ALGORITHMIC PROBLEMS IN LOGICS OF KNOWLEDGE AND TIME

CĂTĂLIN DIMA

Communicated by Marius Iosifescu

We present some recent results on the satisfiability and model-checking problems for combinations of temporal and epistemic logics.

AMS 2010 Subject Classification: 68Q60, 03B70, 03B42.

Key words: temporal epistemic logic, coalition logic, model-checking, satisfiability.

Epistemic logics are modal logics designed for modeling multi-agent systems in which agents have incomplete observation capacities of the system state (including the state of the other agents), but are endowed with the ability to make deductions about properties of the system state, based on their observations. One of the key features of epistemic logics is the possibility to specify and model not only system properties that are observable for an agent, but also properties like the knowledge that agent a has about the system state, or the knowledge that agent a has about the knowledge that agent b has about the system state, etc.

Epistemic logics were initially designed for representing reasoning capabilities of intelligent agents in multi-agent systems. They have become of interest in verification when combined with temporal logics, due to their combined ability to express properties of interest in security, like various types of non-interference, confidentiality or authentication, or to model some protocols in which agent knowledge is essential for protocol correctness [22,34,52,56,65,66]. These applications were accompanied with the development of model-checking tools [27, 47, 55] that which utilize classical model-checking techniques, and hence are based on theoretical results connecting automata with temporal and monadic logics.

However, contrary to the classical satisfiability and model-checking case, the decidability frontier is much lower in combinations of temporal and epis-

temic logics than in temporal logics. First, the above-mentioned tools are mostly based on a state-based interpretation of the epistemic modalities, which means that agents may forget what they have observed during their interaction with other agents. Perfect recall variants of the semantics of epistemic modalities in temporal epistemic logics are much harder to model-check and sometimes have undecidable satisfiability properties [63]. Secondly, if agent capabilities are taken into account in epistemic variants of alternating temporal logics – which are logics for specifying what coalitions may enforce – then in the presence of perfect recall, a decidable model-checking problem could be achieved only for single-agent coalition logics or when coalitions behave like a single agent, in that agents exchange their information and compose what is called *distributed knowledge* in order to find their winning strategies [20, 31]. (Note that the status of the decidability of the satisfiability problem remains open). Thirdly, two-agent coalitions without information exchange and with very simple safety winning conditions may be able to simulate Turing machines, and hence the model-checking problem for alternating temporal logics with at least two agents is undecidable [11]. The same holds when one considers coalitions that cooperate by constructing their strategies based on their *common knowledge*. And finally, when one combines continuous time and epistemic operators, then even the simplest logic with perfect recall becomes undecidable [16].

The organization of this paper is the following: in the next section, we give a brief introduction into epistemic logics (without temporal dimensions). Section 2 presents results reported in [17, 19] on the non-axiomatizability of a linear temporal epistemic logic having only an individual knowledge operator, some related results for a branching temporal logic, and results from [17] concerning a model-checking technique for CTL enriched with individual knowledge operators. In Section 3, we give a brief overview of existing results on Alternating Temporal Logic (ATL) with incomplete information, then report on our self-contained proof of the undecidability of the model-checking problem for this logic [11] and on the result from [16] on the undecidability of ATL with coalitions constructing strategies based on common knowledge, and further then give an alternative semantics for which ATL has a decidable model-checking problem (even if we allow both individual and distributed knowledge operators), results from [20, 30]. In Section 4, we present a variant of a continuous-time temporal epistemic logic and some (positive and negative) results on the decidability of the model-checking problem for this logic, results from [18]. We end with a section with conclusions and directions for further work.

The results on ATL with partial information are the fruit of collaborations with Rodica Bozianu, Raluca Diaconu, Constantin Enea, Dimitar Guelev and Ferucio Ţiplea. This paper is based on the papers [11, 16–20, 30–32].

1. BASICS ON PROPOSITIONAL EPISTEMIC LOGIC

Epistemic logics are modal logics in which the modal operators model various types of knowledge that agents may have about the system state, and about the other agents' knowledge of this system state, etc. These logics manipulate the following operators:

1. *Individual knowledge* operators K_a, indexed by the name of an agent a belonging to a fixed set of agents Ag. The dual of each individual knowledge operator, denoting the fact that an agent a considers *possible* some fact, is denoted P_a.
2. *Distributed knowledge* operators K_A, indexed by a *set of agents* $A \subseteq Ag$, operator also denoted D_A.
3. Operators denoting the fact that "*everybody in a group knows*" some fact, denoted E_A, for $A \subseteq Ag$.
4. The *common knowledge* operators C_A, for groups of agents $A \subseteq Ag$, denoting the fact that each agent in A knows some fact, and that each agent knows that each agent knows the fact, etc.

The simplest framework for giving a semantics to the epistemic modalities is the **possible worlds semantics**, common for modal logic. Formally, a **Kripke structure** for a finite set of agents Ag is a tuple $M = \left(S, \Pi, \pi, (\mathcal{K}_a)_{a \in Ag}\right)$ where

- S is the set of *global states*. It is usual to consider that $S = \prod_{a \in Ag} L_a$ with L_a designating the set of *local states* for agent a.
- Π is a set of atomic propositions.
- $\pi : S \to 2^{\Pi}$ is the truth value assigned to the atomic propositions in each state.
- $\mathcal{K}_a$ is the *indistinguishability relation* (also called the *possibility* relation). $(s, s') \in \mathcal{K}_a$ denotes the fact that, for agent a, states s and s' cannot be distinguished by a's observations.

  In most of the literature, $\mathcal{K}_a$ is defined by the local states, in the sense that $(s, s') \in \mathcal{K}_a$ if $s\big|_a = s'\big|_a$, that is, the local state visible to agent a in both states s and s' is the same.
- Very often $\mathcal{K}_a$ are equivalence relations.

From the indistinguishability relations, we may then build also the *distributed knowledge* relation and the *common knowledge* relation, as follows:

$$\mathcal{K}_A = \bigcap_{a \in A} K_a \qquad\qquad \mathcal{C}_A = \left(\bigcup_{a \in A} K_a \right)^*$$

where $(\cdot)^*$ denotes the reflexive-transitive closure of a binary relation.

The semantics of the knowledge operators is then defined as follows:

- $(M,s) \models K_a\phi$ if $(M,s') \models \phi$ for all s' with $(s,s') \in \mathcal{K}_a$.
- $(M,s) \models K_A\phi$ if $(M,s') \models \phi$ for all s' with $(s,s') \in \mathcal{K}_A$.
- $(M,s) \models E_A\phi$ if $(M,s') \models \phi$ for all s' with $(s,s') \in \bigcup_{a\in A} \mathcal{K}_a$.
- $(M,s) \models C_A\phi$ if $(M,s') \models \phi$ for all s' with $(s,s') \in \mathcal{C}_A$.

with $(M,s) \models p$ being defined as $p \in \pi(s)$ for atomic propositions, relation extended straightforwardly to the Boolean operators.

One of the main meta-properties of this framework is that knowledge acquisition is modeled as observation, which has the effect that everything that can be deduced from a given observation is deduced "instantaneously". This framework is very useful in analyzing systems in which the amount of information about the system state is finite, and hence this ability of any agent to immediately "know" all properties which can be deduced from observations is a reasonable abstraction. However, this strong ability may be considered unrealistic in other situations, and is referred to as the *omniscience* problem. Several solutions to the omniscience problem have been proposed, see again [26]. We will not explore them here, but we mention them as they can be related with attacker capabilities in the analysis of security protocols.

2. LOGICS OF KNOWLEDGE AND TIME

The Kripke semantics from the previous section offers only limited possibilities for the analysis of systems whose states may evolve over the time and in which agents have the ability to incorporate an apriori knowledge about system evolution in their deduction about the system properties. Frameworks that are more appropriate for handling such aspects are combinations of temporal and epistemic logics. Such combinations have been studied since the mid-eighties, starting with [36,37], identifying 96 different logics, distinguished by semantics and/or the presence of common knowledge operators, and presenting decidability and undecidability results for the satisfiability problem in the presented logics. Such logics have recently proved useful in the formal verification applied to various distributed systems in which the knowledge of the participants is essential for the correctness of the system specification. Examples include the verification of confidentiality [66, 67], authentication [22, 49, 56], mutual agreement [52], various types of anonymity [34, 40, 52, 65] or privacy [12].

If we stick to the classical dichotomy between linear and branching time [50], epistemic logics can be combined either with the linear temporal logic, or with branching temporal logics like CTL or CTL*. The syntax of CTL* endowed with epistemic modalities, which subsumes both linear and branching time, is given by the following grammar:

$$\phi ::= p \mid \phi \wedge \phi \mid \neg\phi \mid \bigcirc \phi \mid \phi \mathcal{U} \phi \mid E\phi \mid K_a\phi.$$

Here $\bigcirc$ is the nexttime operator, $\mathcal{U}$ the until operator and E is the existential path quantifier. Distributed knowledge and common knowledge operators can be added to this logic, with the usual notation.

It is common to consider that observability relations are given by so-called *local states* of each agent, which gives what is called the *interpreted systems* semantics [33]. Hence, a global state is a tuple of local states, and system evolutions are presented as runs in a transition system – that is, by combining the usual Kripke semantics for time with the Kripke semantics for the epistemic operators. As usual, the semantics of the temporal operators is interpreted over *positions along runs* in the transition system. These tuples are also called *points* by the epistemic logic community [26], and represent (instantaneous) *local states* in the epistemic Kripke structure.

Formally, a multi-agent transition system with agents in Ag is a tuple

$$\mathcal{M} = \left(S, (L_a)_{a\in Ag}, \rightarrow, S_0, (\sim_a)_{a\in Ag}, \Pi, \pi\right)$$

whose components satisfy the following properties:

1. Ag is a finite set of *agents*.
2. L_a is the *local state*, which itself is the component of the *global state* that agent $a \in Ag$ can observe.
3. S is the set of *global states*, and is defined as: $S = \prod_{a\in Ag} L_a$. For a global state $s = (l_a)_{a\in Ag}$ we denote $s\big|_a = l_a$, the element indexed a from the tuple s.
4. The tuple $(S, \rightarrow, S_0)$ is the *underlying transition system* for $\mathcal{M}$, and hence $\rightarrow \subseteq S \times S$ and $S_0 \subseteq S$.

