qualitative and quantitative. Obviously, defining the fusion with the full real-time CTL (RTCTL) would not be problematic, but we use here the fragment only because it is more suitable for the BMC method that is defined later on in the paper.

The bounded operators can be translated into nested applications of the EX operator, therefore the expressive power of the RTECTLK is the same as ECTLK [11]. However, this translation is often impractical, and RTECTLK provides a much more compact and convenient way of expressing time-critical (quantitative) properties.

To exemplify the use of the BMC techniques we also present a train controller system – a typical example of the multi-agent system.

The rest of the paper is organised as follows. In the next section the logic RTECTLK is introduced. Then, Section 3 defines the BMC method for RTECTLK. Section 4 shows how the BMC method can be applied to the train controller system. In Section 5 we conclude.

2 The Logic RTECTLK

Syntax. Let $\mathcal{PV}$ be a set of propositional variables, $p \in \mathcal{PV}$, $\mathcal{AG}$ a finite set of agents, $i \in \mathcal{AG}$, $\Gamma \subseteq \mathcal{AG}$, and I an interval in $\mathbb{N} = \{0, 1, 2, \ldots\}$ of the form: $[a, b]$, $[a, b)$, $(a, b]$, (a, b), (a, ∞), and $[a, \infty)$, for $a, b \in \mathbb{N}$. The language RTECTLK is defined by the following grammar:

$$\begin{aligned} \varphi := p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \mathrm{EX}\varphi \mid \mathrm{E}(\varphi \mathrm{U}_I \varphi) \mid \\ \mid \mathrm{E}(\varphi \mathrm{R}_I \varphi) \mid \overline{\mathrm{K}}_i \varphi \mid \overline{\mathrm{D}}_\Gamma \varphi \mid \overline{\mathrm{E}}_\Gamma \varphi \mid \overline{\mathrm{C}}_\Gamma \varphi \,. \end{aligned} \tag{1}$$

The remaining temporal operators are introduced in a standard way: $\mathrm{EF}_I\alpha \stackrel{def}{=} \mathrm{E}(\mathit{true}\mathrm{U}_I\alpha)$, $\mathrm{EG}_I\alpha \stackrel{def}{=} \mathrm{E}(\mathit{false}\mathrm{R}_I\alpha)$. U_I is the operator for bounded "Until", and the formula $\mathrm{E}(\alpha \mathrm{U}_I \beta)$ is read as "there exists a computation in which α holds until, in the interval I, β holds". R_I is the operator for bounded "Release", and $\mathrm{E}(\alpha \mathrm{R}_I \beta)$ is read as "there exists a computation in which either β holds until, in the interval I, both β and α hold, or β always holds in the interval I". G_I is the operator for bounded "Globally",

and the formula $\mathrm{EG}_I\alpha$ is read as "there exists a computation in which α always holds in the interval I". F_I is the operator for bounded "Eventually", and the formula $\mathrm{EF}_I\alpha$ is read as "there exists a computation in which α holds at some point in the interval I". $\overline{\mathrm{K}}_i$ is the operator dual for the standard epistemic modality K_i, so $\overline{\mathrm{K}}_i\alpha$ is read as "agent i considers α as possible". Similarly, the modalities $\overline{\mathrm{D}}_\Gamma, \overline{\mathrm{E}}_\Gamma, \overline{\mathrm{C}}_\Gamma$ are the diamonds for $\mathrm{D}_\Gamma, \mathrm{E}_\Gamma, \mathrm{C}_\Gamma$ representing distributed knowledge in the group Γ, "everyone in Γ knows", and common knowledge among agents in Γ.

Semantics. Traditionally, the semantics of temporal epistemic logics is given on interpreted systems [8]. In this formalism each agent $i \in \mathcal{AG}$ and the environment e are modelled by finite and non-empty sets of *local states* (L_i and L_e), finite and non-empty sets of *actions* (Act_i and Act_e), *protocols* ($P_i\colon L_i \to 2^{Act_i}$ and $P_e\colon L_e \to 2^{Act_e}$) and an *evolution function* ($t_i\colon L_i \times L_e \times Act_1 \times \ldots Act_n \times Act_e \to L_i$). Elements of L_i capture the private information of agent i, and elements of L_e capture the public information of environment e, which means that the other agents can have access to this information. Elements of Act_i represent the possible actions that agent i is allowed to perform; for both the agents and the environment a special *null* action (denoted by ϵ) is allowed, which corresponds to the agent or the environment performing no action. The protocol defines which actions may be performed in each local state of a given agent. Notice that the definition of protocols may enable more than one action to be performed for a given local state. When more than one action is enabled, it is assumed that an agent selects non-deterministically which action to perform. The evolution function defines how local states of a particular agent evolve based on the local states of the agent and the environment, and on actions of other agents. The environment e can be seen as a special agent that models the environment in which the agents operate; we refer to [8] for more details on the above.

Let $W = L_1 \times \cdots \times L_n \times L_e$ and $Act = Act_1 \times \ldots Act_n \times Act_e$. Any element $s \in W$ is called a *global state*, and for a given global state s, $l_i(s)$ denotes the local state of agent i in s. Any element $a \in Act$ is called a *joint action*. A global evolution function $t\colon W \times Act \to W$ is defined as follows: $t(s,a) = s'$ iff for all $i \in \mathcal{AG}$ and e, $t_i(l_i(s),a) = l_i(s')$ and $t_e(l_e(s),a) = l_e(s')$.

Definition 1. *For a given finite set of agents $\mathcal{AG}$ and a set of propositions $\mathcal{PV}$, an* interpreted system *is a tuple $IS = (\langle L_i, Act_i, P_i, t_i\rangle_{i\in\mathcal{AG}}, \langle L_e, Act_e, P_e, t_e\rangle, \iota, \mathcal{V})$, where ι is an* initial global *state, and $\mathcal{V}\colon W \to 2^{\mathcal{PV}}$ is an interpretation for the propositions in $\mathcal{PV}$.*

For a given interpreted system *IS* it is possible to associate a Kripke model M_{IS}; this model we will use to interpret the RTECTLK formulae. The model $M_{IS} = (S, T, \{\sim_i\}_{i\in\mathcal{AG}}, \iota, \mathcal{V})$ is defined as follows:

- $S \subseteq W$ is a set of reachable global states; this is to avoid the epistemic accessibility of states that are not reachable from ι via the global evolution function t.
- $T \subseteq S \times S$ is a serial relation on S that is defined by the global evolution function t as follows: $T(s,s')$ iff there exists an action $a \in Act$ such that $t(s,a) = s'$.
- For each agent $i \in \mathcal{AG}$, $\sim_i \subseteq S \times S$ is an equivalence relation defined as follows: $s \sim_i s'$ iff $l_i(s) = l_i(s')$.

– ι is the *initial global* state of *IS*.
– The valuation function $\mathcal{V}\colon S \to 2^{\mathcal{PV}}$ is the valuation function of *IS* that is restricted to the states from S only; the function assigns to each state a set of proposition variables that are assumed to be true at that state.

A *path* in M_{IS} is an infinite sequence $\pi = (s_0, s_1, \ldots)$ of states such that $(s_i, s_{i+1}) \in T$ for each $i \in \{0, 1, \ldots\}$. For a path $\pi = (s_0, s_1, \ldots)$, we take $\pi(i) = s_i$. By $\Pi(s)$ we denote the set of all the paths starting at $s \in S$. Moreover, for the group epistemic modalities we also define the following. If $\Gamma \subseteq \mathcal{AG}$, then $\sim^E_\Gamma \stackrel{def}{=} \bigcup_{i\in\Gamma} \sim_i$, $\sim^C_\Gamma \stackrel{def}{=} (\sim^E_\Gamma)^+$ (the transitive closure of $\sim^E_\Gamma$), and $\sim^D_\Gamma \stackrel{def}{=} \bigcap_{i\in\Gamma} \sim_i$.

