The Complexity of Epistemic Model Checking: Clock Semantics and Branching Time*

X. Huang and R. van der Meyden

School of Computer Science and Engineering,
University of New South Wales
email: {xiaoweih,meyden}@cse.unsw.edu.au

May 30, 2011

Abstract

In the clock semantics for epistemic logic, two situations are indistinguishable for an agent when it makes the same observation and the time in the situations is the same. The paper characterizes the complexity of model checking branching time logics of knowledge in finite state systems with respect to the clock semantics.

1 Introduction

Epistemic logic has been shown to provide a useful formalism for reasoning about systems in which the representation of agents' states of uncertainty is a critical factor [6]. The range of problems to which epistemic logic has been applied encompasses distributed and multiagent systems, computer security, diagnosis and recoverability. These applications have motivated the development of verification techniques based on logics that combine temporal and epistemic features.

In particular, model checking combinations of temporal and epistemic logics has been a topic of recent interest. In model checking, the verification problem is treated as the problem of checking that a formula is satisfied in a given model. For suitably restricted representations of models and specification languages this problem can be shown to be decidable. Model checkers implement this decidable problem using a variety of sophisticated heuristics and symbolic implementation techniques.

Operators in the logic of knowledge can be given a number of different semantics, depending on the basic information from which an agent derives what it knows. For example, we can treat agent's knowledge as those facts that it can derive from just its current observation. This semantics is basis of much of the literature on model checking temporal and epistemic logics. However, the model checking problem can also be shown to be decidable for stronger interpretations of knowledge. In this paper, we consider the model checking problem for an interpretation of knowledge in which an agent's knowledge is taken to be what it can derive from its current observation plus the current clock value. We call this the clock semantics for knowledge. The significance of the clock semantics is that many systems are built with clocks, and they are used in protocols, e.g., for timeouts. The observational semantics is too weak to capture information present in clock values.

The model checking problem with respect to the clock semantics has previously been shown to be decidable for linear time temporal logics extended by knowledge operators by Engelhardt et al [5]. In this paper we consider the effect of taking the temporal basis for the specification language to be instead a branching time temporal logic. We show that this combination also leads to a decidable model checking problem, and characterise its complexity for a

*This paper is an extended version of a paper to appear in ECAI-2010. Work supported by Australian Research Council Linkage Grant LP0882961 and Defence Research and Development Canada (Valcartier) contract W7701-082453.

Logic	Bound	Combined Complexity	Model Complexity	Formula Complexity
$\mathrm{LTLK}_n^{\mathtt{clk}}$	upper	PSPACE	PSPACE	PSPACE
(from [5])	lower	PSPACE	PH	PSPACE
$\mathrm{CTL}^*\mathrm{K}_n^{\mathtt{clk}}$	upper	PSPACE	PSPACE	PSPACE
(this paper)	lower	PSPACE	PH	PSPACE
$\mathrm{CTLK}_n^{\mathtt{clk}}$	upper	PSPACE	PSPACE	LOGSPACE
(this paper)	lower	PSPACE	PH	LOGSPACE
$\mathrm{CTL}^-\mathrm{K}_n^{\mathtt{clk}}$	upper	P^{NP}	P^{NP}	LOGSPACE
(this paper)	lower	$\mathrm{P}^{\mathrm{NP[log]}}$	$\mathrm{P}^{\mathrm{NP[log]}}$	LOGSPACE

Table 1: Complexity Results

number of different fragments of a logic $\mathrm{CTL}^*\mathrm{K}_n^{\mathtt{clk}}$ that combines the linear time operators, branching operators and epistemic operators.

The main results of the paper are presented in Table 1, which gives the complexity (both upper and lower bounds) of each of the model checking problems we consider. For purposes of comparison, we include in this table the known results for the complexity of the linear time epistemic logic with respect to clock semantics ($\mathrm{LTLK}_n^{\mathtt{clk}}$).

It turns out that the addition of branching operators to the combination $\mathrm{LTLK}_n^{\mathtt{clk}}$ of linear time temporal logic and epistemic logic (giving the logic $\mathrm{CTL}^*\mathrm{K}_n^{\mathtt{clk}}$) does not increase the computational complexity of model checking, which remains PSPACE-complete. More interestingly, the model checking problem for the fragment $\mathrm{CTLK}_n^{\mathtt{clk}}$ based in the branching time logic CTL has the same complexity as for the linear time case ($\mathrm{LTLK}_n^{\mathtt{clk}}$), viz., PSPACE-complete. Prima facie, this result is a little surprising since the complexities of model checking linear time temporal logic and branching time logic in the absence of epistemic operators are known to be different, viz., PTIME for the branching time temporal logic CTL and PSPACE-complete for linear time temporal logic LTL.

However, our general result masks some subtleties, and a more careful analysis reveals differences between the linear time and branching time cases. In particular, a difference is apparent if one considers the complexity of model checking as a function of the size of the model, or as a function of the size of the formula. *Formula complexity* is the complexity of model checking a varying formula when the model is held fixed: this gives a measure of the complexity of the model checking as a function of the size of the formula. Alternately, *model complexity* is the complexity of the model checking problem when a formula is held fixed and the model is varied: this gives a measure of the complexity of the model checking problem as a function of the size of the model. We find that the complexity difference between CTL and LTL continues to be reflected when one adds epistemic operators in the case of formula complexity, where $\mathrm{CTLK}_n^{\mathtt{clk}}$ is LOGSPACE-complete and $\mathrm{LTLK}_n^{\mathtt{clk}}$ is PSPACE-complete (which is the same as the formula complexities for CTL and LTL respectively.) However, with respect to model complexity there is no change.

We also explore the impact of a further restriction on the set of branching time operators, taking these to be just EF (at some future time in some branch) and EX (at some successor), giving the logic $\mathrm{CTL}^-\mathrm{K}_n^{\mathtt{clk}}$. Here, we show that the complexity of model checking falls down to a low level of the polynomial hierarchy, viz P^{NP}.

The structure of the paper is as follows. In section 2, we define the syntax and semantics of the logics that we study, as well as the model checking problems that we consider. Section 3 proves the main complexity results of the paper. Section 4 discusses related work; in particular, we argue that some previous work on bounded model checking $\mathrm{CTLK}_n^{\mathtt{clk}}$ is incorrect.

2 Syntax and Semantics

We work with logics that combine temporal logics and the logic of knowledge and common knowledge for n agents. All the logics that we consider are fragments of the logic $\mathrm{CTL}^*\mathrm{K}_n^{\mathtt{clk}}$. Let *Prop* be a set of atomic propositions and

$Ags = \{1, \ldots, n\}$ be a set of n agents. The syntax of the logic $\text{CTL}^*\text{K}_n^{\text{clk}}$ is given by the following grammar:

$$\begin{array}{rl} \phi ::= & p \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid X\phi \mid F\phi \mid \phi_1 U \phi_2 \mid G\phi \mid \\ & E\phi \mid A\phi \mid \\ & K_i\phi \mid C_G\phi \end{array}$$

where $p \in Prop$ and $i \in Ags$ and $G \in \mathcal{P}(Ags) \setminus \{\emptyset\}$. The first line gives basic propositional logic plus linear time temporal operators that refer to the future. Intuitively, $X\phi$ says that ϕ holds at the next time, $F\phi$ says that ϕ holds at some future time, $\phi_1 U \phi_2$ says that ϕ_1 holds until ϕ_2 does, and $G\phi$ says that ϕ holds at all times in the future. The operators in the second line are from branching time temporal logic and refer to possible alternate futures: $A\phi$ says that ϕ holds in all possible futures and $E\phi$ says that ϕ holds in some possible future. The final line gives epistemic operators: $K_i\phi$ says that agent i knows ϕ and $C_G\phi$ says that ϕ is common knowledge to the group of agents G. If we take just the first line of this grammar we have the linear time temporal logic LTL, and adding the second line gives the branching temporal logic CTL^*. The logic $\text{LTLK}_n^{\text{clk}}$ is the fragment obtained by taking just the first and the third lines of the grammar. The branching time temporal logic CTL is obtained from CTL^* by placing a restriction on the permitted combinations of linear and branching time temporal operators: we replace these in the grammar by the restricted cases $QX\phi$, $Q\phi_i U \phi_2$, $QF\phi$ and $QG\phi$, where Q is either A or E. Adding the epistemic operators to CTL gives the logic $\text{CTLK}_n^{\text{clk}}$. The fragment $\text{CTL}^-\text{K}_n^{\text{clk}}$ is obtained by combining the epistemic operators with the branching temporal operators EX and EF. Dually, this logic contains AX and AG, so can express the important class of *epistemic safety* properties.