   Runs in S are finite or infinite sequences of states connected by transitions, $(s_i)_{i\in I}$ with $(s_i, s_{i+1}) \in \rightarrow$. Furthermore, the set of *points* of this transition system is the set:

   $$Points(\mathcal{M}) = \{(\rho, i) \mid \rho \text{ is an infinite run in } \mathcal{M} \text{ and } i \in \mathbb{N}\}.$$

5. $\sim_a$ is the *observability relation* (also called the *indistinguishability relation*) for agent a, and is defined on the set of points of $\mathcal{M}$, $\sim_a \subseteq Points(\mathcal{M}) \times Points(\mathcal{M})$.
6. π is the interpretation of atomic propositions, $\pi : S \rightarrow 2^{\Pi}$.

In the most general presentation, the observability relations are simply relations on $Points(\mathcal{M})$, without any further constraint. But this framework is of limited interest in applications of temporal logics of knowledge. Instead, supplementary constraints are imposed on the indistinguishability relations, constraints related with the possibility that agents have to remember their observations along the time, to foresee the future behavior of the system and/or to have access to a global clock.

The first property of interest is *perfect recall* (also called *non-forgetting*), and models the situation in which an agent is able to memorize *changes in local states* upto the current moment, and to tell apart two runs which do not have the same history of changes in the local state. The second property is the dual of perfect recall and is called *no learning*, and models situations in which agents are provided, from the very beginning the sequence of observations that they will make during system behavior. The third property is *synchrony*, and models situations in which agents know the exact absolute time of each point (that is, the i component of the point). These properties can be combined in any way, giving further constraints to the observability relations.

The formal introduction of these properties requires some further notations on transition systems. The set of finite runs is denoted $\mathsf{FinRuns}(\mathcal{T})$, and the set of infinite runs is denoted $\omega\mathsf{Runs}(\mathcal{T})$. The i-th state in the run ρ is denoted $\rho[i]$. Also, given a run $\rho = (s_i)_{0\leq i<k}$ of length k ($k \in \mathbb{N}\cup\{\infty\}$), and some $k' \leq k$, the *prefix of length* k' of ρ is the run denoted $\rho[0..k'] = (s_i)_{0\leq i<k'}$. The suffix of ρ starting at k' is denoted $\rho[k'..]$.

Furthermore, given a transition system $\mathcal{T} = (S, \to, S_0)$ and a surjective mapping $f : S \to S'$, the set S' can be endowed with a transition system structure as usual, $\mathcal{T}' = (S', \to_f, S'_0)$ with $S'_0 = f(S_0)$ and $u \to_f u'$ if there exists $s, s' \in S$ with $f(s) = u, f(s') = u$ and $s \to s'$. We then say that $\mathcal{T}'$ is the *projection* of $\mathcal{T}$ by f. In the same setting, given a run $\rho = \left(s_i\right)_{1\leq i<\eta} \in \mathsf{Runs}(\mathcal{T})$, the *projection* of ρ is the following run of $\mathcal{T}'$: $f(\rho) = \left(f(s_i)\right)_{1\leq i<\eta}$. Here we will have situations in which $\mathcal{T}$ and $\mathcal{T}'$ are such that $S = 2^T$ and $S' = 2^V$ for some sets T and $V \subseteq T$, and the projection is defined by $f(X) = X \cap V$. In these situations, the projection f is also denoted $\cdot\big|_V$, notation which is also employed for the projection of a run ρ by f: we denote $\rho\big|_V$ instead of $f(\rho)$.

Given a run $\rho = \left(s_i\right)_{1\leq i<\eta} \subseteq \mathsf{Runs}(\mathcal{T})$, the *removal of suttering steps from* ρ, denoted $\mathsf{stut}(\rho)$, is the sequence of states s_{i_j} resulting by removing any successive states that are identical in ρ. For example, for $\rho = (s_1s_2s_2s_2s_3s_2s_3s_1s_1s_1)$, then $\mathsf{stut}(\rho) = (s_1s_2s_3s_2s_3s_1)$. Note that $\mathsf{stut}(\rho)$ is also a run of $\mathcal{T}$.

We are now in position to define formally the various observability relations:

Definition 1. 1. The **perfect recall observability** (pr-observability) relation for agent $a \in Ag$ is the relation defined as following:

$$\sim_a^{pr} \subseteq Points(\mathcal{M}) \times Points(\mathcal{M}), \quad (\rho, i) \sim_a^{pr} (\rho', i') \text{ if } \mathsf{stut}(\rho[0..i]\big|_{L_a}) = \mathsf{stut}(\rho'[0..i']\big|_{L_a})$$

2. The **non-learning observability** (nl-observability) relation for agent

$a \in Ag$ is the relation defined as following:

$$\sim_a^{nl} \subseteq Points(\mathcal{M}) \times Points(\mathcal{M}), \quad (\rho, i) \sim_a^{nl} (\rho', i') \text{ if } \mathsf{stut}(\rho[i..]\big|_{L_a}) = \mathsf{stut}(\rho'[i'..]\big|_{L_a})$$

3. The **synchronous observability** (s-observability) relation for agent $a \in Ag$ is the relation defined as following:

$$\sim_a^{s} \subseteq Points(\mathcal{M}) \times Points(\mathcal{M}), \quad (\rho, i) \sim_a^{s} (\rho', i') \text{ if } i = i'$$

4. The **perfect recall and synchronous observability** (prs-observability) relation for agent $a \in Ag$, which is the relation defined as following:

$$\sim_a^{prs} \subseteq Points(\mathcal{M}) \times Points(\mathcal{M}), \quad (\rho, i) \sim_a^{prs} (\rho', i') \text{ if } i = i' \text{ and } \rho[0..i]\big|_{L_a} = \rho'[0..i]\big|_{L_a}$$

Note that agents that have s-observability can always distinguish two finite runs having different lengths.

The semantics of CTL* with epistemic modalities is given by the following rules, parameterized by the type of observability relation $rel \in \{pr, nl, s, prs\}$ and in which $(\rho, i) \in Points(\mathcal{M})$:

$(\mathcal{M}, \rho, i) \vDash_{rel} p$	if $p \in \pi(\rho[i])$
$(\mathcal{M}, \rho, i) \vDash_{rel} \phi_1 \wedge \phi_2$	if $(\mathcal{M}, \rho, i) \vDash_{rel} \phi_1$ and $(\mathcal{M}, \rho, i) \vDash_{rel} \phi_2$
$(\mathcal{M}, \rho, i) \vDash_{rel} \neg\phi$	if $(\mathcal{M}, \rho, i) \nvDash_{rel} \phi$
$(\mathcal{M}, \rho, i) \vDash_{rel} \bigcirc \phi$	if $(\mathcal{M}, \rho, i+1) \vDash_{rel} \phi$
$(\mathcal{M}, \rho, i) \vDash_{rel} \phi_1 \mathcal{U} \phi_2$	if $\exists j \geq i$ with $(\mathcal{M}, \rho, j) \vDash_{rel} \phi_2$ and $\forall i \leq k < j$, $(\mathcal{M}, \rho, i) \vDash_{rel} \phi_1$
$(\mathcal{M}, \rho, j) \vDash_{rel} E\phi$	if $\exists \rho' \in Points(\mathcal{M})$ such that $\rho'[0..i] = \rho[0..i]$ and $(\mathcal{M}, \rho', i) \vDash_{rel} \phi$
$(\mathcal{M}, \rho, i) \vDash_{rel} K_a\phi$	if $\forall \rho' \in Points(\mathcal{M})$ and $\forall j \in \mathbb{N}$ with $(\rho, i) \sim_a^{rel} (\rho', j)$ we have $(\mathcal{M}, \rho', j) \vDash_{rel} \phi$.

Subclasses of CTL* with epistemic modalities have been identified since [36, 37], differentiated either by the type of observability relation, or by using linear or branching time, or even by the inclusion or the exclusion of common knowledge operators. The linear-time variants were denoted with KL and the branching-time variants with KB, with subscripts identifying the number of agents. Hence, the notation KL_n refers to the linear-time epistemic logic without common knowledge operators with n agents, while CKB_1 refers to the branching-time logic with common knowledge operators with only one agent. All these logics are then interpreted over classes of interpreted systems denoted

$\mathcal{C}$ with subscripts denoting the type of observability. Hence $\mathcal{C}_{sync,nf}$ denotes the class of interpreted systems with synchronous and non-forgetting observability.

The axiomatizability of these logics has been addressed in [35], and the decidability of their satisfiability problem in [36]. Most notably, the undecidability results have been related with undecidability results for temporal logics on bidimensional structures, observing that especially the presence of the common knowledge operator makes it possible to encode grid-like structures.

On the other hand, there has been some interest in relating the interpreted systems semantics with general Kripke semantics, in which the indistinguishability relations are defined on global states [48]. We will come back to this problem in the following subsection.

2.1. **A concrete observability relation**

In many of the above-mentioned applications of temporal epistemic logics, the observability relations are intimately related with the truth values for atomic propositions. For instance, in [45, 49, 57, 65], the local states for each agent incorporate items describing what agents *sent/received* and what they *possess* during the protocol behavior, and there are atomic formulas encoding exactly the possession, the reception or the issue of items during the protocol run. It is therefore quite natural to consider a "concrete" observability relation, in which local states for an agent a are characterized by truth values for some fixed subset of atomic propositions Π_a.

Formally, such observability relations can be stated as follows:

Definition 2. A multi-agent system $\mathcal{M}$ **has concrete observability** for agent $a \in Ag$ if there exists a subset of atomic propositions $\Pi_a \subseteq \Pi$ such that for each $s, s' \in L$, $s\big|_a = s'\big|_a$ if and only if $\nu(s) \cap \Pi_a = \nu(s') \cap \Pi_a$.

The set Π_a is the set of atomic propositions whose truth values can be observed by the agent a *in any state of the system*. It is essential to note the equivalence, required by this definition, between the state-based observability and the observability of a specific set of atomic propositions, Π_a.