Definition 2. *Let M_{IS} be a model, s a state, and α, β the* RTECTLK *formulae. $M_{IS}, s \models \alpha$ denotes that α is true at the state s in the model M_{IS}. The relation $\models$ is defined inductively as follows:*
$M_{IS}, s \models p$ *iff* $p \in \mathcal{V}(s)$, $M_{IS}, s \models \alpha \vee \beta$ *iff* $M_{IS}, s \models \alpha$ *or* $M_{IS}, s \models \beta$,
$M_{IS}, s \models \neg p$ *iff* $p \notin \mathcal{V}(s)$, $M_{IS}, s \models \alpha \wedge \beta$ *iff* $M_{IS}, s \models \alpha$ *and* $M_{IS}, s \models \beta$,
$M_{IS}, s \models \mathrm{EX}\alpha$ *iff* $\exists\, \pi \in \Pi(s)$ *such that* $M_{IS}, \pi(1) \models \alpha$,
$M_{IS}, s \models \mathrm{E}(\alpha \mathrm{U}_I \beta)$ *iff* $(\exists\, \pi \in \Pi(s))(\exists\, m \in I)$ $[M_{IS}, \pi(m) \models \beta$ *and* $(\forall\, j < m)$ $M_{IS}, \pi(j) \models \alpha]$,
$M_{IS}, s \models \mathrm{E}(\alpha \mathrm{R}_I \beta)$ *iff* $\exists\, \pi \in \Pi(s)$ *such that either* $(\forall\, m \in I)$ $M_{IS}, \pi(m) \models \beta$ *or* $(\exists\, m \in I)$ $[M_{IS}, \pi(m) \models \alpha$ *and* $(\forall\, j \leq m)$ $M_{IS}, \pi(j) \models \beta]$,
$M_{IS}, s \models \overline{\mathrm{K}}_i \alpha$ *iff* $(\exists\, s' \in S)(s \sim_i s'$ *and* $M_{IS}, s' \models \alpha)$,
$M_{IS}, s \models \overline{\mathrm{D}}_\Gamma \alpha$ *iff* $(\exists\, s' \in S)(s \sim^D_\Gamma s'$ *and* $M_{IS}, s' \models \alpha)$,
$M_{IS}, s \models \overline{\mathrm{E}}_\Gamma \alpha$ *iff* $(\exists\, s' \in S)(s \sim^E_\Gamma s'$ *and* $M_{IS}, s' \models \alpha)$,
$M_{IS}, s \models \overline{\mathrm{C}}_\Gamma \alpha$ *iff* $(\exists\, s' \in S)(s \sim^C_\Gamma s'$ *and* $M_{IS}, s' \models \alpha)$.

Definition 3. *A* RTECTLK *formula φ is* valid *in M_{IS} (denoted $M_{IS} \models \varphi$) iff $M_{IS}, \iota \models \varphi$, i.e., φ is true at the initial state of the model M_{IS}.*

3 Bounded Model Checking

Like any other BMC method, also the one defined in this section consists in translating the model checking problem for the RTECTLK into the problem of satisfiability of a propositional formula. In this translation the solution for the epistemic part of the RTECTLK follows the one presented in [11].

We start by defining a notion of k-bounded semantics for the RTECTLK, where $k \in \mathbb{N}_+ = \{1, 2, \ldots\}$. Then, we prove that the model checking problem for a RTECTLK formula can be reduced to the bounded model checking problem for this formula. Finally, we show that the model checking problem for the RTECTLK can be reduced to the satisfiability problem of a propositional formula that is a conjunction of two formulae: the first one encodes a fragment of a model under consideration unfolded up to the depth k, and the second one encodes the checked RTECTLK formula.

3.1 Bounded Semantics of RTECTLK

We begin with some auxiliary definitions. Let M_{IS} be a model associated to an interpreted system *IS*, and $k \in \mathbb{N}_+$ a bound. A *k-path* π in M_{IS} is a finite sequence of states

$(s_0,\ldots,s_k)$ such that $(s_i,s_{i+1})\in T$ for each $0\leq i<k$. A k-path $\pi=(s_0,\ldots,s_k)$ is an *a-loop* for some $a\geq 0$, if there exists $a\leq l\leq k$ such that $T(\pi(k),\pi(l))$. By $\Pi_k(s)$ we denote the set of all the k-paths starting at s in M_{IS}. Note that this set is a convenient way of representing the k-bounded subtree rooted at s of the tree resulting from unwinding the model M_{IS} from s. Moreover, if a k-path is an a-loop, then it represents an *infinite* path.

In the bounded case satisfaction of the temporal operator ER_I with $I=[b,\infty)$ or $I=(b,\infty)$ on a k-path π depends on whether or not π is an a-loop. Therefore, we introduce a function $loop\colon \Pi_k\times\mathbb{N}\to 2^{\mathbb{N}}$ that allows for the identification of the k-paths that are actually a-loops. This function is defined by: $loop(\pi,a)=\{l\mid a\leq l\leq k$ and $T(\pi(k),\ \pi(l))\}$ for some $a\geq 0$. Further, if I is an interval of the form $[a,b]$, $[a,b)$, $(a,b]$, (a,b), $[a,\infty)$ and (a,∞), then by $left(I)$ we denote the left end of the interval I, i.e., $left(I)=a$.

Definition 4. *Let M_{IS} be a model associated to an interpreted system IS, and $k\in\mathbb{N}_+$ a bound. A* k-model *for M_{IS} is a structure $M_k=(S,P_k,\{\sim_i\}_{i\in\mathcal{AG}},\iota,\mathcal{V})$, where $P_k=\bigcup_{s\in S}\Pi_k(s)$.*

Now we can define a notion of bounded satisfaction for the RTECTLK formulae on bounded structures.

Definition 5. *Let $k\in\mathbb{N}_+$, M_{IS} be a model associate to an interpreted system IS, M_k its k-model, and α,β* RTECTLK *formulae. $M_k,s\models\alpha$ denotes that α is true at the state s of M_k. The satisfaction relation $\models$ is defined inductively as follows:*

$M_k,s\models p$ *iff* $p\in\mathcal{V}(s)$, $M_k,s\models\alpha\vee\beta$ *iff* $M_k,s\models\alpha$ *or* $M_k,s\models\beta$,
$M_k,s\models\neg p$ *iff* $p\notin\mathcal{V}(s)$, $M_k,s\models\alpha\wedge\beta$ *iff* $M_k,s\models\alpha$ *and* $M_k,s\models\beta$,
$M_k,s\models\mathrm{EX}\alpha$ *iff* $(\exists\,\pi\in\Pi_k(s))M_k,\pi(1)\models\alpha$,
$M_k,s\models\mathrm{E}(\alpha\mathrm{U}_I\beta)$ *iff* $(\exists\,\pi\in\Pi_k(s))(\exists\,0\leq j\leq k)(j\in I$ *and* $M_k,\pi(j)\models\beta$
and $(\forall\,0\leq i<j)\ M_k,\pi(i)\models\alpha)$,
$M_k,s\models\mathrm{E}(\alpha\mathrm{R}_I\beta)$ *iff* $(\exists\,\pi\in\Pi_k(s))(\exists\,0\leq j\leq k)[(j\in I$ *and* $M_k,\pi(j)\models\alpha$
and $(\forall\,0\leq i\leq j)M_k,\pi(i)\models\beta)$ *or* $(\forall\,j\in I)(M_k,\pi(j)\models\beta$
and $I\cap[0,k]=I)$ *or* $(\forall\,left(I)<j\leq k)(M_k,\pi(j)\models\beta$
and $loop(\pi,left(I)+1)\neq\emptyset)$ *or* $(\forall\,left(I)\leq j\leq k)$
$(M_k,\pi(j)\models\beta$ *and* $loop(\pi,left(I))\neq\emptyset)]$,
$M_k,s\models\overline{\mathrm{K}}_i\alpha$ *iff* $(\exists\,\pi\in\Pi_k(\iota))(\exists\,0\leq j\leq k)(M_k,\pi(j)\models\alpha$ *and* $s\sim_i\pi(j))$,
$M_k,s\models\overline{\mathrm{D}}_\Gamma\alpha$ *iff* $(\exists\,\pi\in\Pi_k(\iota))(\exists\,0\leq j\leq k)(M_k,\pi(j)\models\alpha$ *and* $s\sim_\Gamma^D\pi(j))$,
$M_k,s\models\overline{\mathrm{E}}_\Gamma\alpha$ *iff* $(\exists\,\pi\in\Pi_k(\iota))(\exists\,0\leq j\leq k)(M_k,\pi(j)\models\alpha$ *and* $s\sim_\Gamma^E\pi(j))$,
$M_k,s\models\overline{\mathrm{C}}_\Gamma\alpha$ *iff* $(\exists\,\pi\in\Pi_k(\iota))(\exists\,0\leq j\leq k)(M_k,\pi(j)\models\alpha$ *and* $s\sim_\Gamma^C\pi(j))$.