To give semantics to all these logics it suffices to give semantics to $\text{CTL}^*\text{K}_n^{\text{clk}}$. We do this using a variant of interpreted systems [6], specialised to the clock semantics. Let S be a set, which we call the set of global states. A *run* over S is a function $r : \mathbf{N} \to S$. A *point* is a pair (r, m) where r is a run and $n \in \mathbf{N}$. Given a set $\mathcal{R}$ of runs, we define *Points*$(\mathcal{R})$ to be the set of all points of runs $r \in \mathcal{R}$. An *interpreted system* for n agents is a tuple $\mathcal{I} = (\mathcal{R}, \sim_1, \ldots, \sim_n, \pi)$, where $\mathcal{R}$ is a set of runs over S, each $\sim_i$ is an equivalence relation on *Points*$(\mathcal{R})$ (called agent i's *indistinguishability relation*) and $\pi : S \to \mathcal{P}(Prop)$ is an interpretation function. We say that a run r' is *equivalent to a run r up to time* $m \in \mathbf{N}$ if $r'(k) = r(k)$ for $0 \leq k \leq m$.

We can define a general semantics of $\text{CTL}^*\text{K}_n^{\text{clk}}$ by means of a relation $\mathcal{I}, (r, m) \models \phi$, where $\mathcal{I}$ is an intepreted system, (r, m) is a point of $\mathcal{I}$ and ϕ is a formula. This relation is defined inductively as follows:

- $\mathcal{I}, (r, m) \models p$ if $p \in \pi(r(m))$,
- $\mathcal{I}, (r, m) \models \neg\phi$ if not $\mathcal{I}, (r, m) \models \phi$
- $\mathcal{I}, (r, m) \models \phi_1 \vee \phi_2$ if $\mathcal{I}, (r, m) \models \phi_1$ or $\mathcal{I}, (r, m) \models \phi_2$
- $\mathcal{I}, (r, m) \models E\phi$ if there exists a run $r' \in \mathcal{R}$ equivalent to r up to time m such that $\mathcal{I}, (r', m) \models \phi$
- $\mathcal{I}, (r, m) \models A\phi$ if for all runs $r' \in \mathcal{R}$ equivalent to r up to time m, we have $\mathcal{I}, (r', m) \models \phi$
- $\mathcal{I}, (r, m) \models X\phi$ if $\mathcal{I}, (r, m + 1) \models \phi$
- $\mathcal{I}, (r, m) \models \phi_1 U \phi_2$ if there exists $m' \geq m$ such that $\mathcal{I}, (r, m') \models \phi_2$, and $\mathcal{I}, (r, k) \models \phi_1$ for $m \leq k < m'$.
- $\mathcal{I}, (r, m) \models G\phi$ if $\mathcal{I}, (r, k) \models \phi$ for all $k \geq m$
- $\mathcal{I}, (r, m) \models K_i\phi$ if for all points (r', m') of $\mathcal{I}$ such that $(r, m) \sim_i (r', m')$, we have $\mathcal{I}, (r', m') \models \phi$
- $\mathcal{I}, (r, m) \models C_G\phi$ if for all sequences of points $(r, m) = (r_0, m_0), (r_1, m_1), \ldots (r_k, m_k)$ of $\mathcal{I}$, such that for each $j = 0 \ldots k - 1$, there exists $i \in G$ such that $(r_j, m_j) \sim_i (r_{j+1}, m_{j+1})$, we have $\mathcal{I}, (r_k, m) \models \phi$.

For the knowledge operators, this semantics is essentially the same as the usual interpreted systems semantics. For the temporal operators, it corresponds to a semantics for branching time known as the *bundle semantics* [2, 21]. To specialise this general semantics to the clock semantics, we suppose that we have have for each agent i an observation function $O_i : S \to O$, for some set O, such that $O_i(s)$ represents agent i's observation in state s. Say that the equivalence relations $\sim_i$ in the system are *derived from the observation functions* O_i when $(r, m) \sim_i (r', m')$ iff $m = m'$

and $O_i(r(m)) = O_i(r'(m'))$. A system is a *clock system* if there exist observation functions from which its indistinguishability relations $\sim_i$ are derived. Intuitively, in such systems, an agent's knowledge is determined from its current observation plus the clock value.

For *model checking* we require the decidability of the problem of checking that a formula holds in a model. In order to do so, we require a finite state representation for the model. Interpreted systems are unsuitable for this, since they are based on infinite runs. We therefore treat interpreted systems as *generated* from an alternate finite representation. Define a (finite) *model* to be a tuple $M = (S, I, \Rightarrow, O, \pi)$ where S is a (finite) set of states, $I \subseteq S$ is the set of initial states, $\Rightarrow \subseteq S \times S$ is a serial temporal transition relation, $O = \{O_i\}_{i \in Ags}$ is a family of observation functions $O_i : S \to O$, and $\pi : S \to \mathcal{P}(Prop)$ is a propositional interpretation. We write $\mathcal{E}$ for the set of all finite models

Given a model M with states S, we may construct clock system $\mathcal{I}(M) = (\mathcal{R}, \sim_1, \ldots, \sim_n, \pi)$ over global states S, as follows. The component π in $\mathcal{I}(M)$ is identical to that in M. The set of runs is defined as follows. We say that a *fullpath* from a state s is an infinite sequence of states $s_0 s_1 ...$ such that $s_0 = s$ and $s_i \Rightarrow s_{i+1}$ for all $i \geq 0$. We use $Path(s)$ to denote the set of all fullpaths from state s. A *run* of the system is a fullpath $s_0 s_1 \ldots$ with $s_0 \in I$. We define $\mathcal{R}$ to be the set of runs of M. Finally, we take the indistinguishability relations $\sim_i$ to be the relations derived from the observation functions O_i in the environment.

A formula ϕ is said to *hold* in a model M, written $M \models \phi$, if $\mathcal{I}(M), (r, 0) \models \phi$ for all $r \in \mathcal{R}$. The model checking problem we study is defined as follows: given a finite model M and a formula ϕ, determine if $M \models \phi$. We are interested in this problem for a range of different languages $\mathcal{L}$, and as a function of its parameters as well as in general. More precisely, the *combined complexity* of model checking $\mathcal{L}$ is the complexity of the set $\{(M, \phi) \in \mathcal{E} \times \mathcal{L} \mid M \models \phi\}$. The *model complexity* of a fixed formula ϕ is the complexity of the set $\{M \in \mathcal{E} \mid M \models \phi\}$. This gives a measure of the complexity of model checking as a function of the size of the model. The *formula complexity* of $\mathcal{L}$ for a fixed model M is the complexity of the set $\{\phi \in \mathcal{L} \mid M \models \phi\}$. This captures the contribution to the complexity of model checking that derives from the formula.