In such contexts, we may redefine the observability relations discussed in the previous section as follows:

Definition 3 ([19]). A multi-agent system in which all agents have **concrete perfect recall observability** (*cpr*-observability) is a system $\mathcal{M} = \big(S, \to, S_0, (\sim_a^{cpr})_{a\in Ag}, \Pi, (\Pi_a)_{a\in Ag}, \pi\big)$ in which $\Pi_a \subseteq \Pi$ for all agents $a \in Ag$ and

(1) $(\rho, i) \sim_a^{cpr} (\rho', i')$ if and only if $\mathsf{stut}(\pi(\rho[0..i])\big|_{\Pi_a}) = \mathsf{stut}(\pi(\rho'[0..i'])\big|_{\Pi_a})$.

A multi-agent system in which all agents have **concrete non-learning observability** (*cnl*-observability) is a system $\mathcal{M} = \big(S, \to, S_0, (\sim_a^{cnl})_{a\in Ag}, \Pi,$

$(\Pi_a)_{a\in Ag}, \pi\big)$ in which $\Pi_a \subseteq \Pi$ for all agents $a \in Ag$ and

(2) $(\rho, i) \sim_a^{cnl} (\rho', i')$ if and only if $\mathsf{stut}(\pi(\rho[i..])\big|_{\Pi_a}) = \mathsf{stut}(\pi(\rho'[i'..])\big|_{\Pi_a})$

A multi-agent system in which all agents have **concrete perfect recall and synchronous observability** (*cprs*-observability): is a system $\mathcal{M} = \big(S, \to, S_0, (\sim_a^{cprs})_{a\in Ag}, \Pi, (\Pi_a)_{a\in Ag}, \pi\big)$ with

(3) $(\rho, i) \sim_a^{cprs} (\rho', i')$ if and only if $i = i'$ and $\pi(\rho[0..i])\big|_{\Pi_a} = \pi(\rho'[0..i])\big|_{\Pi_a}$

At a first sight, it may look that the semantics is just a particular case of the semantics based on interpreted systems: on one hand, in the interpreted systems semantics, one may include the local states in the set of atomic propositions and declare the set of atomic propositions corresponding to the local states for agent a as the set of observable atoms for a. For the reverse, one might try to axiomatize the correspondence between local states and validity of Π_a, by considering axioms $\vdash p \to K_a p$ and $\vdash \neg p \to K_a \neg p$ for each atomic proposition $p \in \Pi_a$.

Unfortunately, this set of axioms does not completely characterize observability for agent a by means of observability of truth values for Π_a. In fact, as we prove in [19], the concrete semantics is not axiomatizable. We prove this result for both the perfect recall semantics, and for the perfect recall and synchronous case.

THEOREM 1 ([17, 19]). 1. *Satisfiability of LTLK or CTLK formulas w.r.t. the cprs-observability semantics is undecidable.*

2. *The semantics of LTLK or CTLK based on cprs-observability or on cpr-observability do not have a recursively enumerable axiomatization.*

The reason for these results is that there is no possibility to impose, axiomatically, the identical observability of two histories, on the basis of the observability of truth values for some given subset of atomic propositions. To compare with the classical framework of interpreted systems, note that [35] gives complete axiom systems for both cases, and their satisfiability problem is shown decidable in [36].

The interest in studying this concrete observability semantics is twofold. First, it is related with the problem of finding the exact relationship between interpreted systems semantics and general Kripke semantics. In [48], for special types of interpreted systems, called *hypercube systems*, the two semantics are shown to be equivalent, and the authors suggest that "*further research could be undertaken [..] to have a general methodology for translating interesting classes of interpreted systems into classes of Kripke models*". The results from our paper [19] show that this equivalence does not hold in general.

The second reason defending this research is related with the possibility to compare the expressive power of temporal epistemic logics with the expressive power of fragments of monadic logics over tree structures with some auxiliary interpreted predicates. As known, epistemic temporal logics are interpreted over transition systems endowed with some additional relations. If we have in mind that unfoldings of transition systems are infinite trees, and that MSO, the monadic second order logic of trees [59], accounts as the reference logic for specifying system behaviors as it is as expressive as the mu-calculus, tree automata or quantified propositional temporal logics, one may ask the question how combinations of temporal and epistemic logics compare, in expressivity, with MSO. The concrete observability semantics gives a natural setting for exploring this question, because the propositional symbols can be interpreted as 2nd order monadic variables in MSO, (*i.e.* as sets of positions in a tree).

One of the sources of inspiration for the concrete semantics is the Logic of Local Propositions from [25]. The atomic propositions in the sets Π_a are interpreted as local propositions for agent a, in the sense of [25]. But, contrary to [25], in the concrete semantics there is no quantifications over atomic propositions, and, as such, our logical framework is strictly less expressive than the Logic of Local Propositions.

The result from [19] is based on a proof of the undecidability of the satisfiability problem for the Linear Temporal Logic of Knowledge, with either the perfect recall semantics or the perfect recall and symmetric semantics – that is, the logic KL_n interpreted in the concrete variant of the class $\mathcal{C}_{sync,nf}$, if we are to stick to the acronyms used in [36]. This proof of the undecidability of satisfiability works by coding the configurations and transitions along the computation of a Turing machine as runs in a multi-agent system. This proof idea was utilized many times in the literature for proving undecidability of various epistemic temporal logics, starting from [36] where it is proved that CKL_n, which is LTL *with common knowledge operators*, has an undecidable satisfiability problem. But recall that KL_n interpreted over $\mathcal{C}_{nf}$, resp. $\mathcal{C}_{nf,sync}$, is proved to have a decidable satisfiability problem in [36]. We also cite the undecidability result of [63] for LTL *with common knowledge too*, which utilizes the same type of argument. Another paper which utilizes this argument is [60], in which it is shown that several variants of branching-time logics *with common knowledge operators* have an undecidable satisfiability problem. Contrary to these results, our undecidability result holds *without a common knowledge operator*. To complete the figure, recall that the only undecidable cases of temporal logics without common knowledge studied in [36] concern only *non-learning* variants of observability, based on previous results from [43]. But the coding technique used in [43] heavily relies on asynchrony, and on the non-learning character of

the semantics. Our proof uses a different technique, which relies on synchrony and the possibility to manipulate, in the logic, formulas which exactly identify observations (by means of the atomic propositions which are observable for each agent).

A similar undecidability result is proved in our paper [17] for the branching-time epistemic logic with concrete perfect recall and synchronous observability, that is, for KB_n interpreted in the concrete variant of $\mathcal{C}_{sync,nf}$. The technique used in [19] for transferring this result to observability relations which are not synchronous (but have perfect recall) can be applied with no difficulty to the branching case.

2.2. **Model-checking for logics of knowledge and time**

The model-checking problem is one of the central problems in verification [15]. In its most general presentation, it can be stated as follows:

Problem 1. Given a class of models $\mathcal{C}$ and a class of formulas $\mathcal{F}$ into some logic $\mathcal{L}$, does there exist an algorithm that, given as inputs a model $\mathcal{M} \in \mathcal{C}$ and a formula $\phi \in \mathcal{F}$, can check whether $\mathcal{M} \models \phi$?

The model-checking problem for temporal logics of knowledge has started receiving attention in relation with the verification of communication protocols in which the knowledge of the participants is important for ensuring protocol correctness, like in the Alternating Bit Protocol [67], the Chaum's Dining Cryptographers Protocol [41,65], or in security-related protocols [45].

The model-checking problem for perfect recall and/or synchronous semantics for temporal logics of knowledge has been studied in [17,54,63,64]. The approach of [54,63], called *model-checking at a run*, is to consider a restricted problem which binds also the run on which the formula is to be checked. The approach proposed in the latter paper is to code the model-checking problem as a satisfiability problem in Chain Logic [24,58].

Some decidable cases of the general problem for linear temporal logics with individual knowledge and perfect recall and synchronous semantics are presented in [64], where the complexity of the model-checking problem is shown to be nonelementary. The approach proceeds by state splitting of the model, embodying sufficient information that is needed for checking subformulas with nested knowledge operators. This requires a subset construction for each knowledge operator, and hence, for formulas containing at most k nested knowledge operators, the state splitting leads a state explosion which in the worst case is a tower of k exponentials in the size of the initial model. This technique has been implemented in the model-checker MCK [27].

On the negative side, the inclusion of the common knowledge operators, in conjunction with perfect recall semantics was shown to make undecidable the model-checking problem [63,64].

The model-checking problem for *state-based observability relations* has been extensively studied by A. Lomuscio and his collaborators, leading to a now well-developed tool and methodology [47]. State-based observability is defined as the relation on points in interpreted systems which is only based of the "current" state of the point. More precisely, for any two points $(\rho, i), (\rho', i') \in Points(\mathcal{M})$ and agent a,

$$(\rho, i) \sim_a (\rho', i') \text{ iff } \rho[i] \sim_a \rho'[i'],$$

where $\rho[i]$ is the i-th global state in the run ρ, and $\sim_a$ is the usual observability relation based on the local-state decomposition of global states. State-based observability may show useful in systems where the information available to each agent at each time point can be encoded in the system states – that is, agents only need a finite amount of memory for remembering the information they can ever see about the system behavior.

In [17] we take a direct approach for the model-checking problem for KB_n, the combination of CTL with individual knowledge operators. Our approach is to adapt the classical model-checking algorithm of [14], with the aid of an extra procedure. This extra procedure is a state labeling with knowledge formulas which involves a subset construction on the given model, since one needs to identify all histories which may be identically observed by agent i, when one wants to label states with formulas involving the K_i modality. Note also that the subset construction is also essential in the model-checking algorithm for LTL with knowledge from [64].

This approach does not improve the worst-case complexity of the algorithm, since each nesting of knowledge operators induces an exponential explosion, thus leading to a nonelementary complexity. But we believe that our approach could be more practical for formulas with low nesting of knowledge operators. In the approach of [54], the system is first translated into the Chain Logic, which needs then to be coded into Monadic Second Order Logic [58], and then an MSO-based tool like Mona [23] has to be applied. In such an approach, since the system coding creates some formula with quantifier alternation, some unnecessary determinization steps for the resulting Büchi automata are then needed. Our approach avoids this, as each non-knowledge operator requires only state relabeling, and no state explosion.