A RTECTLK formula φ is *valid in k-model M_k* (denoted $M\models_k\varphi$) iff $M_k,\iota\models\varphi$, i.e., φ is true at the initial state of the k-model M_k.

3.2 Equivalence of Bounded and Unbounded Semantics

We start with some auxiliary definitions. By $\mid M_{IS}\mid$ we denote the size of M_{IS}, i.e., the sum of the elements of the set S and the elements of the set T. If I is an interval of the form $[a,b]$, $[a,b)$, $(a,b]$, (a,b), $[a,\infty)$ and (a,∞), for $a,b\in\mathbb{N}$, then by $right(I)$

we denote the right end of the interval I, i.e., $right(I) = b$ if $b \in \mathbb{N}$, and otherwise $right(I) = \infty$. Further, let φ be a RTECTLK formula. Then, we denote by $BI(\varphi)$ the set of all the bounded intervals (i.e., intervals of the form $[a,b]$, $[a,b)$, $(a,b]$, (a,b)) that appear in φ, by $UI(\varphi)$ the set of all the unbounded intervals (i.e., intervals of the form $[a,\infty)$, (a,∞)) that appear in φ, by $Max(BI(\varphi))$ the maximal value of the set $\{b \mid right(I) = b \text{ and } I \in BI(\varphi)\}$, and by $Max(UI(\varphi))$ the maximal value of the set $\{a \mid left(I) = a \text{ and } I \in UI(\varphi)\}$.

By straightforward induction on the length of a RTECTLK formula φ we can show that the following lemma holds.

Lemma 1. *Let $k \in \mathbb{N}_+$, M_{IS} be a model associated to an interpreted system IS, M_k its k-model, and φ a* RTECTLK *formula. Then, for any s in M_{IS}, $M_k, s \models \varphi$ implies $M_{IS}, s \models \varphi$.*

Lemma 2. *Let M_{IS} be a model associated to an interpreted system IS, M_k its k-model, φ a* RTECTLK *formula, s a state of M_{IS}, and $k = \lfloor max\{Max(BI(\varphi)), Max(UI(\varphi))\}/ \mid M_{IS} \mid \rfloor * \mid M_{IS} \mid + \mid M_{IS} \mid$. If $M_{IS}, s \models \varphi$, then $M_k, s \models \varphi$.*

Proof (By induction on the length of φ). The lemma follows directly for the propositional variables and their negations. Next, assume that the hypothesis holds for all the proper sub-formulae of φ. If φ is equal to either $\alpha \wedge \beta$, $\alpha \vee \beta$, or $\mathrm{EX}\alpha$, then it is easy to check that the lemma holds. For the epistemic operators, i.e., $\varphi = \overline{\mathrm{K}}_i\alpha, \overline{\mathrm{E}}_\Gamma\alpha, \overline{\mathrm{D}}_\Gamma\alpha, \overline{\mathrm{C}}_\Gamma\alpha$, the proof is like in [11] (see Lemma 2). So, consider φ to be of the following forms:

1. Let $\varphi = \mathrm{E}(\alpha \mathrm{U}_I \beta)$. From the unbounded semantics we have that there exist a path $\pi \in \Pi(s)$ and $m \in I$ such that $M_{IS}, \pi(m) \models \beta$ and $M_{IS}, \pi(i) \models \alpha$ for all $0 \leq i < m$. Since $m \in I$ and $k = \lfloor max\{Max(BI(\varphi)), Max(UI(\varphi))\}/ \mid M_{IS} \mid \rfloor * \mid M_{IS} \mid + \mid M_{IS} \mid$, then it is easy to see that $m \leq k$. Thus, by the inductive assumption we have that $M_k, \pi(m) \models \beta$, and $M_k, \pi(i) \models \alpha$ for all $0 \leq i < m$. Now, consider the prefix π_k of length k of π. We have that $\pi_k \in \Pi_k(s)$. Since $m \in I$, by the definition of the bounded semantics we can conclude that $M_k, s \models \mathrm{E}(\alpha \mathrm{U}_I \beta)$.
2. Let $\varphi = \mathrm{E}(\alpha \mathrm{R}_I \beta)$. From the unbounded semantics we have that there is a path $\pi \in \Pi(s)$ such that either (1) $(\forall\, m \in I) M_{IS}, \pi(m) \models \beta$ or (2) $(\exists\, m \in I)$ $(M_{IS}, \pi(m) \models \alpha$ and $(\forall\, 0 \leq i \leq m) M_{IS}, \pi(i) \models \beta)$. Let us consider the following cases:
   - Assume that (1) holds and $I \cap [0,k] = I$. Thus, $m \leq k$ for all $m \in I$, and by the inductive assumption we have that $M_k, \pi(m) \models \beta$ for all $m \in I$. Now, consider the prefix π_k of length k of π. We have that $\pi_k \in \Pi_k(s)$. Thus, by the definition of the bounded semantics we can conclude that $M_k, s \models \mathrm{E}(\alpha \mathrm{R}_I \beta)$.
   - Assume that (1) holds and $I = [b,\infty)$ for some $b \geq 0$. Since the set of state of M_{IS} is finite, we have that the path π must be an a-loop. Thus, we have that $loop(\pi, b) \neq \emptyset$. Since $k = \lfloor max\{Max(BI(\varphi)), Max(UI(\varphi))\}/ \mid M_{IS} \mid \rfloor * \mid M_{IS} \mid + \mid M_{IS} \mid$, the prefix of π of the length k must belong to $\Pi_k(s)$. Further, by the inductive assumption we have that $M_k, \pi(m) \models \beta$ for all $b \leq m \leq k$. Therefore, by the definition of the bounded semantics we have that $M_k, s \models \mathrm{E}(\alpha \mathrm{R}_I \beta)$.

– Assume that (1) holds and $I = (b, \infty)$ for some $b \geq 0$. The proof is analogous to the case above. Assume that (2) holds. The proof is analogous to the until case.

The main theorem of this section states that $\lfloor max\{Max(BI(\varphi)), Max(UI(\varphi))\} / \mid M_{IS} \mid \rfloor * \mid M_{IS} \mid + \mid M_{IS} \mid$-bounded satisfaction is equivalent to the unbounded one.