3 Complexity

We now consider the complexity of the model checking problems introduced in the previous section, for the language $\mathrm{CTL}^*\mathrm{K}_n^{\mathtt{clk}}$and several of its sublanguages. For purposes of comparison, we first recall a number of known results for model checking temporal and epistemic logics.

Previous Results

Our logics build on the temporal logics LTL, CTL and CTL*, whose model checking complexities are already well understood. In the case of CTL*, the combined complexity is known to be PSPACE-complete [3, 4]. Indeed, this is already the case for the linear time logic LTL [1]. On the other hand, the logic CTL has a combined complexity in PTIME [3, 13]. A fortiori, the model and formula complexities of CTL are also in PTIME. For both LTL and CTL*, the model complexity is PTIME [22, 8, 13], and the formula complexity is PSPACE-complete [13]. Thus, the complexity of model checking linear time temporal logic derives primarily from the contribution made by the formula. Since the formula we wish to check is typically small (but the model may be large) this result explains the feasibility in practice of LTL model checking even in the face of its PSPACE-hardness in the sense of combined complexity.

Given these results, we easily obtain that the combined complexity and the formula complexity of $\mathrm{LTLK}_n^{\mathtt{clk}}$ are at least PSPACE hard. It is shown in [5] that, in fact, the addition of epistemic operators (interpreted with respect to clock semantics) does not necessarily add to the complexity of LTL. Both the combined and formula complexities of $\mathrm{LTLK}_n^{\mathtt{clk}}$ turn out to be in PSPACE (hence PSPACE-complete, since the fragment LTL is already PSPACE-hard). A difference is found in the case of model complexity, however. It is shown that whereas LTL has PTIME model complexity, for each level Π_k^p of the polynomial hierarchy, there exists a formula $\phi \in \mathrm{LTLK}_n^{\mathtt{clk}}$ such that $\{M \in \mathcal{E} \mid M \models \phi\}$ is Π_k^p-hard. An exact characterization of the model complexity of $\mathrm{LTLK}_n^{\mathtt{clk}}$ remains an open problem. However, it is possibly a very hard problem: note that whereas QBF is PSPACE complete, and the QBF formulas starting with $\forall$ and having k-alternations correspond to Π_k^p, it is not known whether the polynomial hierarchy is equal to PSPACE. It appears that the gap between the upper and lower bounds for model complexity of $\mathrm{LTLK}_n^{\mathtt{clk}}$ may be related to this problem.

CTL*$\mathrm{K}_n^{\texttt{clk}}$

We now show that we can derive complexity bounds for CTL*$\mathrm{K}_n^{\texttt{clk}}$ from the known results for LTL$\mathrm{K}_n^{\texttt{clk}}$. Given a model M and a CTL*$\mathrm{K}_n^{\texttt{clk}}$ formula ϕ, we will show that the model checking problem $M \models \phi$ is equivalent to another model checking problem $M' \models \phi'$, where ϕ' is a LTL$\mathrm{K}_{n+1}^{\texttt{clk}}$ formula and M' and ϕ' are of polynomial size w.r.t. M and ϕ, respectively. With this equivalence, we can move all the complexity results for LTL$\mathrm{K}_n^{\texttt{clk}}$ to CTL*$\mathrm{K}_n^{\texttt{clk}}$. To obtain model M', we add an agent $\top$ to the model M and define its observations by $O_\top(s) = s$ for each state $s \in S$. We take ϕ' to be the formula obtained from ϕ by replacing all A operators with $K_\top$ and all E operators with $\neg K_\top \neg$. The correctness of the equivalence claim now follows by an induction using the argument in the following result:

Lemma 1 *For all points (r, m) of $\mathcal{I}(M)$, and $\phi \in$CTL*$\mathrm{K}_n^{\texttt{clk}}$ we have $\mathcal{I}(M), (r, m) \models A\phi$ iff $\mathcal{I}(M'), (r, m) \models K_\top\phi$.*

Proof: The direction from right to left is trivial. It is easily seen that the the semantics of $\mathcal{I}(M), (r, m) \models A\phi$ refers to a subset of the points (r', m') referred to by the semantics of $\mathcal{I}(M), (r, m) \models K_\top\phi$. (Note that in both cases the fact that we are using clock semantics means that $m = m'$.) Conversely, suppose that $\mathcal{I}(M), (r, m) \models A\phi$. Considering $\mathcal{I}(M'), (r, m) \models K_\top\phi$, let (r', m') be a point such that $(r, m) \sim_\top (r', m')$. Then $m = m'$ and $r(m) = O_\top(r(m)) = O_\top(r'(m')) = r'(m') = r'(m)$. It follows that the sequence $r'' = r(0)r(1)\dots r(m)r'[m + 1..\infty]$ is a run. Since the operators in ϕ refer only to the future, an easy induction shows that $\mathcal{I}(M'), (r'', m) \models \phi$ iff $\mathcal{I}(M'), (r', m) \models \phi$. Note that r'' is equivalent to time m to r. Hence, by assumption that $\mathcal{I}(M), (r, m) \models A\phi$, we obtain that $\mathcal{I}(M), (r'', m) \models \phi$, hence $\mathcal{I}(M'), (r'', m) \models \phi$, from which it follows using the above observation that $\mathcal{I}(M'), (r', m) \models \phi$. This completes the proof that $\mathcal{I}(M'), (r, m) \models K_\top\phi$. □

This transformation immediately enables us to derive upper bounds of PSPACE for the combined, model and formula complexity of CTL*$\mathrm{K}_n^{\texttt{clk}}$. Since LTL$\mathrm{K}_n^{\texttt{clk}}$ is a sublanguage of CTL*$\mathrm{K}_n^{\texttt{clk}}$, lower bounds of PSPACE, Π_k^p (for any k, by some formula) and PSPACE respectively also follow directly.

CTL$\mathrm{K}_n^{\texttt{clk}}$

The combined complexity and model complexity upper bounds for CTL*$\mathrm{K}_n^{\texttt{clk}}$ are also upper bounds for CTL$\mathrm{K}_n^{\texttt{clk}}$, because CTL$\mathrm{K}_n^{\texttt{clk}}$ is a sublanguage of CTL*$\mathrm{K}_n^{\texttt{clk}}$.

The logic CTL has formula complexity of LOGSPACE [13], which is lower than the PSPACE-complete formula complexity of LTL, as noted above. We now show that this difference remains reflected in the extended logic CTL$\mathrm{K}_n^{\texttt{clk}}$. We present an algorithm that shows that this logic has LOGSPACE formula complexity, lower than the PSPACE-complete formula complexity for the extended logic CTL*$\mathrm{K}_n^{\texttt{clk}}$and matching the LOGSPACE formula complexity for CTL model checking.

Given a fixed model $M = (S, I, \Rightarrow, \sim_1, ..., \sim_n, \pi)$, we construct another model $M' = (S \times \mathcal{P}(S), I', \Rightarrow', \sim_1', ..., \sim_n', \pi')$, with states of the form (s, P) where $s \in S$ and $P \subseteq S$, such that

1. $(s, P) \in I'$ iff $s \in I$ and $P = I$,
2. $(s, P) \Rightarrow' (t, Q)$ iff $s \Rightarrow t$ and $Q = \{t \in S \mid \exists s \in P(s \Rightarrow t)\}$,
3. $(s, P) \sim_i' (t, Q)$ iff $P = Q$ and $s \sim_i t$,
4. $\pi'(s, P) = \pi(s)$.