Another technique that was proposed in [55] is the encoding of instances of the model-checking problem into the above-cited *Logic of Local Propositions*. For state-based observability, which is subsumed by the approach of [47], this

leads to encodings of instances of the model-checking problems for temporal epistemic logics into instances of the model-checking problems for purely temporal logics. But, as noted in [27], this approach leads to an "explosion in the number of temporal formulas that need to be checked when there are negative occurrences" (of the knowledge operators).

3. COALITIONS AND KNOWLEDGE

Alternating-time Temporal Logic (ATL) [5, 6] is a generalization of the Computational Tree Logic (CTL) in which path quantifiers "E" and "A" are replaced by *cooperation modalities* $\langle\langle A\rangle\rangle$ in which A denotes a set of *agents* who act as a *coalition*. A formula $\langle\langle A\rangle\rangle\phi$ expresses the fact that the agents in *coalition* A can cooperate to ensure that ϕ holds in an appropriate type of multiplayer game.

The precise semantics of the cooperation modalities varies depending on whether the knowledge that each agent has of the current state of the game is complete or not, and whether agents can use their knowledge of the past game states when deciding on their next move or not. These alternatives are known as *complete*, resp. *incomplete information*, and *perfect*, resp. *imperfect recall*. In the case of imperfect recall further subdivisions depend on how much memory an agent is allowed for storing information on the past in addition to its possibly incomplete view of the current state. In the extreme case agents and, consequently, the strategies they can carry out, are *memoryless*.

The formal analysis of multi-agent systems has generated some interest in the study of combinations of ATL with modal logics of knowledge [39, 62]. Such combinations can be viewed as related to temporal logics of knowledge (cf. *e.g.* [26]) in the way ATL is related to computational tree logic CTL. Epistemic goals make it essential to study strategic ability with incomplete information. Variants of the cooperation modalities which correspond to different forms of coordination within coalitions were proposed in [39]. [38] proposes a combination of ATL with the epistemic modalities for collective knowledge. In that system formulas are interpreted at *sets* of states and the existence of strategies which are winning for all the epistemically indiscernible states can be expressed by combining epistemic and cooperation modalities. Such strategies are called *uniform* with respect to the corresponding form of collective knowledge. The survey [9] gives an overview of the variants of the semantics of ATL in the presence of perfect as well as imperfect information.

Along with the alternating transition systems proposed in [6], ATL has been given semantics on *interpreted systems*, which are known from the study of knowledge-based programs [26], and other structures, some of which have

been shown to be equivalent [28]. Most of the proposed extensions of ATL and other temporal logics by epistemic modalities include only the future temporal operators and the observability relations which are needed for the semantics of the S5 epistemic modalities are either defined as the equality of current local states of the corresponding agents or assumed to be given explicitly in the respective structures and required to respect equality of local state [46,61].

Formally, the syntax of ATL is given by the following grammar:

$$\phi ::= p \mid \phi \wedge \phi \mid \neg\phi \mid \langle\!\langle A \rangle\!\rangle \bigcirc \phi \mid \langle\!\langle A \rangle\!\rangle \phi \mathcal{U} \phi \mid \langle\!\langle A \rangle\!\rangle \phi_1 \mathcal{W} \phi_2 \mid K_A \phi,$$

where p ranges over the set Π of atomic propositions, and A ranges over the set of subsets of Ag. According to [10,44], the weak-until operator must be included in order to fully capture all the temporal operators in the scope of the coalition operators.

The semantics is given in terms of transitions systems with actions, named *game arenas*.

Definition 4. A **game arena** is a tuple $\Gamma = (Ag, Q, (Act_a)_{a\in Ag}, \to, Q_0, (\sim_a)_{a\in Ag}, \Pi, \lambda)$, where

- Ag is a set of agents.
- Q is a set of *states*.
- Act_a is a finite set of *actions* available to agent a. We write Act_A for $\prod_{a\in A} Act_a$ and Act for Act_{Ag}.
- $Q_0 \subseteq Q$ is the set of *initial states*.
- $\sim_a$ is the *indistinguishability relation* for agent a, which is supposed to be an equivalence relation.
- Π, is a set of *atomic propositions*.
- $\lambda : Q \to 2^{\Pi}$ is the *state-labeling function*.
- $(Q, C, \to, Q_0, \lambda)$ is a *labeled transition system*, in the sense that $\to \subseteq Q \times C \times Q$, with the further requirement that for any $q \in Q$ and $c \in C$, there exists some $q' \in Q$ with $q \xrightarrow{c} q'$.

An element $c \in C$ will be called an *action tuple*. We write $q \xrightarrow{c} r$ for *transitions* $(q, c, r) \in \delta$.

Given a run $\rho = q_0 \xrightarrow{c_1} q_1 \xrightarrow{c_2} \ldots$, we denote q_i by $\rho[i]$, $i = 0, \ldots, |\rho|$, and c_{i+1} by $act(\rho, i)$, $i = 0, \ldots, |\rho|-1$. We write $\rho[0..i]$ for the *prefix* $q_0 \xrightarrow{c_1} q_1 \xrightarrow{c_1} \ldots \xrightarrow{c_i} q_i$ of ρ of length i.

A *coalition* is a subset of Ag. Given a coalition $A \subseteq Ag$, the *distributed indistinguishability relation* for A is the relation $\sim_A = \bigcap_{a\in A} \sim_a$. This is the same definition of distributed indistinguishability from Kripke structures for epistemic temporal logics. Also the common knowledge indistinguishability

can be defined as $\sim_{C_A} = \left(\bigcup_{a \in A} \sim_a\right)^*$, with $(\cdot)^*$ denoting reflexive-transitive closure, similar to the temporal epistemic framework.

Runs ρ and ρ' are *indistinguishable (observationally equivalent)* to coalition A, denoted $\rho \sim_A \rho'$, if $|\rho| = |\rho'|$, $act(\rho, i)\big|_A = act(\rho', i)\big|_A$ for all $i < |\rho|$, and $\rho[i] \sim_A \rho'[i]$ for all $i \leq |\rho|$. Again, this is the straightforward generalization of the synchronous and perfect recall observability relations to the multi-player games setting presented here.

Definition 5 ([16, 20]). A **strategy for an agent** a is any mapping $\sigma : (Q/\sim_a)^* \to Act_a$. The set of strategies for agent a in the game arena Γ is denoted $\Sigma(a, \Gamma)$.

A **strategy for a coalition** A is a tuple of strategies with incomplete information for each member of the coalition. The set of strategies for coalition A in the game arena Γ is denoted $\Sigma_{coal}(A, \Gamma)$.

A **cooperative strategy for a coalition** A is any mapping $\sigma : (Q/\sim_A)^* \to Act_A$. The set of cooperative strategies for coalition A in the game arena Γ is denoted $\Sigma_{coop}(A, \Gamma)$.

A **strategy based on common knowledge for a coalition** A is any mapping $\sigma : (Q/\sim_{C_A})^* \to Act_A$. The set of strategies based on common knowledge for coalition A in the game arena Γ is denoted $\Sigma_{cl}(A, \Gamma)$.

A strategy σ of any of the above type is **memoryless** if for any sequence of items $w \in \mathsf{dom}(\sigma)$, if $last(w) = last(w')$ then $\sigma(w) = \sigma(w')$.

All the above definitions introduce strategies with *incomplete information*. Strategies with complete information for an agent a or for a coalition A are defined as mappings $\sigma : Q^* \to Act_a$, resp $\sigma : Q^* \to Act_A$.

An individual strategy σ for agent a is *compatible* with a run $\rho = q_0 \xrightarrow{c_1} q_1 \xrightarrow{c_2} \ldots$ if

$$\sigma(\left[\rho[0]\right]_{\sim_a} \cdots \left[\rho[i]\right]_{\sim_a}) = c_{i+1}\big|_a$$

for all $i \leq |\rho|$.

A cooperative strategy σ for coalition A is *compatible* with a run $\rho = q_0 \xrightarrow{c_1} q_1 \xrightarrow{c_2} \ldots$ if

$$\sigma(\left[\rho[0]\right]_{\sim_A} \cdots \left[\rho[i]\right]_{\sim_A}) = c_{i+1}\big|_A$$

for all $i \leq |\rho|$. The compatibility relation for strategies with common knowledge can be defined similarly.

Obviously if σ is compatible with run ρ then it is compatible with any run that is indistinguishable from ρ to a (resp. A if it's a cooperative strategy).

Cooperating strategies for coalitions are introduced in [20, 30]. In such strategies, coalition members have a communication mechanism which enables the coalitions to carry out strategies that are based on their *distributed knowledge*. (Recall that a coalition has *distributed knowledge* of fact ϕ iff ϕ is a logical

consequence of the combined knowledge of the coalition members.) We assume that a coalition has a strategy to achieve a goal ϕ only if the same strategy can be used in all the cases which are indistinguishable from the actual one with respect to the distributed knowledge of the coalition. This choice is known as *de re* strategies [38], and rules out the possibility for a coalition to be able to achieve ϕ by taking chances, or to be able to achieve ϕ in some of the cases which are consistent with its knowledge and not in others. This variant of ATL which is obtained by adopting these conventions is called *Alternating Time Logic with Knowledge and Communicating Coalitions*, and denoted ATL_{iR} indicate distributed knowledge, incomplete information and perfect recall characteristics of this logic.

On the other hand, cooperating strategies based on common knowledge are introduced in [16]. In this setting, the agents identify their abilities to achieve a common goal only based on their common knowledge of the current state of the system. This setting could be interpreted as the minimal scenario requiring communication between agents when establishing a coalition: practically, the agents need only agree on their common goal, no other information exchange is needed concerning the local state of each agent.