Theorem 1. *Let M_{IS} be a model associated to an interpreted system, M_k its k-model, $k = \lfloor max\{Max(BI(\varphi)), Max(UI(\varphi))\}/ \mid M_{IS} \mid \rfloor * \mid M_{IS} \mid + \mid M_{IS} \mid$, and φ a* RTECTLK *formula. Then, $M_{IS} \models \varphi$ iff $M_{IS} \models_k \varphi$.*

Proof. The proof follows from Lemmas 1 and 2.

3.3 Submodels of k-models

The previous subsection ends with the conclusion that to check whether a model M_{IS} associated to an interpreted system *IS* is a model to a RTECTLK formula φ under consideration, it is enough to check whether φ holds on the k-model M_k, for some $k \leq \lfloor max\{Max(BI(\varphi)), Max(UI(\varphi))\}/ \mid M_{IS} \mid \rfloor * \mid M_{IS} \mid + \mid M_{IS} \mid$. In this subsection we prove that φ holds on M_{IS} if and only if φ holds on a *submodel* of M_k.

Definition 6. *A* submodel *of a k-model $M_k = (S, P_k, \{\sim_i\}_{i \in \mathcal{AG}}, \iota, \mathcal{V})$ is a tuple $M'(s) = (S', P'_k, \{\sim'_i\}_{i \in \mathcal{AG}}, s, \mathcal{V}')$ rooted at state $s \in S$ such that $P'_k \subseteq P_k$, $S' = \{r \in S \mid (\exists \pi \in P'_k)(\exists\, i \leq k)\pi(i) = r\} \cup \{s\}$, $\sim'_i = \sim_i \cap (S' \times S')$ for each $i \in \mathcal{AG}$, and $\mathcal{V}' = \mathcal{V} \restriction S'$.*

Satisfaction for RTECTLK over a submodel $M'(s)$ is defined as for M_k.

Now, we introduce a function f_k that gives a bound on the number of k-paths in the submodel $M'(s)$. Namely, the function f_k: RTECTLK $\to \mathbb{N}$ is defined by: $f_k(p) = f_k(\neg p) = 0$, where $p \in \mathcal{PV}'$, $f_k(\alpha \vee \beta) = max\{f_k(\alpha), f_k(\beta)\}$, $f_k(\mathrm{E}(\alpha \mathrm{U}_I \beta)) = k \cdot f_k(\alpha) + f_k(\beta) + 1$, $f_k(Y\alpha) = f_k(\alpha) + 1$, for $Y \in \{\overline{\mathrm{K}}_i, \overline{\mathrm{D}}_\Gamma, \overline{\mathrm{E}}_\Gamma\}$, $f_k(\alpha \wedge \beta) = f_k(\alpha) + f_k(\beta)$, $f_k(\mathrm{E}(\alpha \mathrm{R}_I \beta)) = (k+1) \cdot f_k(\beta) + f_k(\alpha) + 1$, $f_k(\overline{\mathrm{C}}_\Gamma \alpha) = f_k(\alpha) + k$.

In the following, we will show that the validity of φ in M_k is equivalent to the validity of φ in $M'(s)$ provided that the bound k is chosen by means of the function f_k. We start with an auxiliary Lemma 3 that can be proved by straightforward induction on the length of a RTECTLK formula φ.

Lemma 3. *Let $M'(s)$ and $M''(s)$ be two submodels of M_k with $P'_k \subseteq P''_k$, and φ a* RTECTLK *formula. If $M'(s) \models_k \varphi$, then $M''(s) \models_k \varphi$.*

Lemma 4. *$M_k, s \models \varphi$ iff there is a submodel $M'(s)$ of M_k with $\mid P'_k \mid \leq f_k(\varphi)$ such that $M'(s), s \models \varphi$.*

Proof. The implication from right to left is straightforward. To prove the implication left to right, we will use induction on the length of φ.

The "left-to-right" implication follows directly for the propositional variables and their negations. Next, assume that the hypothesis holds for all the proper sub-formulae of φ. If $\varphi = \alpha \wedge \beta$ or $\varphi = \alpha \vee \beta$, then the proof is straightforward. For the epistemic operators, i.e., $\varphi = \overline{\mathrm{K}}_l \alpha, \overline{\mathrm{E}}_\Gamma \alpha, \overline{\mathrm{D}}_\Gamma \alpha, \overline{\mathrm{C}}_\Gamma \alpha$, the proof is like in [11] (see Lemma 3). For $\varphi = \mathrm{EX}\alpha$ the proof is like in [12] (see Lemma 3). Consider φ to be of the following forms:

- Let $\varphi = \mathrm{E}(\alpha \mathrm{U}_I \beta)$ and $M_k, s \models \varphi$. By the definition, there is a k-path $\pi \in \Pi_k(s)$ such that $(\exists m \in I)(M_k, \pi(m) \models \beta \text{ and } (\forall 0 \leq i < m) M_k, \pi(i) \models \alpha)$. Hence, by the inductive assumption, (1) for all i such that $0 \leq i < m$ there are submodels $M^i(\pi(i))$ of M_k with $\mid P_k^i \mid \leq f_k(\alpha)$ and $M^i(\pi(i)), \pi(i) \models \alpha$, (2) and there is a submodel $M^m(\pi(m))$ of M_k with $\mid P_k^m \mid \leq f_k(\beta)$ and $M^m(\pi(m)), \pi(m) \models \beta$. Consider a submodel $M'(s)$ of M_k such that $P'_k = \bigcup_{i=0}^{m} P_k^i \cup \{\pi\}$. Thus, by the construction of $M'(s)$, we have that $\pi \in P'_k$. Therefore, since conditions (1), and (2) hold, by the definition of the bounded satisfaction, we have that $M'(s), s \models \mathrm{E}(\alpha \mathrm{U}_I \beta)$ and $\mid P'_k \mid \leq k \cdot f_k(\alpha) + f_k(\beta) + 1$.
- Let $\varphi = \mathrm{E}(\alpha \mathrm{R}_I \beta)$ and $M_k, s \models \varphi$. By the definition of bounded semantics, there is a k-path $\pi \in \Pi_k(s)$ such that

$$(\exists j \in I)(M_k, \pi(j) \models \alpha \text{ and } (\forall 0 \leq i \leq j) M_k, \pi(i) \models \beta) \text{ or} \tag{2}$$

$$(\forall j \in I)(M_k, \pi(j) \models \beta \text{ and } I \cap [0,k] = I) \text{ or} \tag{3}$$

$$(\forall \mathit{left}(I) < j \leq k)(M_k, \pi(j) \models \beta \text{ and } \mathit{loop}(\pi, \mathit{left}(I)+1) \neq \emptyset) \text{ or} \tag{4}$$

$$(\forall \mathit{left}(I) \leq j \leq k)(M_k, \pi(j) \models \beta \text{ and } \mathit{loop}(\pi, \mathit{left}(I)) \neq \emptyset)\,. \tag{5}$$