Though M' might be exponentially larger than M, it is also a fixed structure. Now $M \models \phi$ is equivalent to $M' \models \phi$ for observational semantics defined as follows.

1. $M', (s, P) \models p$ iff $p \in \pi'(s, P)$.
2. $M', (s, P) \models \neg\phi$ iff not $M', (s, P) \models \phi$.
3. $M', (s, P) \models EX\phi$ iff there exists (t, Q) such that $M', (t, Q) \models \phi$ and $(s, P) \Rightarrow' (t, Q)$.

4. $M', (s,P) \models E[\phi_1 U \phi_2]$ iff there exists a path $(s,P) = (s_0,P_0) \Rightarrow' (s_1,P_1) \Rightarrow' \ldots \Rightarrow' (s_m,P_m)$ such that $M', (s_m,P_m) \models \phi_2$ and $M', (s_i,P_i) \models \phi_1$ for $0 \leq i \leq m-1$.

5. $M', (s,P) \models EG\phi$ iff there exists an infinite path $(s,P) = (s_0,P_0) \Rightarrow' (s_1,P_1) \Rightarrow' \ldots$, such that $M', (s_i,P_i) \models \phi_1$ for all $i \geq 0$.

6. $M', (s,P) \models K_i\phi$ iff $M', (t,Q) \models \phi$ for all (t,Q) with $(s,P) \sim_i' (t,Q)$.

7. $M', (s,P) \models C_G\phi$ iff for all sequences $(s,P) = (s_0,P_0), (s_1,P_1), \ldots, (s_m,P_m)$ with $m \geq 0$ and $(s_i,P_i) \sim_j (s_{i+1},P_{i+1})$ for $j \in G$ and $0 \leq i \leq m-1$, we have $M', (s_m,P_m) \models \phi$.

By [13], for CTL, $M \models \phi$ can be checked in space $O(|M| \cdot log|\phi|)$. With respect to the observational semantics, the addition of epistemic operators to CTL does not increase the complexity of model checking, since we may easily reduce the epistemic transitions to a special type of temporal transition. Therefore, the formula complexity of $\mathrm{CTLK}_n^{\mathtt{clk}}$ is in LOGSPACE.

Lower bounds for model checking $\mathrm{CTLK}_n^{\mathtt{clk}}$ can be obtained from the proof of lower bounds for $\mathrm{LTLK}_n^{\mathtt{clk}}$ in [5] by noting that the proofs of these lower bounds use structures of the following special form.

Definition 1 *Let $M = (S, I, \Rightarrow, \sim_1, \ldots, \sim_n, \pi)$ be a model. Say that M is a* lasso-bundle *if for any $s \in S$, there exists a unique state s' such that $s \Rightarrow s'$. M is a* lasso-structure *if M is a* lasso-bundle *and I is a singleton set.*

For lasso-structures, the semantics of LTL and CTL are known to coincide [9, 17]. An LTL formula can be evaluated in a lasso-bundled model by a CTL model checker by prefixing each temporal operator by an A/E path quantifier which results in a CTL formula. The reverse also holds.

We show that a similar result applies to $\mathrm{LTLK}_n^{\mathtt{clk}}$ and $\mathrm{CTLK}_n^{\mathtt{clk}}$. Define a transformation function f mapping a normalized $\mathrm{LTLK}_n^{\mathtt{clk}}$ formula (in which all negative operators occur in front of atomic propositions) into a $\mathrm{CTLK}_n^{\mathtt{clk}}$ formula.

1. $f(p) = p$
2. $f(\neg\phi) = \neg f(\phi)$
3. $f(\phi_1 \vee \phi_2) = f(\phi_1) \vee f(\phi_2)$
4. $f(X\phi) = AXf(\phi)$, $f(F\phi) = AFf(\phi)$, $f(G\phi) = AGf(\phi)$
5. $f(\phi_1 U \phi_2) = A[f(\phi_1) U f(\phi_2)]$
6. $f(Y\phi) = Yf(\phi)$, where $Y \in \{K_i, C_G\}$.

The transformation prefixes each temporal operator with a universal operator A. The following proposition concludes that this transformation preserves the satisfiability of formulae under lasso-bundled models.

Proposition 1 *For any lasso-bundled model M and any $LTLK_n^{\mathtt{clk}}$ formula ϕ, and run r of $\mathcal{I}(M)$, we have $\mathcal{I}(M), (r,n) \models \phi \Leftrightarrow \mathcal{I}(M), (r,m) \models f(\phi)$.*

Proof: By a straightforward induction on the construction of ϕ. Note that it follows from the fact that M is lasso-bundled that if r' is a run equivalent to r up to time n then in fact $r' = r$. This implies that $\mathcal{I}(M), (r,m) \models \phi$ iff $\mathcal{I}(M), (r,m) \models A\phi$ iff $\mathcal{I}(M), (r,m) \models E\phi$. □

The lower bounds of $\mathrm{LTLK}_n^{\mathtt{clk}}$ logic are proved in [5] by a reduction from the satisfiability problem of QBF. For any alternation depth k, a $\mathrm{LTLK}_n^{\mathtt{clk}}$ formula ϕ_k is constructed, such that for each QBF formula Ψ of alternation depth k, a lasso-bundled model M_Ψ can be constructed, such that Ψ is satisfiable iff $\mathcal{I}(M_\Psi) \models \phi_k$. Now by Proposition 1, ϕ_k can be further reduced to a $\mathrm{CTLK}_n^{\mathtt{clk}}$ formula $f(\phi_k)$, and Ψ is satisfiable iff $\mathcal{I}(M_\Psi) \models f(\phi_k)$. Therefore, we can move the lower bounds of combined complexity and model complexity for $\mathrm{LTLK}_n^{\mathtt{clk}}$ to $\mathrm{CTLK}_n^{\mathtt{clk}}$. In particular, we conclude that the combined complexity is PSPACE-hard, and for each k there exists a formula whose model complexity is Π_k^p-hard. The lower bound for formula complexity can be deemed as LOGSPACE, since every problem in LOGSPACE is complete under log-space reductions.

CTL$^-$K$_n^{\mathtt{clk}}$

We reduce the problem of model checking CTL$^-$K$_n^{\mathtt{clk}}$ to a fragment of Presburger Arithmetic, called extended Min-Max Arithmetic (MMA) in [7]. We use the notation $[i, j] = \{i, i+1, ..., j\}$ and $[j] = [1, j]$. An extended MMA dag-formula α based in a finite partial order $(X, \leq)$ is a collection of definitions of variables $(\alpha_i)_{i\in X}$, where the definition for each α_i is one of the following, with $i > j, k$ and $\sim \in \{\leq, \geq\}$ and $m, n \in \mathbb{N}$ in each case: (1) $\equiv m \mod n$ with $n > 0$ and $m \in \mathbb{Z}/n\mathbb{Z}$, (2) $\sim n$, (3) $\neg\alpha_j$, (4) $\alpha_j \wedge \alpha_k$, (5) $\sim \max(\alpha_j, n)$, (6) $m \sim \max(\alpha_j, n)$, (7) $\alpha_j \odot a$, with $\odot \in \{+, -\}$ and $a \in \mathbb{N}$ written in *unary*. The semantics of α_i in the context of an extended MMA formula α, written as $[\![\alpha_i]\!]$, is the set of natural numbers satisfying the definition of α_i, e.g., $[\![\sim \max(\alpha_j, n)]\!] = \{k \in \mathbb{N} | k \sim \max([\![\alpha_j]\!] \cap [0, n])\}$, and $[\![m \sim \max(\alpha_j, n)]\!]$ is either $\mathbb{N}$ or $\emptyset$ depending on whether $m \sim \max [\![\alpha_j]\!] \cap [0, n])$, and $[\![\alpha_j \odot a]\!] = \{k \mid k = n \odot a \geq 0,\ n \in [\![\alpha_j]\!]\}$. The following two lemmas are from [7].