Satisfaction of ATL_{iR} formulas is defined with respect to a given arena Γ, a run $\rho \in \mathsf{Runs}^{\omega}(\Gamma)$, a position i in ρ by the clauses and depends on the type of coalitions used for interpreting the coalition operators, which is used as a subscript of the modeling relation $\vDash_X$ with $X \in \{coal, coop, cnl\}$:

- $(\Gamma, \rho, i) \vDash_X p$ if $p \in \lambda(\rho[i])$.
- $(\Gamma, \rho, i) \vDash_X \phi_1 \wedge \phi_2$, if $(\Gamma, \rho, i) \vDash_X \phi_1$ and $(\Gamma, \rho, i) \vDash_X \phi_2$.
- $(\Gamma, \rho, i) \vDash_X \neg\phi$ if $(\Gamma, \rho, i) \nvDash_X \phi$.
- $(\Gamma, \rho, i) \vDash_X \langle\!\langle A \rangle\!\rangle \bigcirc \phi$ if there exists a strategy $\sigma \in \Sigma_X(A, \Gamma)$ such that $(\Gamma, \rho', i+1) \vDash_X \phi$ for all runs $\rho' \in \mathsf{Runs}^{\omega}(\Gamma)$ which are compatible with σ and satisfy $\rho'[0..i] \sim_A \rho[0..i]$.
- $(\Gamma, \rho, i) \vDash_X \langle\!\langle A \rangle\!\rangle \phi_1 \mathcal{U} \phi_2$ iff there exists a strategy $\sigma \in \Sigma_X(A, \Gamma)$ such that for every run $\rho' \in \mathsf{Runs}^{\omega}(\Gamma)$ which is compatible with σ and satisfies $\rho'[0..i] \sim_A \rho[0..i]$ there exists $j \geq i$ such that $(\Gamma, \rho', j) \vDash_X \phi_2$ and $(\Gamma, \rho', k) \vDash_X \phi_1$ for all $i \leq k \leq j-1$.
- $(\Gamma, \rho, i) \vDash_X \langle\!\langle A \rangle\!\rangle \phi_1 \mathcal{W} \phi_2$ iff there exists a strategy $\sigma \in \Sigma_X(A, \Gamma)$ such that for every run $\rho' \in \mathsf{Runs}^{\omega}(\Gamma)$ which is compatible with σ and satisfies $\rho'[0..i] \sim_A \rho[0..i]$ one of the two situations occur:
  1. Either there exists $j \geq i$ such that $(\Gamma, \rho', j) \vDash_X \phi_2$ and $(\Gamma, \rho', k) \vDash_X \phi_1$ for all $k = i, \ldots, j-1$.
  2. Or $(\Gamma, \rho', k) \vDash_X \phi_1$ for all $k \geq i$.

– $(\Gamma, \rho, i) \vDash_X K_A\phi$ iff $(\Gamma, \rho', i) \vDash_X \phi$, for all runs $\rho' \in \mathsf{Runs}^\omega(\Gamma)$ which satisfy $\rho'[0..i] \sim_A \rho[0..i]$.

The semantics for logics in which strategies have complete information can be defined similarly, by removing any reference to runs that give identical observations.

The rest of the combinations between the temporal connectives and the cooperation modalities $\langle\!\langle A \rangle\!\rangle$ and $[\![A]\!]$ are defined as follows:

$$P_A\phi = \neg K_A \neg\phi \qquad [\![A]\!]\bigcirc\phi = \neg\langle\!\langle A\rangle\!\rangle\bigcirc\neg\phi$$
$$[\![A]\!]\phi\,\mathcal{U}\,\psi = \neg\langle\!\langle A\rangle\!\rangle(\neg\psi\,\mathcal{W}\,\neg\psi \wedge \neg\varphi) \qquad [\![A]\!]\phi\,\mathcal{W}\,\psi = \neg\langle\!\langle A\rangle\!\rangle(\neg\psi\,\mathcal{U}\,\neg\psi \wedge \neg\varphi)$$
$$\langle\!\langle A\rangle\!\rangle\Diamond\,\phi = \langle\!\langle A\rangle\!\rangle \mathsf{true}\,\mathcal{U}\,\phi \qquad \langle\!\langle A\rangle\!\rangle\Box\,\phi = \langle\!\langle A\rangle\!\rangle\phi\,\mathcal{W}\,false$$
$$[\![A]\!]\Diamond\,\phi = [\![A]\!]\mathsf{true}\,\mathcal{U}\,\phi \qquad [\![A]\!]\Box\,\phi = [\![A]\!]\phi\,\mathcal{W}\,false\,.$$

A formula ϕ is *valid in a game arena* Γ, written $\Gamma \vDash \phi$, if $(\Gamma, \rho, 0) \vDash \phi$ for all $\rho \in \mathsf{Runs}^\omega(\Gamma)$. The *model-checking problem* for ATL_{iR} is to decide whether $\Gamma \vDash \phi$ for a given formula ϕ and arena Γ.

Implementing strategies which rely on distributed knowledge requires some care. For instance, simply supplying coalition members with a mechanism to share their observations with each other would have the side effect of enhancing the knowledge at each agent's disposal upon considering the reachability of subsequent goals as part of possibly different coalitions, whereas we assume that each agent's knowledge is just what follows from its personal experience at all times. One may therefore assume that coalition activities are carried out through the guidance of corresponding *virtual supervisors* who receive the coalition members' observations and previously accumulated knowledge and in return direct their actions for as long as the coalition exists without making any additional information available.

3.1. **Results on model-checking variants of ATL with incomplete information**

It is known that the model-checking problem for the case of complete information is decidable in polynomial time [5]. In the case of incomplete information and perfect recall the following theorem is stated without proof in [5], (attributed to an unpublished paper of M. Yannakakis):

THEOREM 2. *The model-checking problem for ATL with incomplete information and perfect recall is undecidable.*

In [11], we give a self-contained proof of this result. Our proof is based on a direct simulation of Turing machines by concurrent game structures under

imperfect information and perfect recall, which allows for a reduction of the non-halting problem for Turing machines to the model checking problem for ATL under imperfect information and perfect recall semantics. The configurations of a given Turing machine are encoded horizontally in the levels of the tree which represents the outcome of the desired winning strategy.

On the other hand, variants of ATL with memoryless agents have been shown to have decidable model checking in [1,53,61]. Results on model-checking ATEL with memoryless strategies can be found in [1, 42, 53, 61]. Other results on ATL with complete information can be found in [10, 28].

In [20] we prove the following result:

THEOREM 3. *The model-checking problem is decidable for ATL_{iR} (i.e. with cooperating strategies for coalitions).*

We prove our model-checking result by induction on the construction the formula to be checked, like in model-checking algorithms for ATL or CTL, with two significant differences. Firstly, the implicit distributed knowledge operator hidden in the coalition operator is handled by means of a "subset construction" for identifying states with indistinguishable histories, a technique used *e.g.* [13] for transforming games with imperfect observation to games with perfect observation. Secondly, checking whether in a given set of indistinguishable states the coalition has a strategy to achieve goal ϕ involves building a tree automaton, which can be seen as a game between the coalition (supervisor) and the rest of the agents. This game resembles the two-player games with one player having imperfect information from [13], but also has a notable difference: the goal of the player with imperfect information is *not fully observable*. Such a goal can be achieved *at different times* along different yet indistinguishable runs. Therefore, we have a bookkeeping mechanism for the time of achieving the goal along each run.

The tree automata we use employs only "occurrence" accepting conditions: the set of states occurring along each run of the tree is required to belong to some given set of sets of states. No Muller conditions, *i.e.*, no restrictions on the set of states occurring *infinitely often*, are involved.

The model-checking algorithm proceeds by constructing *refinements* of the given game arena Γ, unlike in CTL and ATL model-checking where the only modifications of the given arena are the insertion of new propositional variables (corresponding to subformulas of the formula to model-check). This refinement enables telling apart classes of histories which are indistinguishable to coalition members.

Preliminary results are reported in [30], where the knowledge modalities are required to have only argument formulas from the past subset of LTL.

In [20], ATL_{iR} has only future operators, but past LTL operators can be added to ATL_{iR} at no technical cost. Also, the model-checking algorithm for ATL_{iR} is based on tree-automata and not on the syntactical transformation of past formulas as in [30].

We also note the following result from [16]:

THEOREM 4. *The model checking problem for ATL_{iR} with strategies based on common knowledge is undecidable.*

The proof of this theorem is inspired from the proof of the undecidability of the model-checking problem for CTLK with common knowledge operators from [63].

4. A REAL-TIME EPISTEMIC LOGIC

The whole theory presented above on combinations of temporal and epistemic logics only considers discrete linear or branching time. It is natural to consider also extensions to the case of continuous time, and to study the possibility to combine the techniques used for model-checking continuous-time logics like TCTL [2], or MITL [4] with epistemic logics.

In [18] we investigate one possibility to extend these results to the continuous case. We present a logic, called TCTLK, which is based on an epistemic extension of the Timed Computational Tree Logic, with the addition of freeze operators and clock variables for capturing continuous-time passage. The epistemic operators are interpreted using a synchronous and perfect recall semantics which relies on timed automata [3]. The observability relations used for defining the epistemic operators include the ability of agents to observe truth values of some atomic propositions (and the inability to observe some others), as well as their ability to observe *clock values*, as a generalization of the ability to observe only time passage.

We show that model-checking for this continuous-time version of epistemic CTL with synchrony and perfect recall is undecidable, a result which is somewhat expectable as the discrete time model-checking algorithm needs a subset construction for both CTL and LTL endowed with epistemic operators [17,64], construction which is known not to be available for timed automata [3]. Our result is even stronger: model-checking for TCTLK is undecidable even without the freeze quantifiers from TCTL, and the result shows that only one unobservable clock suffices for undecidability. This also means that undecidability holds even if the agents are only able to observe time passage. The source of this strong result comes more from the expressive power of the models than from the logic itself, and is strongly related with the impossibility to generalize the subset construction for timed automata.

On the other hand, we show that if agents are able to fully observe clock values – and this includes the observability of the values for the freeze clocks of TCTL – then model-checking becomes decidable. The proof goes by a straightforward adaptation of the classical algorithm for TCTL model-checking [2], and is based on the fact that, with full clock observability, the subset construction only concerns the "untimed" part of the model.

Previous work on combining epistemic and continuous time expressivity includes [68]. There, a continuous time variant of CTL with knowledge operators is also presented, with a state-based semantics that includes state and clock observability, but does not allow agents to memorize the whole history of observations or to observe the absolute time. Therefore, the logic in [68] has a decidable model-checking problem. However our logic is more expressive, due to its perfect recall semantics.