  Let us consider the four cases. First, assume that condition (2) holds. Then, by the inductive assumption, for all i such that $0 \leq i \leq j$ there are submodels $M^i(\pi(i))$ of M_k with $\mid P_k^i \mid \leq f_k(\beta)$ and

$$M^i(\pi(i)), \pi(i) \models \beta\,, \tag{6}$$

  and there is a submodel $M^{''}(\pi(m))$ of M_k with $\mid P_k^{''} \mid \leq f_k(\alpha)$ and

$$M^{''}(\pi(m)), \pi(m) \models \alpha\,. \tag{7}$$

  Consider the submodel $M'(s)$ of M_k such that $P'_k = \bigcup_{i=0}^{j} P_k^i \cup P_k^{''} \cup \{\pi\}$. Thus, by the construction of $M'(s)$, we have that $\pi \in P'_k$. Therefore, since the conditions (2), (6) and (7) hold, by the definition of the bounded satisfaction we have that $M'(s), s \models \mathrm{E}(\alpha \mathrm{R}_I \beta)$ and $\mid P'_k \mid \leq (k+1) \cdot f_k(\beta) + f_k(\alpha) + 1$.
  Assume now that condition (3) holds. Then, by the inductive assumption, for all j such that $j \in I$ there are submodels $M^j(\pi(j))$ of M_k with $\mid P_k^j \mid \leq f_k(\beta)$ and

$$(M^j(\pi(j)), \pi(j) \models \beta)\,. \tag{8}$$

  Consider the submodel $M'(s)$ of M_k such that $P'_k = \bigcup_{j \in I} P_k^j \cup \{\pi\}$. Thus, by the construction of $M'(s)$, we have that $\pi \in P'_k$. Therefore, since conditions (2) and (8) hold, by the definition of bounded satisfaction we have that $M'(s), s \models \mathrm{E}(\alpha \mathrm{R}_I \beta)$ and $\mid P'_k \mid \leq (k+1) \cdot f_k(\beta) + f_k(\alpha) + 1$.
  The remaining two cases can be proved similarly.

The following theorem shows that a RTECTLK formula φ holds on M_{IS} if and only if φ holds on a *submodel* $M'(\iota)$ of M_k.

Theorem 2. *Let M_{IS} be a model associated to an interpreted system, M_k its k-model, φ a* RTECTLK *formula, and $k = \lfloor max\{Max(BI(\varphi)), Max(UI(\varphi))\}/ \mid M_{IS} \mid \rfloor * \mid M_{IS} \mid + \mid M_{IS} \mid$. Then, $M_{IS} \models \varphi$ iff there exists a submodel $M'(\iota)$ of M_k with $P'_k \leq f_k(\varphi)$ and $M'(\iota) \models_k \varphi$.*

Proof. Follows from Theorem 1 and Lemma 4.

3.4 Translation to Boolean Formulae

Given a RTECTLK formula φ and a model M_{IS}. As it was already mentioned, the main idea of the BMC method for the RTECTLK consists in translating the model checking problem for the RTECTLK into the satisfiability problem of a propositional formula $[M_{IS}, \varphi]_k$ that is a conjunction of two formulae, i.e.: $[M_{IS}, \varphi]_k = [M_{IS}^{\varphi,\iota}]_k \wedge [\varphi]_{M_k}$. $[M_{IS}^{\varphi,\iota}]_k$ represents all the possible submodels of M_{IS} that consist of $f_k(\varphi)$ k-paths of M_{IS}, and $[\varphi]_{M_k}$ encodes constraints that must be satisfied by $f_k(\varphi)$-submodels of M_{IS} for φ to be satisfied. Once this translation is defined, checking satisfiability of a resulting formula can be done by means of a SAT-checker.

In order to define the formula $[M_{IS}, \varphi]_k$ we proceed as follows. We assume that each state s of M_{IS} is encoded by a bit-vector whose length, say n, depends on the number of agents' local states. Thus, each state s of M_{IS} we can represent by a vector $w = (w[1], \ldots, w[n])$ of propositional variables (usually called *state variables*) to which we refer to as a *global state variable*. A finite sequence $(w_0, \ldots, w_k)$ of global state variables we call a *symbolic k-path*. Since, in general, we may need to consider more than one symbolic k-paths, we introduce a notion of the j-th symbolic k-path, which is denoted by $(w_{0,j}, \ldots, w_{k,j})$, where $w_{i,j}$ are global state variables for $1 \leq j \leq f_k(\varphi)$ and $0 \leq i \leq k$. Note that the exact number of necessary symbolic k-paths depends on the checked formula φ, and it can be calculated by means of the function f_k.

For two global state variables w, w', we define the following propositional formulae:

- $I_s(w)$ is a formula over w that is true for a valuation s_w of w iff $s_w = s$.
- $p(w)$ is a formula over w that is true for a valuation s_w of w iff $p \in \mathcal{V}(s_w)$ (encodes a set of states of M_{IS} in which $p \in \mathcal{PV}$ holds).
- $H(w, w')$ is a formula over w and w' that is true for two valuations s_w of w and $s_{w'}$ of w' iff $s_w = s_{w'}$ (encodes equivalence of two global states).
- $H_i(w, w')$ is a formula over w, w' that is true for two valuations s_w of w and $s_{w'}$ of w' iff $l_i(s_w) = l_i(s_{w'})$ (encodes equivalence of local states of agent i).
- $\mathcal{R}(w, w')$ is a formula over w, w' that is true for two valuations s_w of w and $s_{w'}$ of w' iff $(s_w, s_{w'}) \in T$ (encodes the transition relation of M_{IS}).

Let a, b be vectors of propositional formulae built only over propositional constants *true* and *false*; note that the vectors a and b can be seen as bit-vectors. We define the following auxiliary propositional formulae:

- $\Theta\colon \{0, \ldots, 2^t - 1\} \to \{true, false\}^t$ is a function that converts each natural number smaller than 2^t to the bit-vector of the length t.
- $eq(a,b) := \bigwedge_{i=1}^{t} a[i] \Leftrightarrow b[i]$, $ge(a,b) := \bigvee_{i=1}^{t} \big(a[i] \wedge \neg b[i] \wedge \bigwedge_{j=i+1}^{t} a[j] \Leftrightarrow b[j]\big)$, $geq(a,b) := eq(a,b) \vee ge(a,b)$, $le(a,b) := \neg geq(a,b)$, $leq(a,b) := \neg ge(a,b)$,
- $IN(j, I) := \begin{cases} le(\Theta(a), \Theta(j)) \wedge le(\Theta(j), \Theta(b)), & \text{if } I = (a,b) \\ leq(\Theta(a), \Theta(j)) \wedge leq(\Theta(j), \Theta(b)), & \text{if } I = [a,b] \\ leq(\Theta(a), \Theta(j)) \wedge le(\Theta(j), \Theta(b)), & \text{if } I = [a,b) \\ le(\Theta(a), \Theta(j)) \wedge leq(\Theta(j), \Theta(b)), & \text{if } I = (a,b] \\ ge(\Theta(a), \Theta(j)), & \text{if } I = (a,\infty) \\ geq(\Theta(a), \Theta(j)), & \text{if } I = [a,\infty) \end{cases}$
  The formula $IN(j, I)$ encodes that $j \in I$.

The propositional formula $[M_{IS}, \varphi]_k$ is defined over state variables $w_{0,0}$, $w_{n,m}$, for $0 \leq m \leq k$ and $1 \leq n \leq f_k(\varphi)$. We start off with the definition of its first conjunct, i.e., the definition of $[M_{IS}^{\varphi,\iota}]_k$, which constrains the $f_k(\varphi)$ symbolic k-paths to be valid k-path of M_k. Namely,

$$[M_{IS}^{\varphi,\iota}]_k := I_\iota(w_{0,0}) \wedge \bigwedge_{n=1}^{f_k(\varphi)} \bigwedge_{m=0}^{k-1} \mathcal{R}(w_{m,n}, w_{m+1,n}). \tag{9}$$

The formula $[\varphi]_{M_k} = [\varphi]_k^{[0,0]}$, is inductively defined as follows:

$[p]_k^{[m,n]} := p(w_{m,n}), \quad [\alpha \wedge \beta]_k^{[m,n]} := [\alpha]_k^{[m,n]} \wedge [\beta]_k^{[m,n]},$

$[\neg p]_k^{[m,n]} := \neg p(w_{m,n}), [\alpha \vee \beta]_k^{[m,n]} := [\alpha]_k^{[m,n]} \vee [\beta]_k^{[m,n]},$