Lemma 2 *Given an extended MMA dag-formula $\alpha = (\alpha_i)_{i\in X}$, there exist numbers L_i and T_i, computable in PTIME, such that, if $n_1, n_2 > T_i$ and $n_1 \equiv n_2 \mod L_i$ then $n_1 \in [\![\alpha_i]\!] \Leftrightarrow n_2 \in [\![\alpha_i]\!]$.*

Lemma 3 *Given $n \in \mathbb{N}$ and an extended MMA dag-formula α, whether $n \in [\![\alpha_i]\!]$ can be decided with complexity $\triangle_2^P = P^{NP}$.*

Another result we use is Chrobak's normal form for nondeterministic finite automata [18]:

Lemma 4 *Given a model M, we may compute in polynomial time a collection of arithmetic progressions $R(s, t) = \{a_i + b_i\mathbb{N} \mid i = 1..k\}$ such that $\cup R(s, t) = \{n \in \mathbb{N} \mid s \Rightarrow^n t\}$, $|R(s, t)| \leq |S|^2$ and $a_i = O(|S|^2)$ and $b_i = O(|S|)$.*

We write $M(\psi)$ for $\{(s, n) \in S \times \mathbb{N} \mid \exists r(\mathcal{I}(M), (r, n) \models \psi$ and $r(n) = s)\}$. (Note that $\mathcal{I}(M), (r, n) \models \psi$ and $r(n) = s$ implies $\mathcal{I}(M), (r', n) \models \psi$ for all runs r' with $r'(n) = s$.) Let $subf(\phi)$ be the set of subformulas of ϕ, and let $\mathbb{T}$ denote tautology.

Lemma 5 *Given a formula $\phi \in CTL^-K_n^{\mathtt{clk}}$ and a model M, let $X = S \times (subf(\phi) \cup \{\mathbb{T}\})$, equipped with the partial order defined by $(s_1, \psi_1) \geq (s_2, \psi_2)$ iff $s_1 = s_2$ and $\psi_2 \in subf(\psi_1) \cup \{\mathbb{T}\}$. There exists an extended MMA-dag $(\alpha_x)_{x\in X}$, computable in time polynomial in $|M| + |\phi|$ such that for all $(s, \psi) \in X$, we have $[\![\alpha_{(s,\psi)}]\!] = \{n \in N \mid (s, n) \in M(\psi)\}$.*

Proof: We define $\alpha_{(s,\psi)}$ by induction on $\geq$. First, note that by Lemma 4, for each pair $s, t \in S$, we may write the set $\{n \mid s \Rightarrow^n t\}$ as a polynomial size union $R(s, t)$ of arithmetic progressions $a + b\mathbb{N}$. Thus, we may take

$$\alpha_{(t,\mathbb{T})} = \bigwedge_{s\in I} \bigwedge_{a+b\mathbb{N}\in R(s,t)} \equiv a \bmod b$$

for the base case. In the inductive case, we proceed as follows:

1. $\psi = p$: here we take $\alpha_{(s,p)} = \alpha_{(s,\mathbb{T})}$ if $p \in \pi(s)$, else we take $\alpha_{(s,p)} = \mathbb{F}$ (where $\mathbb{F} = (> 1) \wedge (< 1)$ so that $[\![\mathbb{F}]\!] = \emptyset$).
2. $\psi = \psi_1 \wedge \psi_2$: here $\alpha_{(s,\psi)} = \alpha_{(s,\psi_1)} \wedge \alpha_{(s,\psi_2)}$
3. $\psi = \neg\psi_1$: here $\alpha_{(s,\psi)} = \alpha_{(s,\mathbb{T})} \wedge \neg\alpha_{(s,\psi_1)}$
4. $\psi = K_i\psi_1$: here
$$\alpha_{(s,\psi)} = \alpha_{(s,\mathbb{T})} \wedge \bigwedge_{t\in S, s\sim_i t} (\alpha_{(t,\mathbb{T})} \Rightarrow \alpha_{(t,\psi_1)})$$
5. $\psi = C_G\psi_1$: here we take
$$\alpha_{(s,\psi)} = \alpha_{(s,\mathbb{T})} \wedge \bigwedge_{t\in S} (\gamma(s, t, G, |S|) \Rightarrow \alpha_{(t,\psi_1)})$$
where $\gamma(s, t, G, k)$ is an extended MMA-dag formula expressing times at which t is reachable from s in k steps through the relation $\cup_{i\in G} \sim_i$. This can be represented efficiently by the following induction: in the base cases, $\gamma(s, t, G, 0) = \alpha_{(s,\mathbb{T})}$ if $s = t$ and $\mathbb{F}$ otherwise, $\gamma(s, t, G, 1) = \alpha_{(s,\mathbb{T})} \wedge \alpha_{(t,\mathbb{T})}$ if $s \sim_i t$ for some $i \in G$, otherwise $\gamma(s, t, G, 1) = \mathbb{F}$. Inductively, for $k \geq 2$, we define
$$\gamma(s, t, G, k) = \bigvee_{s'\in S} (\gamma(s, s', G, \lceil k/2 \rceil) \wedge \gamma(s', t, G, \lfloor k/2 \rfloor)).$$

6. $\psi = EX\psi_1$: here $\alpha_{(s,\psi)} = \alpha_{(s,\mathbb{T})} \wedge \bigvee_{s \Rightarrow t}(\alpha_{(t,\psi_1)} - 1)$.

7. $\psi = EF\psi_1$: we break this case down into several possibilities. Suppose that $(s, n) \in M(EF\psi_1)$. Then there exist $t \in S$ and $n' \in \mathbb{N}$ such that $s \Rightarrow^{n'-n} t$ and $(t, n') \in M(\psi_1)$.

   It follows using Lemma 4 that there exists $a + b\mathbb{N} \in R(s, t)$ such that $n' - n \in a + b\mathbb{N}$. If $b = 0$, then plainly we have $n \in \alpha_{(t,\psi_1)} - a$.

   Consider the case where $b \neq 0$, so $n' - n \equiv a \ mod\ b$. Moreover, inductively, there exist, by Lemma 2, numbers T and L such that $n_1, n_2 > T$ and $n_1 \equiv n_2 \mod L$ implies $n_1 \in [\![\alpha_{(t,\psi_1)}]\!]$ iff $n_2 \in [\![\alpha_{(t,\psi_1)}]\!]$. We consider the different possible values of c, the residue of $n' \ mod\ b$. There are two possibilities for a given c: either there exist an infinite number of $n' \equiv c \ mod\ b$ such that $(n', t) \in M(\psi_1)$, or there exist a finite number such values.