The semantics of timed automata is given in [18] in a weakly-monotonic setting which draws some similarities with the work of [51]. In this setting, the time domain is divided into a sequence of closed intervals $I_1, I_2, \ldots$ with $I_i \cap I_{i+1} = \alpha$ for some $\alpha \in \mathbb{R}_{\geq 0}$. A **w-point** (**weakly-monotonic point**) is then a tuple (k, β) consisting of an interval index k and an element of the interval I_k.

This semantics avoids problems related to the density of real numbers, that lead sometimes authors to consider a "nonstrict" variant of the until operator, as in [7]. We introduce a weakly-monotonic presentation for both trajectories and runs. Trajectories are, in some sense, annotations of timed words [3] with clock information, and, as such, are a generalization of the concept of word in finite automata. Runs are continuous presentations of behaviors of timed automata, embodying the possibility to have, in a model, some finite control (locations) which is not captured by formulas. The observability relation is then introduced using projection and equivalence on trajectories. Equivalence is needed as there may be several trajectories representing the same behavior, due to the possibility to have *silent transitions* in a timed automaton. Also projection cannot be defined directly on runs, as it models "forgetting" some part of the observable state of a system.

The following subsections give the formalization of the above, and are taken from [18].

4.1. **Trajectories over continuous time**

Definition 6. A **trajectory** over $\mathcal{X}$ and a set of state symbols Π is a pair $T = (\mathcal{I}, \theta)$, where $\mathcal{I}$ is a (finite or infinite) sequence of pairs of sets of state symbols and intervals

– $\mathcal{I} = \big(S_i, [\alpha_{i-1}, \alpha_i]\big)_{1 \le i < \eta}$ with $\eta \in \mathbb{N} \cup \{\infty\}$, $S_i \subseteq \Pi$, $\alpha_{i-1}, \alpha_i \in \mathbb{R}_{\ge 0}$, $\alpha_{i-1} \le \alpha_i$ and $\alpha_0 = 0$.

and θ is a continuous mapping of clock intervals:

– $\theta = (\theta_i)_{1 \le i < \eta}$ with $\theta_i : [\alpha_{i-1}, \alpha_i] \times \mathcal{X} \to \mathbb{R}_{\ge 0}$

subject to the following properties.

1. For all $i < \eta$ and all $t, t' \in [\alpha_{i-1}, \alpha_i]$, $t < t'$, and for all $x \in \mathcal{X}$, $\theta_i(t', x) = \theta(t, x) + t' - t$.
2. For all $i < \eta - 1$ there exists a (possibly empty) set of clocks $X \subseteq \mathcal{X}$ such that $\theta_{i+1}(\alpha_i, \cdot) = \theta_i(\alpha_i, \cdot)[X := 0]$.

Remark 1. Note that for two successive elements of $\mathcal{I}$, $(S_i, [\alpha_{i-1}, \alpha_i])$, $(S_{i+1}, [\alpha_i, \alpha_{i+1}])$, intervals are adjacent.

The *weakly-monotonic time domain* of a trajectory $T = (\mathcal{I}, \theta)$ is the set $\mathcal{I}_{intv} = \bigcup_{1 \le i < \eta} \{i\} \times [\alpha_{i-1}, \alpha_i]$. Elements of $\mathcal{I}_{intv}$ will be called *weakly-monotonic time points* (or *w-points* for short). $\mathcal{I}_{intv}$ can be totally-ordered as usual: given $(n, t), (n', t') \in \mathcal{I}_{intv}$, we say that (n, t) *precedes* (n', t') and denote $(n, t) \prec (n', t')$ if either $n < n'$ or $n = n'$ and $t < t'$.

Remark 2. Note that, for some $1 \le i < \eta$, $\mathcal{I}_{intv}$ might contain only one element of the form (i, α). The intuition behind this is that the i-th state in the trajectory is transient, the system must pass through this state but it rests there for zero amount of time.

Given a trajectory $T = (\mathcal{I}, \theta)$ and some w-point $(n, \beta) \in \mathcal{I}_{intv}$, we may define the *suffix of T starting with* (n, β), denoted by $T[(n, \beta)..] = \big(\mathcal{I}[(n, \beta)..], \theta[(n, \beta)..]\big)$, as follows:

1. $\mathcal{I}[(n, \beta)..] = \big(S_{i+n-1}, [\alpha'_{i-1}, \alpha'_i]\big)_{1 \le i < \eta'}$ with $\eta' = \eta - n + 1$ and $\alpha'_i = \alpha_{i+n-1}$ for all i with $1 \le i < \eta'$.
2. $\theta[(n, \beta)..] = (\theta'_i)_{1 \le i < \eta'}$ with $\theta'_i(t, x) = \theta_{i+n-1}(t, x)$ for all i with $1 \le i < \eta'$.

Similarly, the *prefix of T upto* (n, β), denoted by $T[0..(n, \beta)] = \big(\mathcal{I}[0..(n, \beta)], \theta[0..(n, \beta)]\big)$, can be defined as follows:

1. $\mathcal{I}[0..(n, \beta)] = \big(S_i, [\alpha'_{i-1}, \alpha'_i]\big)_{1 \le i < n+1}$ with $\alpha'_i = \alpha_i$ for $i < n$, and $\alpha'_n = \beta$.
2. $\theta[0..(n, \beta)] = (\theta'_i)_{1 \le i \le n}$ is defined by $\theta'_i(t, x) = \theta_i(t, x)$ for all $1 \le i \le n$, $t \in [0, \alpha'_i]$.

Another useful operation is *resetting some clock z at some w-point* (k, β): given a trajectory $T = (\mathcal{I}, \theta)$ and some w-point $(k, \beta) \in \mathcal{I}_{intv}$, the trajectory $T[z := 0 \text{ at } (k, \beta)] = (\mathcal{I}', \theta')$ is defined as follows:

– $\mathcal{I}' = (S'_i, [\alpha'_{i-1}, \alpha'_i])_{1 \le i < \eta'}$ where $\eta' = \eta + 1$ and $S'_i = S_i$ for $i \le k$, $S'_i = S_{i-1}$ for $k < i < \eta'$ and $\alpha'_i = \alpha_i$ for $i < k$, $\alpha'_k = \beta$ and $\alpha'_i = \alpha_{i-1}$ for $k < i < \eta'$.

– $\theta' = (\theta'_i)_{1\le i<\eta'}$ is defined by

$$\theta'_i(t,z) = \begin{cases} \theta_i(t,z), \text{ for } i \le k \text{ and } (i,t) \prec (k,\beta) \\ \theta_{i-1}(t,z), \text{ if } k<i<\eta', \exists j \le i-1, \theta_j(t,z)=0 \\ \theta_{i-1}(t,z) - \theta_{i-1}(\beta,z), \text{ otherwise} \end{cases}$$

and, for $x \neq z$, $\theta'_i(t,x) = \theta'_i(t,x)$ for $i \le k$ and $(i,t) \prec (k,\beta)$, and $\theta'_i(t,x) = \theta'_{i-1}(t,x)$ otherwise.

Figure 1 gives an example of a trajectory T, and of $T[x := 0 \text{ at } (1,1)]$.

A trajectory is *initialized* if all the clocks are set to zero at the initial w-point, *i.e.*for all $x \in X$, $\alpha_1(0,x) = 0$.

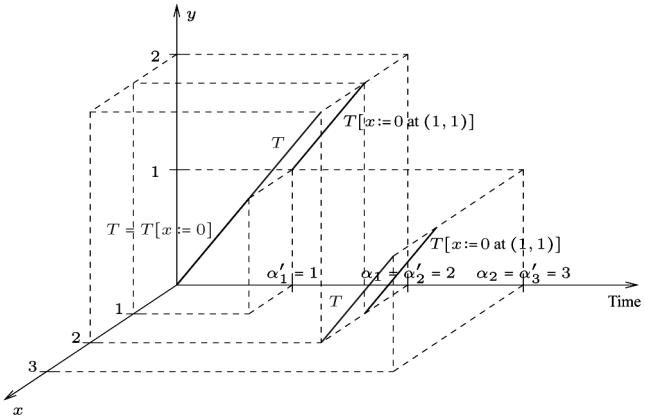

Fig. 1. Resetting clock x in T at (1,1).

4.2. **Trajectories and observability**

In this subsection, we give the observability relation on trajectories, which represents the essence of the epistemic part of the logic. This requires the definition of a projection operator on trajectories, which tells what are the clock values and state symbols that are observable by some agent along some trajectory.

Given a set of clocks $\mathcal{X}'$, a set of states Π', and a trajectory $T = (\mathcal{I}, \theta)$ with $\mathcal{I} = \big(S_i, [\alpha_{i-1}, \alpha_i]\big)_{1\le i<\eta}$ and $\theta = \big(\theta_i\big)_{1\le i<\eta}$, the **projection** of T **defined by** $(\Pi, \mathcal{X}')$ is the trajectory $T\big|_{\Pi,\mathcal{X}'} = (\mathcal{I}\big|_{\Pi,\mathcal{X}'}, \theta\big|_{\Pi,\mathcal{X}'})$ with $\mathcal{I}\big|_{\Pi,\mathcal{X}'} = \big(S_i \cap \Pi', [\alpha_{i-1}, \alpha_i]\big)_{1\le i<\eta}$ and $\theta\big|_{\Pi,\mathcal{X}'} = \big(\theta'_i\big)_{1\le i<\eta}$ where $\theta'_i(t,x) = \theta_i(t,x)$ for all $i < \eta, t \in [\alpha_{i-1}, \alpha_i]$, and $x \in \mathcal{X}$.

Our presentation of trajectories has the problem that some behavior can be presented by distinct trajectories: it is what we call the "stuttering" phenomenon. A trajectory is *stuttering-free* if either two consecutive intervals in

$\mathcal{I}$ are labeled with distinct state symbols or if some clock is reset in between the two intervals. Formally, a trajectory $T = (\mathcal{I}, \theta)$ is **stuttering-free** if for all $i < \eta - 1$, if $S_i \neq S_{i+1}$ then there exists $x \in X$ with $\theta_{i+1}(\alpha_i, x) = 0 \neq \theta_i(\alpha_i, x)$.