$[\mathrm{EX}\alpha]_k^{[m,n]} := \bigvee_{i=1}^{f_k(\varphi)} (H(w_{m,n}, w_{0,i}) \ \wedge \ [\alpha]_k^{[1,i]}),$

$[\overline{\mathrm{K}}_l\alpha]_k^{[m,n]} := \bigvee_{i=1}^{f_k(\varphi)} (I_\iota(w_{0,i}) \wedge \bigvee_{j=0}^{k} ([\alpha]_k^{[j,i]} \wedge H_l(w_{m,n}, w_{j,i}))),$

$[\overline{\mathrm{D}}_\Gamma\alpha]_k^{[m,n]} := \bigvee_{i=1}^{f_k(\varphi)} (I_\iota(w_{0,i}) \wedge \bigvee_{j=0}^{k} ([\alpha]_k^{[j,i]} \wedge \bigwedge_{l\in\Gamma} H_l(w_{m,n}, w_{j,i}))),$

$[\overline{\mathrm{E}}_\Gamma\alpha]_k^{[m,n]} := \bigvee_{i=1}^{f_k(\varphi)} (I_\iota(w_{0,i}) \wedge \bigvee_{j=0}^{k} ([\alpha]_k^{[j,i]} \wedge \bigvee_{l\in\Gamma} H_l(w_{m,n}, w_{j,i}))),$

$[\overline{\mathrm{C}}_\Gamma\alpha]_k^{[m,n]} := [\bigvee_{i=1}^{k} (\overline{\mathrm{E}}_\Gamma)^i\alpha]_k^{[m,n]}$

$[\mathrm{E}(\alpha \mathrm{U}_I \beta)]_k^{[m,n]} := \bigvee_{i=1}^{f_k(\varphi)} (H(w_{m,n}, w_{0,i}) \wedge \bigvee_{j=0}^{k} ([\beta]_k^{[j,i]} \wedge \bigwedge_{l=0}^{j-1} [\alpha]_k^{[l,i]} \wedge IN(j, I))),$

$$\begin{aligned}[\mathrm{E}(\alpha \mathrm{R}_I \beta)]_k^{[m,n]} := \bigvee_{i=1}^{f_k(\varphi)} [H(w_{m,n}, w_{0,i}) \wedge (\bigvee_{j=0}^{k} ([\alpha]_k^{[j,i]} \ \wedge \ \bigwedge_{l=0}^{j} [\beta]_k^{[l,i]} \wedge IN(j, I)) \\ \vee \bigwedge_{j=left(I)}^{min\{right(I),k\}} ([\beta]_k^{[j,i]} \wedge IN(j, I) \wedge IN(j, [0,k])), \\ \vee \bigwedge_{j=left(I)+1}^{k} ([\beta]_k^{[j,i]} \wedge \bigvee_{l=left(I)+1}^{k} \mathcal{R}(w_{k,i}, w_{l,i})), \\ \vee \bigwedge_{j=left(I)}^{k} ([\beta]_k^{[j,i]} \wedge \bigvee_{l=left(I)}^{k} \mathcal{R}(w_{k,i}, w_{l,i}))].\end{aligned}$$

This fully defines the encoding of the formula $[M_{IS}, \varphi]_k$.

Now we show that the validity of a RTECTLK formula φ on a submodel $M'(s)$, defined by using the function f_k, is equivalent to the satisfiability of the formula $[M_{IS}, \varphi]_k$. Once we show this, we can conclude that the validity of φ on the model M_{IS} is equivalent to the satisfiability of $[M_{IS}, \varphi]_k$ (see Theorem 3).

Lemma 5. *Let M_{IS} be a model, M_k its k-model, and φ a* RTECTLK *formula. For each state s of M_{IS}, the following holds: $[M_{IS}^{\varphi,s}]_k \wedge [\varphi]_{M_k}$ is satisfiable iff there is a submodel $M'(s)$ of M_k with $\mid P'_k \mid \leq f_k(\varphi)$ such that $M'(s), s \models \varphi$.*

Proof. $(=>)$ Let $[M_{IS}^{\varphi,s}]_k \wedge [\varphi]_{M_k}$ be satisfiable. By the definition of the translation, the propositional formula $[\varphi]_{M_k}$ encodes all the sets of k-paths of size $f_k(\varphi)$ which satisfy the formula φ. By the definition of the unfolding of the transition relation, the propositional formula $[M^{\varphi,s}]_k$ encodes $f_k(\varphi)$ symbolic k-paths to be valid k-paths of M_k. Hence, there is a set of k-paths in M_k, which satisfies the formula φ of size smaller or equal to $f_k(\varphi)$. Thus, we conclude that there is a submodel $M'(s)$ of M_k with $\mid P'_k \mid \leq f_k(\varphi)$ such that $M'(s), s \models \varphi$.

$(<=)$ The proof is by induction on the length of φ. The lemma follows directly for the propositional variables and their negations. For $\varphi = \alpha \vee \beta, \ \alpha \wedge \beta, \ \mathrm{EX}\alpha$ the proof is like in [12] (see Lemma 6.3). For epistemic operators, i.e., $\varphi = \overline{\mathrm{K}}_l\alpha, \overline{\mathrm{E}}_\Gamma\alpha, \overline{\mathrm{D}}_\Gamma\alpha, \overline{\mathrm{C}}_\Gamma\alpha$, the proof is like in [11] (see Lemma 3). Consider the following cases:

A. Let $\varphi = \mathrm{E}(\alpha \mathrm{U}_I \beta)$ and $M'_k, s \models \varphi$ with $\mid P'_k \mid \leq f_k(\varphi)$. By the definition of the bounded semantics we have that there is a k-path $\pi \in \Pi_k(s)$ and exists $0 \leq j \leq k$ such that $j \in I$ and $M'_k, \pi(j) \models \beta$ and $M'_k, \pi(t) \models \alpha$ for all $0 \leq t < j$. Hence, by induction we obtain that for some $j \leq k$ and $j \in I$ the propositional formula $IN(j, I)$ is satisfiable, and also the following propositional formula are satisfiable: $[\beta]_k^{[0,0]} \wedge [M^{\beta,\pi(j)}]_k$ and $[\alpha]_k^{[0,0]} \wedge [M^{\alpha,\pi(t)}]_k$, for each $t \leq j$. Let i be the index of a new symbolic k-path which satisfies the formula $H(w_{0,0}, w_{0,i})$. Because of the above and Lemma 3, we obtain that the propositional formula $H(w_{0,0}, w_{0,i}) \wedge \bigvee_{j=0}^{k} \left([\beta]_k^{[j,i]} \wedge IN(j, I) \wedge \bigwedge_{t=0}^{j-1} [\alpha]_k^{[t,i]}\right) \wedge [M^{\mathrm{E}(\alpha \mathrm{U}_I \beta),\pi(0)}]_k$ is satisfiable. Therefore, the following propositional formula is satisfiable too: $\bigvee_{i=1}^{f(\varphi)} \left(H(w_{m,n}, w_{0,i}) \wedge \bigvee_{j=0}^{k} ([\beta]_k^{[j,i]} \wedge IN(j, I) \wedge \bigwedge_{t=0}^{j-1} [\alpha]_k^{[t,i]})\right) \wedge [M^{\mathrm{E}(\alpha \mathrm{U} \beta),\pi(0)}]_k$. Hence, by the definition of the translation of a RTECTLK formula, the above formula is equal to the propositional formula $[\mathrm{E}(\alpha \mathrm{U}_I \beta)]_k^{[0,0]} \wedge [M^{\mathrm{E}(\alpha \mathrm{U}_I \beta),\pi(0)}]_k$.