   We first note that in the case of an infinite number of such values, there must exist such a value in the range $T < n' \leq T + L$, by the periodicity condition on $\alpha_{(t,\psi_1)}$. Writing $\beta(t, \psi_i) = \alpha_{(t,\psi_1)} \wedge (\equiv c \ mod(gcd(b, L)))$, and $inf(t, \psi_1, b, c) = T < max(\beta(t, \psi_i), t + L)$, we find that $inf(t, \psi_1, b, c)$ expresses that there exist an infinite number of values of $n' \equiv c \ mod\ b$ such that $(n', t) \in M(\psi_1)$. (Note that if $(m, t) \in M(\psi_1)$ and $m \equiv c \ mod(gcd(b, L))$, then by the Chinese Remainder theorem, the simultaneous equation $x \equiv c \ mod\ b$ and $x \equiv m \ mod\ L$ has a solution, which may be taken to be greater than T, so that by periodicity of $\alpha_{(t,\psi_1)}$ we have $(x, t) \in M(\psi_1)$ and $x \equiv c \mod b$.) Thus, we may represent the case of n with an infinite number of $n' \equiv c \mod b$ using the formula

$$\chi_\infty(t, \psi_1, b, c) = inf(t, \psi_1, b, c) \wedge (\equiv c - a \ mod\ b) .$$

   In the case of a finite number of such values, note that if $[\![inf(t, \psi_1, b, c)]\!] = \emptyset$ then for all $n' \in [\![\alpha_{(t,\psi_1)}]\!]$ with $n' \equiv c \ mod\ b$ we have $n' < T$. (If there exists such a value n' larger than T then by L-periodicity we can find one in the range $[T, T + L]$ with $n' \equiv c \ mod\ gcd(b, L)$.) Thus, in this case we can express the possible values of n by the expression

$$\chi_{<\infty}(t, \psi_1, b, c) = \neg inf(t, \psi_1, b, c) \wedge \leq max((\alpha_{(t,\psi_1)} \equiv c \ mod\ b), t) .$$

   Putting the pieces together, we may define $\alpha_{(s,\psi)}$ to be

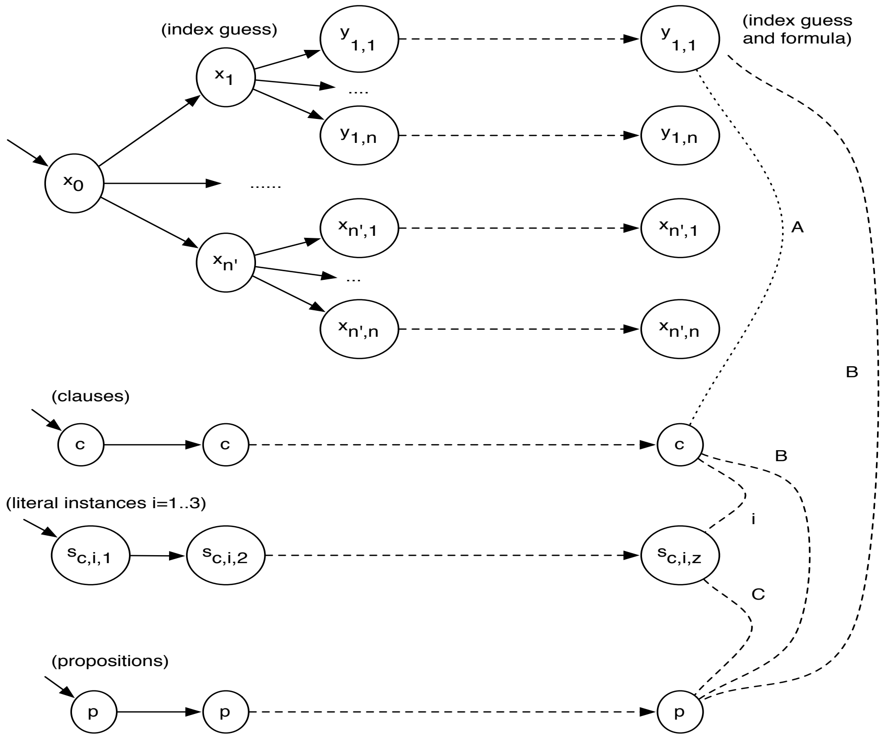
$$\bigvee_{t \in S} \left(\begin{array}{l} (\bigvee_{a+b\mathbb{N} \in R(s,t),\ b=0} \quad \alpha_{(t,\psi_1)} - a) \\ \vee \bigvee_{a+b\mathbb{N} \in R(s,t), 0 \leq c < b} \quad \chi_\infty(t, \psi_1, b, c) \vee \chi_{<\infty}(t, \psi_1, b, c) \end{array} \right) .$$

□

Combining this result with Lemma 3, we obtain the upper bound of P^{NP} for the combined complexity of model checking CTL$^-$K$_n^{\texttt{clk}}$.

By way of lower bound, we show that the model complexity (hence also the combined complexity) of model checking CTL$^-$K$_n^{\texttt{clk}}$ is hard for $P^{NP[log]}$, i.e., the complexity class corresponding to PTIME computations with a logarithmic number of queries to an NP oracle.

(The same lower bound also holds for model complexity, by a slightly more elaborate proof that we leave for a longer version of this paper.) This is equivalent to the class of PTIME computations with *independent* queries to an NP oracle, i.e., the construction of an oracle query may not depend on the outcome of any other oracle query. The following problem ODD-SAT is known to be complete for this class: given boolean formulas $\phi_1, \ldots \phi_n$, determine if there exists an odd i such that $\phi_1, \ldots \phi_i$ are satisfiable and $\phi_{i+1}, \ldots \phi_n$ are not satisfiable.



Theorem 1 *Model complexity for $CTL^{-}K_n^{\mathtt{clk}}$ is $P^{NP[log]}$ hard.*

Proof: By encoding of the problem INDEX-ODD: given a list $\mathbf{F} = F_1, \ldots, F_n$ of boolean formulas in 3-CNF, does there exist an odd index k such that $F_1, \ldots, F_k$ are all satisfiable and $F_{k+1}, \ldots, F_n$ are all unsatisfiable? This problem is $P^{NP[log]}$-complete [7]. Without loss of generality, we may assume that the F_i are given as a set of monotone 3-clauses, and divide each into the set F_i^+ of positive clauses and F_i^- of negative clauses. We may also assume without loss of generality that distinct formulas do not share propositional constants. Hence, they also do not share any clauses. We write F for the union of the F_i, F^+ for the union of the F_i^+ and F^- for the union of the F_i^-.

We construct a system $M_{\mathbf{F}}$ with agents $A, B, C, 1, 2, 3$. A depiction of the structure of the runs of the system is given in Figure 3. The system has atomic propositions *start*, *clause*, *pos*, *propn*, t, *val* and the following states:

1. a state x_0, taken to be initial, and satisfying the proposition *start*. No other state satisfies *start*. This state has temporal successors $x_1, x_3, \ldots x_{n'}$, where n' is the largest odd number less than or equal to n. Further, each state x_i has temporal successors $y_{i,j}$ where $1 \leq j \leq n$. Intuitively, $y_{i,j}$ represents that we are checking satisfaction of formula j and guessing that i is the maximal odd index for which $F_1, \ldots, F_i$ are satisfiable and $F_{i+1}, \ldots, F_n$ are unsatisfiable. Accordingly, we take $y_{i,j} \models sat$ iff $j \leq i$.

2. For each clause $c \in F$ a state c, such that $c \models clause$. No other states satisfy *clause*. Further, $c \models pos$ iff $c \in F^+$. Values of *pos* on other states are irrelevant. We take state c to be initial, and the only transition from c to be $c \rightarrow c$.

3. For each atomic proposition p occurring in F, a state p, such that $p \models propn$. No other states satisfy *propn*. We take state p to be initial, and the only transition from p to be $p \rightarrow p$.