Stuttering-free trajectories represent "normal forms" for behaviors of timed automata: consider two trajectories $T_j = (\mathcal{I}^j, \theta^j)$ $(j = 1, 2)$, where $\mathcal{I}^j = \left(S_i^j, [\alpha_{i-1}^j, \alpha_i^j]\right)_{1 \leq i < \eta_j}$ and $\theta^j = \left(\theta_i^j\right)_{1 \leq i < \eta_j}$. Assume that T_2 is stuttering-free. We say that T_2 **represents** T_1 if there exists some surjective increasing function $\varphi : [1..\eta_1 - 1] \to [1..\eta_2 - 1]$ satisfying the following requirements:

- For all $i < j < \eta_1$, if $\varphi(i) = \varphi(j)$ then $S_i^1 = S_j^1 = S_{\varphi(i)}^2$.
- $\alpha_{\varphi(i)}^2 = \max\left\{\alpha_j^1 \mid \varphi(j) = \varphi(i)\right\}$.
- For all $i < \eta_1$ and $t \in [\alpha_{i-1}, \alpha_i]$, $\theta_{\varphi(i)}^2(t, x) = \theta_i^1(t, x)$.

We also say that two (arbitrary) trajectories T_1, T_2 are **identical**, denoted $T_1 \equiv T_2$, if there exists a third stuttering-free trajectory T_3 which represents both T_1 and T_2.

4.3. **A weakly-monotonic semantics of timed automata**

A *simple constraint* over $\mathcal{X}$ is a constraint of the form $x \in I$, where I is an interval with nonnegative integer bounds. For a simple constraint C and a clock valuation v, we denote as usual $v \vDash C$ if v satisfies the constraint C.

Definition 7. A **timed automaton** [3] is a tuple $\mathcal{A} = (Q, \mathcal{X}, \Pi, \delta, \lambda, Q_0)$ where Q is a finite set of *locations*, $\mathcal{X}$ is a finite set of *clocks*, Π is a finite set of *state labels*, $\lambda : Q \to 2^{\Pi}$ is the *state-labeling function*, $Q_0 \subseteq Q$ is the set of *initial* locations, and δ is a finite set of tuples called *transitions* of the form (q, C, X, q'), where $q, q' \in Q$, $X \subseteq \mathcal{X}$, and C is a conjunction of simple constraints over $\mathcal{X}$.

Our semantics of timed automata will be given in terms of runs, which are behaviors of timed automata along trajectories. This is just an extension of the classical notion of a run in a timed automaton, enhanced with the possibility to have "transient" locations (in which control stays for a zero amount of time) and adapted such that each time point be associated with the current location.

A *run* in a timed automaton $\mathcal{A} = (Q, \mathcal{X}, \Sigma, \delta, \lambda, Q_0)$ is then a pair $R = (\mathcal{I}, \rho)$, where

- $\mathcal{I}$ is a sequence of pairs of locations and intervals, $\mathcal{I} = \left(q_i, [\alpha_{i-1}, \alpha_i]\right)_{1 \leq i < \eta}$, where $\eta \in \mathbb{N} \cup \{\infty\}$, $q_i \subseteq Q$ and $\alpha_{i-1}, \alpha_i \in \mathbb{R}_{\geq 0}$ with $\alpha_{i-1} \leq \alpha_i$, and $\alpha_0 = 0$.
- $\rho = (\rho_i)_{1 \leq i < \eta}$ with $\rho_i : [\alpha_{i-1}, \alpha_i] \times \mathcal{X} \to \mathbb{R}_{\geq 0}$.

subject to the following constraints:

1. For all $1 \leq i < \eta$ and all $t, t' \in [\alpha_{i-1}, \alpha_i]$, $t < t'$, and for all $x \in \mathcal{X}$, $\theta_i(t', x) = \theta(t, x) + t' - t$.
2. For all $1 \leq i < \eta - 1$ there exists some transition $(q_i, C_i, X_i, q_{i+1}) \in \delta$ for which $\rho_i(\alpha_i, \cdot) \models C_i$ and $\rho_{i+1}(\alpha_i, \cdot) = \rho_i(\alpha_i, \cdot)[X_i := 0]$.

We say that the run R is **unbounded** (or **has an unbounded domain**) if $\eta = \infty$ or $\alpha_{\eta-1} = \infty$.

At second item of the enumeration above, we say that the transition (q_i, C_i, X_i, q_{i+1}) *is associated with the w-point* (i, α_i) – and also associated with $(i+1, \alpha_i)$, both being w-points in $\mathcal{I}_{intv}$ for the run R.

Prefix and postfix operators can also be defined on runs. Hence, given a run $R = (\mathcal{I}, \rho)$ and some w-point $(n, \beta) \in \mathcal{I}_{intv}$, the prefix of R upto (n, t) is denoted $R[0..(n, \beta)] = \big(\mathcal{I}[0..(n, \beta)], \rho[0..(n, \beta)]\big)$ and the suffix of R starting with (n, t) is denoted $R[(n, \beta)..] = \big(\mathcal{I}[(n, \beta)..], \rho[(n, \beta)..]\big)$.

Resetting some clock along a run, $R[z := 0 \text{ at } (k, \beta)]$, can also be defined as for trajectories. However the result might not always be a run: it is a run only when there exists a loop (q_k, C, X, q_k) which can be associated with the w-point (k, β) in the result $R[z := 0 \text{ at } (k, \beta)]$.

Given a run $R = (\mathcal{I}, \rho)$ with $\mathcal{I} = \big(q_i, [\alpha_{i-1}, \alpha_i]\big)_{1 \leq i < \eta}$, the **trajectory generated by** R is the trajectory $\overset{(}{\rightsquigarrow}_R) = (\mathcal{I}', \rho)$ where $\mathcal{I} = \big(\lambda(q_i), [\alpha_{i-1}, \alpha_i]\big)_{1 \leq i < \eta}$.

A run $R = (\mathcal{I}, \rho)$ with $\mathcal{I} = \big(q_i, [\alpha_{i-1}, \alpha_i]\big)_{1 \leq i < \eta}$ and $\rho = (\rho_i)_{1 \leq i < \eta}$ is *accepting* if the first location q_0 is an initial location and the initial clock valuation is $\alpha_1(0, x) = 0$ for all $x \in \mathcal{X}$. A trajectory T is *accepted* if it is associated with an accepting run. Note that in our timed automata, all locations are "accepting".

4.4. **TCTLK: syntax, semantics, model-checking**

The Timed Computational Tree Logic with knowledge operators and with synchronous and perfect recall semantics, denoted TCTLK, has the following syntax:

$$\phi ::= p \mid C \mid \phi \wedge \phi \mid \neg\phi \mid A\phi\mathcal{U}\,\phi \mid E\phi\mathcal{U}\,\phi \mid z \text{ in } \phi \mid K_a\phi,$$

where $p \in \Pi$, the set of atomic propositions, C is a simple constraint over the set of clocks $\mathcal{X}$, y is some clock in $\mathcal{X}$, and $a \in Ag$ an element of the finite set of agents Ag.

The operator z in ϕ is called the *freeze operator*. We denote CTLK the logic defined by formulas not containing the freeze operator.

The semantics of TCTLK is given in terms of runs in a timed automaton which is endowed with some partial observability relations, one for each *agent* a in a finite set of agents Ag. Each partial observability relation models what

is observable by some agent a and is induced by a subset of symbols $\Pi_a \subseteq \Pi$ and a subset of clocks $\mathcal{X}_a \subseteq \mathcal{X}$.

Formally, the semantics is given in terms of a multi-agent timed system, defined as follows:

Definition 8. A **timed system with agents in** Ag is a tuple $\mathcal{M} = (\mathcal{A}, \mathcal{Z}, (\Pi_a)_{a\in Ag}, (\mathcal{X}_a)_{a\in Ag})$ where

- $\mathcal{A}$ is a *state-labeled* timed automaton $\mathcal{A} = (Q, \mathcal{X}, \Pi, \delta, \lambda, Q_0)$, that is, with $\lambda : Q \to 2^{\Pi}$ and $\delta \subseteq Q \times Constr(\mathcal{X}) \times 2^{\mathcal{X}} \times Q$.
- $\mathcal{Z} \subseteq \mathcal{X}$ is a subset of clocks (called *freeze clocks*).
- For each $a \in Ag$, $\Pi_a \subseteq \Pi$, $\mathcal{X}_a \subseteq \mathcal{X}$.

It is also assumed that $\mathcal{A}$ contains, for each location $q \in Q$ and each clock $z \in \mathcal{Z}$, a loop $(q, \mathsf{true}, \{z\}, q)$.

The requirement that each location has a loop resetting a clock in $\mathcal{Z}$ does not represent a strong restriction: we may think that we are given a timed system $\mathcal{M}$ without the clocks $\mathcal{Z}$, and then we enhance $\mathcal{M}$ by appending clocks from $\mathcal{Z}$, such that the freeze quantifiers be interpretable.

For each agent $a \in Ag$, the **observability relation for** a is the following: given two runs R_1 and R_2 in $\mathcal{A}$, and some w-point (k, β) occurring in both, we say that R_1 and R_2 **cannot be distinguished by** a **upto** (k, β) if $\mathsf{traj}(R_1)[0..(k,\beta)]\big|_{\Pi_a,\mathcal{X}_a} \equiv \mathsf{traj}(R_2)[0..(k,\beta)]\big|_{\Pi_a,\mathcal{X}_a}$.