B. Let $\varphi = \mathrm{E}(\alpha \mathrm{R}_I \beta)$. The formal proof of the case is tedious and can be done in a similar way as the one for the EU_I operator. So, we have decided not to present it in detail.

Theorem 3. *Let M_{IS} be a model, and φ a* RTECTLK *formula. Then, $M_{IS} \models \varphi$ iff there exists $k > 0$ such that $[\varphi]_{M_k} \wedge [M^{\varphi,\iota}]_k$ is satisfiable.*

Proof. It follows from Theorem 2 and Lemma 5.

4 A Train Controller System

We now give an example of how the RTECTLK formalism can be used to reason about MASs and, in particular, how the RTECTLK properties of a MAS can be verified using the BMC technique described in the paper. The system we consider is a train controller (TC), adapted from [10], and it consists of two trains and a controller. In the system it is assumed that each train uses its own circular track (the Eastbound or the Westbound) for travelling in one direction. At one point, both trains have to pass through a tunnel, but because there is only one track in the tunnel, trains arriving from each direction cannot use it simultaneously. There are traffic lights on both sides of the tunnel, which can be either red or green. The controller is notified by both trains when they request entry to the tunnel or when they leave the tunnel, and controls the colour of the traffic lights.

In the interpreted systems framework, the TC system can be modelled by three agents: two trains (agents 1 and 3) and a controller (agent 2). Their local states are the following: $L_1 = \{away_1, wait_1, tunnel_1\}$, $L_2 = \{green, red\}$, $L_3 = \{away_2, wait_2, tunnel_2\}$. The state $away_i$ represents the initial state of train i. The state $wait_i$ represents that train i has arrived at the tunnel. The state $tunnel_i$ represents that train i is in the tunnel. The states *green* and *red* represent the colour of the traffic lights.

Given the sets of local states for the above tree agents, the following sets of actions are available to the agents: $Act_1 = \{a_1, a_2, a_3, \epsilon\}$, $Act_2 = \{a_2, a_3, a_5, a_6, \epsilon\}$, $Act_3 = \{a_4, a_5, a_6, \epsilon\}$. Their meaning is the following: a_1 (a_4) – train 1 (train 2) has arrived at the tunnel; a_2 (a_5) – colour of the traffic lights for train 1 (train 2) is green; a_3 (a_6) –

train 1 (train 2) has left the tunnel. The protocols executed by agents are defined by: $P_1(away_1) = \{a_1\}$, $P_1(wait_1) = \{a_2\}$, $P_1(tunnel_1) = \{a_3\}$, $P_3(away_2) = \{a_4\}$, $P_3(wait_2) = \{a_5\}$, $P_3(tunnel_2) = \{a_6\}$, $P_2(green) = \{a_2, a_5\}$, $P_2(red) = \{a_3, a_6\}$. The evolution of the TC system is defined by means of an evolution function t: $(L_1 \times L_2 \times L_3) \times Act \rightarrow (L_1 \times L_2 \times L_3)$, where Act is a subset of $Act_1 \times Act_2 \times Act_3$. More precisely, let us assume that the TC system starts from the following initial state: $(away_1, green, away_2)$, and let $green = g$, $red = r$, $tunnel_i = t_i$, $away_i = aw_i$, $wait_i = w_i$, for $i = 1, 2$. Then, the evolution function for the TC system is defined as follows:
$t((aw_1, g, aw_2), (a_1, \epsilon, \epsilon)) = (w_1, g, aw_2)$, $t((aw_1, g, aw_2), (\epsilon, \epsilon, a_4)) = (aw_1, g, w_2)$,
$t((w_1, g, aw_2), (a_2, a_2, \epsilon)) = (t_1, r, aw_2)$, $t((w_1, g, aw_2), (\epsilon, \epsilon, a_4)) = (w_1, g, w_2)$,
$t((aw_1, g, w_2), (a_1, \epsilon, \epsilon)) = (w_1, g, w_2)$, $t((aw_1, g, w_2), (\epsilon, a_5, a_5)) = (aw_1, r, t_2)$,
$t((t_1, r, aw_2), (\epsilon, \epsilon, a_4)) = (t_1, r, w_2)$, $t((t_1, r, aw_2), (a_3, a_3, \epsilon)) = (aw_1, g, aw_2)$,
$t((w_1, g, w_2), (a_2, a_2, \epsilon)) = (t_1, r, w_2)$, $t((w_1, g, w_2), (\epsilon, a_5, a_5)) = (w_1, r, t_2)$,
$t((aw_1, r, t_2), (a_1, \epsilon, \epsilon)) = (w_1, r, t_2)$, $t((aw_1, r, t_2), (\epsilon, a_6, a_6)) = (aw_1, g, aw_2)$,
$t((t_1, r, w_2), (a_3, a_3, \epsilon)) = (aw_1, g, w_2)$, $t((w_1, r, t_2), (\epsilon, a_6 a_6)) = (w_1, g, aw_2)$.
It determines not only the set of reachable global states $S \subseteq L_1 \times L_2 \times L_3$, but also gives the transition relation T; namely, for all the $s,\ s' \in S$, $(s, s') \in T$ iff there exists $act \in Act$ such that $t(s, act) = s'$.

We have now defined reachable states, actions, protocols, and transitions of the model $M_{IS} = (S, T, \{\sim_i\}_{i \in \mathcal{AG}}, \iota, \mathcal{V})$ for the TC system. Let $\mathcal{PV} = \{inW_1, inW_2, inT_1, inT_2\}$ be a set of propositional variables, which we find useful in analysis of the scenario of the TC system. To conclude, we define a valuation function $\mathcal{V}$: $S \rightarrow 2^{\mathcal{PV}}$ as follows: $inT_1 \in \mathcal{V}(s)$ if $l_1(s) = tunnel_1$, $inT_2 \in \mathcal{V}(s)$ if $l_2(s) = tunnel_2$, $inW_1 \in \mathcal{V}(s)$ if $l_1(s) = wait_1$, and $inW_2 \in \mathcal{V}(s)$ if $l_2(s) = wait_2$.

We have now completed the definition of the model M_{IS} for the TC system, so we can proceed to the verification of the model M_{IS} by means of the BMC method. As an example, let us verify the following properties:

- There exists a behaviour of the TC system such that agent Train i (for $i = 1$ or $i = 3$) considers possible a situation in which it is not true that being in the tunnel always leads to the same situation within a bounded period of time, i.e., within n time units for $n \geq 2$. This property can be formalised by the following RTECTLK formula: $\alpha := \mathrm{EF}_{[0,\infty]}\overline{\mathrm{K}}_i(inT_i \wedge \mathrm{EG}_{[0,n]}(\neg inT_i))$.
- There exists a behaviour of the TC system such that agent Train i (for $i = 1$ or $i = 3$) considers possible a situation in which both he is in his waiting state and in the next state he still waits for the permission to enter the tunnel. This property can be formalised by the following RTECTLK formula: $\beta := \mathrm{EF}_{[0,\infty]}\overline{\mathrm{K}}_i(inW_i \wedge \mathrm{EX}(\neg inT_i))$.
- The model of the TC system does not satisfy the property of n-bounded fairness, i.e., each train should be scheduled for entering the tunnel at least once every n steps of the system, for $n \geq 3$. This property can be formalised by the following RTECTLK formula: $\gamma := \mathrm{EG}_{[0,n]}(\neg inT_1) \vee \mathrm{EG}_{[0,n]}(\neg inT_2) \vee \mathrm{EF}_{[0,\infty]}(\neg inT_1 \wedge \mathrm{EXEG}_{[0,n-1]}(\neg inT_1)) \vee \mathrm{EF}_{[0,\infty]}(\neg inT_2 \wedge \mathrm{EXEG}_{[0,n-1]}(\neg inT_2))$.