4. For each clause $c \in F$ and $i = 1..3$, choose a distinct prime $q_{c,i}$. For each $k = 1 \ldots q_{c,i}$, we have a state $s_{c,i,k}$. We take $s_{c,i,1}$ to be initial, and the only transitions involving these states are $s_{c,i,1} \rightarrow s_{c,i,2} \rightarrow \ldots s_{c,i,q_{c,i}} \rightarrow s_{c,i,1}$. All these states satisfy *val*, and no other state satisfies *val*. We let $s_{c,i,k} \models t$ iff $k = 1$.

Note that since the first k primes can be found in $2 \ldots, k^2$, the model $M_{\mathbf{F}}$ has $O(|\mathbf{F}|^2)$ states.

Intuitively, the set of states $M_{\mathbf{F}}(N)$ occurring at a particular time N represents an assignment of truth values to each of the instances of an atomic proposition occurring in a clause. Note that for each clause c and position $i = 1...3$, exactly one state of the form $s_{c,i,k}$ occurs in $M(N)$. We take the assignment of the proposition instance to be true if t holds at that state. Since the cycle lengths on the cycles on the $s_{c,i,k}$ are co-prime, all selections of one state from each cycle (hence all truth assignments to each proposition instance) are attained at some time N.

Note that assignments defined in this way allow for a proposition to be assigned to be true in one clause and false in another. We use a formula to enforce consistency of the assignments. Let the observations for agent C be defined so that for atomic proposition p, we have $p \sim_C s_{c,i,k}$ iff p is the proposition in literal i of clause c. Let agent B be unable to distinguish any states. Then we may express that the assignments made to the propositions are consistent at all instances by the formula

$$consis = K_B(propn \Rightarrow (K_C(val \Rightarrow t) \vee K_C(val \Rightarrow \neg t)$$

which expresses that for each proposition, either all its occurrences in a clause are assigned true or they are all assigned false.

To check for satisfaction of the clauses, we introduce agents $i = 1, 2, 3$ corresponding to literal positions within the clause. We define the equivalence relations for these agents to be the smallest equivalence relations such that $c \sim_i s_{c,i,k}$ for all $c \in F$, $i = 1..3$ and $k = 1 \ldots q_{c,i}$. Note that clause c always occurs in $M_{\mathbf{F}}(N)$, together with, for each $i = 1..3$, at most one state of the form $s_{c,i,k}$. Thus, clause c is represented as being satisfied in $M_{\mathbf{F}}$ at time N if the formula *satisfied* defined as

$$\begin{aligned} \textit{satisfied} = \quad & (pos \wedge \overline{K}_1(val \wedge t) \wedge \overline{K}_2(val \wedge t) \wedge \overline{K}_3(val \wedge t)) \;\vee \\ & (\neg pos \wedge \overline{K}_1(val \wedge \neg t) \wedge \overline{K}_2(val \wedge \neg t) \wedge \overline{K}_3(val \wedge \neg t)). \end{aligned}$$

holds at a point at time N where the state is c.

To express that all clauses are satisfied, we define the equivalence relation $\sim_A$ to be the smallest equivalence relation such that $y_{i,j} \sim_A c$ for all $i = 1 \ldots n'$, $j = 1, \ldots n$ and $c \in F_j$. That is, $\sim_A$ connects states $y_{i,j}$ at which we are checking formula F_j with the clauses c in F_j. Thus we may express that all clauses in formula F_j are satisfied by the formula

$$K_A(clause \rightarrow satisfied),$$

evaluated at a point at time N where the state is $y_{i,j}$

To expresss that there is a (consistent) satisfying assignment for F_j, evaluating at a state $y_{i,j}$, we may use formula

$$EF(consis \wedge K_A(clause \rightarrow \textit{satisfied})),$$

evaluated at a point at time 3 where the state is $y_{i,j}$. Note that the EF operator takes us to the same state $y_{i,j}$ at a different time$N$ (hence a different guess for the consistent assignment to the propositions, and the K_A operator then checks that all clauses in F_j are satisfied.

Finally, to express that for some odd k, exactly the first k of the formulas $F_1, \ldots, F_n$ is satisfiable, we may use the formula

$$start \Rightarrow EX\,AX(sat \Leftrightarrow EF(consis \wedge K_A(clause \rightarrow satisfied))$$

Intuitively, the first EX guesses an odd index i, and the AX then checks that all formulas up to index i are satisfiable and the remainder are unsatisfiable. This formula is valid in $M_{\mathbf{F}}$ iff $\mathbf{F} \in INDEX - ODD$. □

This lower bound does not quite match the upper bound. We do not know at present whether either can be improved.

4 Related Work

The model checking complexities for temporal logics are collected in Table 2, which is taken from [13].

When combined with epistemic logic under the observational semantics, the model checking complexities do not increase, e.g., for CTLK$_n^{obs}$, PTIME-completeness is shown in [6] and [10], for upper bound and lower bound, respectively.

Logic	Combined Complexity	Model Complexity	Formula Complexity
LTL	PSPACE-complete	NLOGSPACE-complete	PSPACE-complete
CTL*	PSPACE-complete	NLOGSPACE-complete	PSPACE-complete
CTL	PTIME-complete	NLOGSPACE-complete	LOGSPACE
CTL⁻	PTIME-complete	NLOGSPACE-complete	LOGSPACE

Table 2: Complexity Results for Temporal Logics

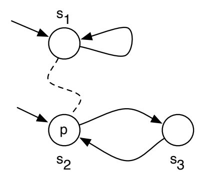


Figure 1: A model

When combined with epistemic logic under the perfect recall semantics, many subtleties are introduced. Linear time temporal epistemic logics on synchronous systems or asynchronous systems are analyzed in [19] and [20], and [16, 15, 14] explore branching time temporal epistemic logics.

For epistemic logic under clock semantics, [5] provides complexity results for its combination with LTL. The present paper complements this work by considering instead several branching time logics. A bounded model checking (BMC) algorithm for the universal fragment of $\mathrm{CTL}^*\mathrm{K}_n^{\mathtt{clk}}$ logic, abbreviated as $\mathrm{ACTL}^*\mathrm{K}_n^{\mathtt{clk}}$, is presented in [11, 12]. However, it appears to be incorrect. A flaw of their algorithm occurs on formulas of the form $AFK_i\phi$, which in negated dual form $EG\neg K_i\neg\psi$ requires finding a witness run on which $\neg K_i\neg\psi$ holds at all times. For bounded model checking temporal logics, runs can be represented as lassos $s_0 \Rightarrow s_1 \Rightarrow \ldots s_k \Rightarrow s_l$ where $0 \leq l \leq k$ and $k \leq |S|$. To satisfy the formula $EG\neg K_i\neg\psi$ would then require that we satisfy $\neg K_i\neg\psi$ at each point in the run. For the observational semantics for knowledge, this can be done by finding for each time m with $0 \leq m \leq k$ another lasso $t_0 \Rightarrow t_1 \Rightarrow \ldots t_{k'} \Rightarrow t_{l'}$ with $s_m \sim_i t_{m'}$ for some m' with ψ holding at the state $t_{m'}$. However, the clock semantics requires that the states s_m and $t_{m'}$ occur at the same time. The approach to this in [11, 12] is to check that the witness state $t_{m'}$ can occur at the same time as the state s_m.

However, this is not sufficient, rather, we require such a witness for *each* time n such that s_k can occur at time n in the original lasso. An example is shown in Figure 1. The model $M = (S, I, \Rightarrow, \sim_a, \pi)$, where $S = \{s_1, s_2, s_3\}$, $I = \{s_1, s_2\}$, $\Rightarrow$ is shown in Figure 1, $\sim_a$ is the least equivalence relation with $s_1 \sim_a s_2$ and $\pi(s_2) = \{p\}, \pi(s_1) = \pi(s_3) = \emptyset$. For the $\mathrm{ACTL}^*\mathrm{K}_n^{\mathtt{clk}}$ formula $\phi = AFK_a\neg p$, the algorithm of [11, 12] would conclude that $M \not\models \phi$ by finding a 'counterexample' lasso of length 1, where the dual $\neg\phi \equiv EG\neg K_a\neg p$ is resolved on loop $s_1 \Rightarrow s_1$, and the fact that $\neg K_a\neg p$ must hold at s_1 is witnessed by the fact that p holds at state $s_2 \sim s_1$ on the lasso $s_2 \Rightarrow s_3 \Rightarrow s_2$. However, though $\neg K_a\neg p$ is satisfiable on the lasso $s_1 \Rightarrow s_1$ at time 0, it is not satisfiable at time 1 since state s_2 is not possible at time 1.

By comparison, the algorithms discussed above for $\mathrm{CTLK}_n^{\mathtt{clk}}$ and $\mathrm{LTLK}_n^{\mathtt{clk}}$ use set-level loops. More specifically, the state-level lasso $s_1 \Rightarrow s_1$ should be lifted to a set-level lasso $(s_1, \{s_1, s_2\}) \Rightarrow (s_1, \{s_1, s_3\}) \Rightarrow (s_1, \{s_1, s_2\})$, where the second element in a pair is the set of possible states at a given time. Now $\neg K_a\neg p$ is not satisfiable on $(s_1, \{s_1, s_3\})$ because state s_2 is not possible at that time. Therefore, the loop $s_1 \Rightarrow s_1$ does not provide a valid counterexample for formula ϕ. Indeed, it can be easily checked that $M \models \phi$.

While we believe that lasso-like structures can be used to provide counter-examples for bounded model checking of $\mathrm{ACTL}^*\mathrm{K}_n^{\mathtt{clk}}$, the fact that the model complexity of $\mathrm{CTLK}_n^{\mathtt{clk}}$ is hard for the polynomial hierachy (compared with the NLOGSPACE-complete model complexities of LTL and CTL) strongly suggests that the claim in [11, 12] that lassos of length at most $|M|$ suffice cannot be upheld. Rather, the apparent necessity of set-level lassos suggests that the bound should be of the order of $2^{|M|}$. We leave a more detailed investigation of this issue for further work.

References

[1] A.P. Sistla and E.M. Clarke, 'The complexity of propositional linear temporal logics', *Journal of Assoc. Comput. Mach.*, **32**(3), 733–749, (1985).

[2] J. Burgess, 'Logic and time', *Journal of Symbolic Logic*, **44**, 556–582, (1979).

[3] Edmund M. Clarke, E. Allen Emerson, and A. Prasad Sistla, 'Automatic verification of finite-state concurrent systems using temporal logic specifications', *ACM Trans. Program. Lang. Syst.*, **8**(2), 244–263, (1986).

[4] E. Allen Emerson and Chin-Laung Lei, 'Modalities for model checking: Branching time logic strikes back', *Sci. Comput. Program.*, **8**(3), 275–306, (1987).

[5] Kai Engelhardt, Peter Gammie, and Ron van der Meyden, 'Model checking knowledge and linear time: PSPACE cases', in *LFCS*, eds., Sergei N. Artemov and Anil Nerode, volume 4514 of *LNCS*, pp. 195–211. Springer, (2007).

[6] Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi, *Reasoning about Knowledge*, MIT Press, 1995.

[7] Stefan Goller, Richard Mayr, and Anthony Widjaja To, 'On the computational complexity of verifying one-counter processes', in *LICS*, pp. 235–244. IEEE Computer Society, (2009).

[8] Kupferman, Vardi, and Wolper, 'An automata-theoretic approach to branching-time model checking', *JACM: Journal of the ACM*, **47**, (2000).

[9] Orna Kupferman and Moshe Y. Vardi, 'Model checking of safety properties', in *CAV*, eds., Nicolas Halbwachs and Doron Peled, volume 1633 of *LNCS*, pp. 172–183. Springer, (1999).

[10] Alessio Lomuscio and Franco Raimondi, 'The complexity of model checking concurrent programs against CTLK specifications', in *AAMAS*, eds., Hideyuki Nakashima, Michael P. Wellman, Gerhard Weiss, and Peter Stone, pp. 548–550. ACM, (2006).

[11] Xiangyu Luo, Kaile Su, Abdul Sattar, Qingliang Chen, and Guanfeng Lv, 'Bounded model checking knowledge and branching time in synchronous multi-agent systems', in *AAMAS*, eds., Frank Dignum, Virginia Dignum, Sven Koenig, Sarit Kraus, Munindar P. Singh, and Michael Wooldridge, pp. 1129–1130. ACM, (2005).

[12] Xiangyu Luo, Kaile Su, Abdul Sattar, and Mark Reynolds, 'Verification of multi-agent systems via bounded model checking', in *Australian Conference on Artificial Intelligence*, eds., Abdul Sattar and Byeong Ho Kang, volume 4304 of *LNCS*, pp. 69–78. Springer, (2006).

[13] Ph. Schnoebelen, 'The complexity of temporal logic model checking', in *Advances in Modal Logic*, eds., Philippe Balbiani, Nobu-Yuki Suzuki, Frank Wolter, and Michael Zakharyaschev, pp. 393–436. King's College Publications, (2002).

[14] Nikolay V. Shilov and Natalya Olegovna Garanina, 'Model checking knowledge and fixpoints', in *FICS*, eds., Zoltan Esik and Anna Ingolfsdottir, volume NS-02-2 of *BRICS Notes Series*, pp. 25–39. University of Aarhus, (2002).

[15] Nikolay V. Shilov and Natalya Olegovna Garanina, 'Well-structured model checking of multiagent systems', in *Ershov Memorial Conference*, eds., Irina Virbitskaite and Andrei Voronkov, volume 4378 of *LNCS*, pp. 363–376. Springer, (2006).

[16] Nikolay V. Shilov, Natalya Olegovna Garanina, and K.-M. Choe, 'Update and abstraction in model checking of knowledge and branching time', *Fundam. Inform.*, **72**(1-3), 347–361, (2006).

[17] Heikki Tauriainen and Keijo Heljanko, 'Testing LTL formula translation into büchi automata', *STTT*, **4**(1), 57–70, (2002).

[18] Anthony Widjaja To, 'Unary finite automata vs. arithmetic progressions', *Inf. Process. Lett.*, **109**(17), 1010–1014, (2009).

[19] Ron van der Meyden, 'Common knowledge and update in finite environments', *Inf. Comput.*, **140**(2), 115–157, (1998).

[20] Ron van der Meyden and Nikolay V. Shilov, 'Model checking knowledge and time in systems with perfect recall (extended abstract)', in *FSTTCS*, eds., C. Pandu Rangan, Venkatesh Raman, and Ramaswamy Ramanujam, volume 1738 of *LNCS*, pp. 432–445. Springer, (1999).

[21] Ron van der Meyden and Kashu Wong, 'Complete axiomatizations for reasoning about knowledge and branching time', *Studia Logica*, **75**(1), 93–123, (2003).

[22] Moshe Y. Vardi and Pierre Wolper, 'An automata-theoretic approach to automatic program verification (preliminary report)', in *LICS*, pp. 332–344. IEEE Computer Society, (1986).