According to the BMC algorithm for the RTECTLK, to perform the method for the TC system and properties α, β, and γ, first, all the states of M_{IS} have to be

represented by bit-vectors. To do this we have to encode all the possible configurations of the system in terms of local states of agents. So, assume that we have the following bit representation of local states. For Train 1 we take $away_1 = (0,0)$, $wait_1 = (0,1)$, $tunnel_1 = (1,0)$, for Train 2 $away_2 = (0,0)$, $wait_2 = (0,1)$, $tunnel_2 = (1,0)$, and for Controller $green = (0)$ and $red = (1)$. So, the global states of the TC system have the following encoding: $(away_1, green, away_2) = (0,0;\ 0;\ 0,0)$, $(wait_1, green, away_2) = (0,1;\ 0;\ 0,0)$, $(tunnel_1, red, away_2) = (1,0;\ 1;\ 0,0)$, etc. In other words, we need 5 state variables $(s[0], \ldots, s[4])$ to encode all the possible configurations of the TC system. Next, the transition relation of M_{IS} has to be encoded by a propositional formula, and formulae α, β and γ have to be translated over all the possible $f_k(\alpha) = 3$ (resp. $f_k(\beta) = 3, f_k(\gamma) = 3$) submodels of M_{IS}.

To proceed with the translation of the transition relation of M_{IS}, the first thing we need to translate is the initial state $\iota = (away_1, green, away_2)$ of the TC system that is represented by the bit-vector $(0,0,0,0,0)$. With this representation ι will be encoded by the following propositional formula $I_\iota(w_{0,0}) = \bigwedge_{i=0}^{4}(\neg\, w_{0,0}[i])$.

The next step is to translate the transitions $\mathcal{R}(w_{i,j}, w_{i+1,j})$, for $i = 0,..,k-1$ and $j = 1,2,3$. For simplicity we report only on the formula $\mathcal{R}(w_{0,1}, w_{1,1})$ representing the first transition of the first path. Let us consider the following transition of our counterexample: $T((away_1, green, away_2), (wait_1, green, away_2))$. The corresponding formula is:

$$\mathcal{R}(w_{0,1}, w_{1,1}) := \bigwedge_{i=0}^{4}(\neg w_{0,1}[i]) \wedge \neg w_{1,1}[0] \wedge w_{1,1}[1] \wedge \bigwedge_{i=2}^{4}(\neg w_{1,1})\,. \qquad (10)$$

In order to encode the whole example we should model all the transitions for all the k's starting from $k := 1$. We do not do it here.

To encode the translation of α, β and γ, first we need to encode the propositions used in these formulae. Namely, $inT_1(w) := (w[0] \wedge \neg w[1])$, which means that inT_1 holds at all the global states with the first local state equal to $(1,0)$. $inT_2(w) := (w[3] \wedge \neg w[4])$, which means that inT_2 holds at all the global states with the third local state equal to $(1,0)$. $inW_1(w) := (\neg w[0] \wedge w[1])$, which means that inW_1 holds at all the global states with the first local state equal to $(0,1)$. $inW_2(w) := (\neg w[3] \wedge w[4])$, which means that inW_2 holds at all the global states with the third local state equal to $(0,1)$.

In so doing, it is sufficient to unfold the formula $[\alpha]_k^{0,0}$, $[\beta]_k^{0,0}$ and $[\gamma]_k^{0,0}$, for $k = 1,2,\ldots$, according to the definition on page 173.

Checking that the TC system satisfies the RTECTLK formulae above can now be done by feeding a SAT solver with the propositional formula generated by this method. This would produce a solution, thereby proving that the constructed propositional formula is satisfiable.

5 Conclusions

In summary, we have shown how to extend the BMC method to the RTECTLK formalism, and we have proved that the RTECTLK bounded model checking can be solved in time $O(\lfloor max\{Max(BI(\varphi)), Max(UI(\varphi))\}/ \mid M_{IS} \mid \rfloor * \mid M_{IS} \mid + \mid M_{IS} \mid)$. We have also considered a train controller system to exemplify a use of the proposed method.

The underlying semantics of time of the considered MASs and the RTECTLK formalism is discrete. However, it is also possible to consider real-time multi-agent systems under the assumption that the time is dense and properties are express in TCTLK; such an attempt has been done in [17].

Finally we should like to stress that this paper belongs to a line of research on model checking time and knowledge encompassing theoretical investigations. A comparison of the experimental results with other model checking technique for MASs we leave for further work.

References

1. Benedetti, M., Cimatti, A.: Bounded Model Checking for Past LTL. In: Proc. of the TACAS'03. LNCS, vol. 2619, pp. 18–33. Springer, Heidelberg (2003)
2. Biere, A., Cimatti, A., Clarke, E., Strichman, O., Zhu, U.: Bounded Model Checking. In: Highly Dependable Software. Advances in Computers, vol. 58, Academic Press (2003)
3. Blackburn, P., de Rijke, M., Venema, Y.: Modal Logic. Cambridge Tracts in Theoretical Computer Science, vol. 53, Cambridge University Press (2001)
4. Bryant, R.: Binary Decision Diagrams and Beyond: Enabling Technologies for Formal Verification. In: Proc. of the ICCAD'95, pp. 236–243 (1995)
5. Clarke, E.M., Grumberg, O., and Peled, D.A.: Model Checking. The MIT Press, Cambridge, Massachusetts (1999)
6. Copty, F., Fix, L., Fraer, R., Giunchiglia, E., Kamhi, G., Tacchella, A., Vardi, M.Y.: Benefits of Bounded Model Checking at an Industrial Setting. In: Proc. of the CAV'01. LNCS, vol. 2102, pp. 436–453. Springer, Heidelberg (2001)
7. Emerson, E.A., Mok, A.K., Sistla, A.P., Srinivasan, J.: Quantitative Temporal Reasoning. Real-Time Systems 4(4), 331–352 (1992)
8. Fagin, R., Halpern, J.Y., Moses, Y., Vardi, M.Y. Reasoning about Knowledge. MIT Press, Cambridge (1995)
9. Fagin, R., Halpern, J.Y., Vardi, M.Y. What Can Machines Know? On the Properties of Knowledge in Distributed Systems. Journal of the ACM 39(2), 328–376 (1992)
10. van der Hoek, W., Wooldridge, M.: Cooperation, Knowledge, and Time: Alternating-time Temporal Epistemic Logic and its Applications. Studia Logica 75(1), 125–157 (2003)
11. Penczek, W., Lomuscio, A.: Verifying Epistemic Properties of Multi-agent Systems via Bounded Model Checking. Fundamenta Informaticae 55(2), 167–185 (2003)
12. Penczek, W., Woźna, B., Zbrzezny, A.: Bounded Model Checking for the Universal Fragment of CTL. Fundamenta Informaticae 51(1-2), 135–156 (2002)
13. Raimondi, F., Lomuscio, A.: Automatic Verification of Multi-agent Systems by Model Checking via OBDDs. Journal of Applied Logic (2005)
14. Sorea, M.: Bounded Model Checking for Timed Automata. In: Proc. of the MTCS'02. ENTCS, vol. 68, Elsevier Science Publishers (2002)
15. van der Meyden, R., Su, K.: Symbolic Model Checking the Knowledge of the Dining Cryptographers. In: Proc. of the CSFW'04. pp. 280–291. IEEE Computer Society (2004)
16. Woźna, B.: Bounded Model Checking for the Universal Fragment of CTL*. Fundamenta Informaticae 63(1), 65–87 (2004)
17. Woźna, B., Lomuscio, A., Penczek, W.: Bounded Model Checking for Knowledge over Real Time. In: Proc. of the AAMAS'05, vol. I, pp. 165–172. ACM Press (2005)