The semantics of TCTLK formulas is given in terms of tuples $(\mathcal{M}, R, k, \beta)$ consisting of an n-agent system $\mathcal{M}$, an *unbounded* run $R = (\mathcal{I}, \rho)$ in the underlying timed automaton of $\mathcal{M}$ and some w-point $(k, \beta) \in \mathcal{I}_{intv}$. The rules defining the semantics of TCTLK formulas are the following:

- $(\mathcal{M}, R, k, \beta) \vDash p$ if $p \in \lambda(q)$, q being the first location in $R[(k, \beta)..]$.
- $(\mathcal{M}, R, k, \beta) \vDash C$ if $v \vDash C$ where v is the clock valuation at (k, β) in R, $v = \rho_k(\beta, \cdot)$ for $R = (\mathcal{I}, \rho)$.
- $(\mathcal{M}, R, k, \beta) \vDash \phi_1 \wedge \phi_2$ if $(\mathcal{M}, R, k, \beta) \vDash \phi_1$ and $(\mathcal{M}, R, k, \beta) \vDash \phi_2$.
- $(\mathcal{M}, R, k, \beta) \vDash \neg\phi$ if $(\mathcal{M}, R, k, \beta) \nvDash \phi$.
- $(\mathcal{M}, R, k, \beta) \vDash z$ in ϕ if $(\mathcal{M}, R[z{:=}0 \text{ at } (k, \beta)], k, \beta) \vDash \phi$.
- $(\mathcal{M}, R, k, \beta) \vDash K_a\phi$ if for any run R' and any w-point (k', β) in R' with $\mathsf{traj}(R')[0..(k', \beta)]\big|_{\Pi_a,\mathcal{X}_a} \equiv \mathsf{traj}(R)[0..(k, \beta)]\big|_{\Pi_a,\mathcal{X}_a}$ we have that $(\mathcal{M}, R', k', \beta) \vDash \phi$.
- $(\mathcal{M}, R, k, \beta) \vDash E\phi_1 \mathcal{U} \phi_2$ if there exists some run $R' = (\mathcal{I}', \rho')$ for which $R[0..(k, \beta)] = R'[0..(k, \beta)]$ and there exists a w-point $(k', \beta') \in \mathcal{I}'_{intv}$ with $(k', \beta') \geq (k, \beta)$, $(\mathcal{M}, R', k', \beta') \vDash \phi_2$ and for all $(k'', \beta'') \in \mathcal{I}'_{intv}$ for which $(k'', \beta'') \prec (k', \beta')$, $(k'', \beta'') \geq (k, \beta)$, we have that $(\mathcal{M}, R', k'', \beta'') \vDash \phi_1$.
- $(\mathcal{M}, R, k, \beta) \vDash A\phi_1 \mathcal{U} \phi_2$ if for any run $R' = (\mathcal{I}', \rho')$ for which $R[0..(k, \beta)] =$

$R'[0..(k,\beta)]$, there exists a w-point $(k',\beta') \in \mathcal{I}'_{intv}$ with $(k',\beta') \geq (k,\beta)$, $(\mathcal{M}, R', k', \beta') \models \phi_2$ and for all $(k'',\beta'') \in \mathcal{I}'_{intv}$ for which $(k'',\beta'') \prec (k',\beta')$, $(k'',\beta'') \geq (k,\beta)$ we have that $(\mathcal{M}, R', k'', \beta'') \models \phi_1$.

The usual abbreviations apply here too, in particular

$$E \Diamond \phi = E\,\mathsf{true}\,\mathcal{U}\,\phi \qquad A \Box \phi = \neg E \Diamond \neg\phi$$
$$A \Diamond \phi = A\,\mathsf{true}\,\mathcal{U}\,\phi \qquad E \Box \phi = \neg A \Diamond \neg\phi$$
$$P_a\phi = \neg K_a \neg\phi$$

An example of a TCTLK formula is $P_a\varphi \wedge \neg\varphi$, with $\varphi = z$ in $E \Diamond (z \leq 3 \wedge danger)$. This might model a situation in which some sensor a might provoke a false alarm, as its observable state would indicate that it's possible that in less than 3 time units the system goes into a dangerous state, whereas this situation cannot occur in the current state.

The following theorems summarize the results from [18]:

THEOREM 5. 1. *The model-checking problem for TCTLK over continuous time domains is undecidable.*

2. *The model-checking problem for TCTLK with full observability of clock values is decidable.*

Here, by *full observability of clock values* we mean that formulas of TCTLK are interpreted over multi-agent systems $\mathcal{M} = (\mathcal{A}, \mathcal{Z}, (\Pi_a)_{a\in Ag}, (\mathcal{X}_a)_{a\in Ag})$ in which $\mathcal{X}_a = \mathcal{X}$ for all $a \in Ag$.

5. CONCLUSIONS

We have presented a number of results towards identifying the frontier of decidability for the satisfiability and the model-checking problems for combinations of temporal and epistemic logics. Unlike the classical case, undecidability occurs even for non-parametric cases, once the epistemic operators are assumed to be non-forgetting. The first interesting result is that a slight change in the semantics of the simplest combination, CTLK, which is natural in that it considers agent observations are based on subsets of atomic propositions and not "abstract" local states as considered classically, leads to undecidability. Then we have seen that most of the interesting cases of alternating temporal logics with non-forgetting semantics have an undecidable model-checking problem. And, finally, CTLK with a continuous time semantics has also an undecidable model-checking problem.

But all these results do not cover the entire topics related with the import of the classical duality between logics and automata [29, 59]. In this classical framework, Monadic Second Order Logic (MSO), Büchi/Muller/Rabin or

tree automata and games are shown to be equally expressive over linear and branching time domains, and various temporal logics can be characterized with fragments of MSO and subclasses of automata. The natural question would then be whether these results could be transferred to the case of temporal epistemic logics. We mention preliminary results from [21] which shows that ATL_{iR} expressivity cannot be captured by the μ-calculus of knowledge and time and the corresponding generalization of *jumping tree automata* (see also [8] for some preliminary results concerning the μ-calculus of knowledge and time). This suggests that expressiveness in temporal epistemic logics is more difficult to characterize.

REFERENCES

[1] Th. Agotnes, V. Goranko and W. Jamroga, *Alternating-time temporal logics with irrevocable strategies.* In: D. Samet (Ed.), Proceedings of TARK'07, 2007, pp. 15–24.

[2] R. Alur, C. Courcoubetis and D.L. Dill, *Model-checking in dense real-time.* Inf. Comput. **104** (1993), *1*, 2–34.

[3] R. Alur and D. Dill, *A theory of timed automata.* Theoret. Comput. Sci. **126** (1994), 183–235.

[4] R. Alur, T. Feder and Th. Henzinger, *The benefits of relaxing punctuality.* Journal of ACM **43** (1996), *1*, 116–146.

[5] R. Alur, Th. Henzinger and O. Kupferman, *Alternating-time temporal logic.* Proceedings of COMPOS **1536** of LNCS, 23–60. Springer Verlag, 1998.

[6] R. Alur, Th. Henzinger and O. Kupferman, *Alternating-time temporal logic.* Journal of the ACM **49** (2002), *5*, 672–713.

[7] Ch. Baier and J.-P. Katoen, *Principles of Model Checking.* The MIT Press, 2008.

[8] R. Bozianu, C. Dima and C. Enea, *Model-checking an epistemic μ-calculus with synchronous and perfect recall semantics.* Proceedings of the 14th Conference on Theoretical Aspects of Rationality and Knowledge (TARK'13), 2013, pp. 176–186.

[9] N. Bulling, J. Dix and W. Jamroga, *Model checking logics of strategic ability: Complexity.* In: M. Dastani, K.V. Hindriks and J.-J.C. Meyer (Eds.), *Specification and Verification of Multi-Agent Systems.* Springer, 2010, pp. 125–160.

[10] N. Bulling and W. Jamroga, *Model checking ATL* is harder than it seemed.* Technical Report IfI-09-13, Clausthal University of Technology, 2009.

[11] Dima C and F.L. Tiplea, *Model-checking ATL under imperfect information and perfect recall semantics is undecidable.* CoRR, abs/1102.4225, 2011.

[12] R. Chadha, S. Delaune and S. Kremer, *Epistemic logic for the applied pi calculus.* Proceedings of FMOODS/FORTE'09, **5522**, Lecture Notes in Comput. Sci., Springer, 2009, pp. 182–197.

[13] K. Chatterjee, L. Doyen, Th. Henzinger and J.-F. Raskin, *Algorithms for omega-regular games with imperfect information.* Proceedings of CSL'06, **4207**, Lecture Notes in Comput. Sci., Springer, 2006, pp. 287–302.

[14] E. Clarke, E. Emerson and A.P. Sistla, *Automatic verification of finite-state concurrent systems using temporal logic specifications.* ACM Trans. of Programming Languages and Systems **8** (1986), *2*, 244–263.

[15] E. Clarke, O. Grumberg and D. Peled, *Model Checking*. The MIT Press, 2000.

[16] C. Diaconu and C. Dima, *Model-checking alternating-time temporal logic with strategies based on common knowledge is undecidable*. Appl. Artificial Intelligence **26** (2012), 331–348.

[17] C. Dima, *Revisiting satisfiability and model-checking for CTLK with synchrony and perfect recall*. Proceedings of the 9th International Workshop on Computational Logic in Multi-Agent Systems (CLIMA IX) **5405**, LNAI, 2008, 117–131.

[18] C. Dima, *Positive and negative results on the decidability of the model-checking problem for an epistemic extension of timed ctl*. Proceedings of TIME'09, pp. 29–36. IEEE Computer Society, 2009.

[19] C. Dima, *Non-axiomatizability for linear temporal logic of knowledge with concrete observability*. J. of Logic Comput., 2010. To appear.

[20] C. Dima, C. Enea and D. Guelev, *Model-checking an alternating-time temporal logic with knowledge, imperfect information, perfect recall and communicating coalitions*. Electron. Proceedings in Theoret. Comput. Sci. **25** (2010), 103–117.

[21] C. Dima, B. Maubert and S. Pinchinat, *Relating paths in transition systems: The fall of the modal mu-calculus*. Proceedings of 40th International Symposium on Mathematical Foundations of Computer Science 2015 (MFCS), Part I, **9234**, Lecture Notes in Comput. Sci., Springer, 2015, pp. 179–191.

[22] Cl. Dixon, M.C. Fernandez Gago, M. Fisher and W. van der Hoek, *Temporal logics of knowledge and their applications in security*. Electron. Notes Theor. Comput. Sci. **186** (2007), 27–42.

[23] J. Elgaard, N. Klarlund and A. Møller, *MONA 1.x: New techniques for WS1S and WS2S*. Proceedings of CAV'98, **1427**, Lecture Notes in Comput. Sci., Springer, 1998, pp. 516–520.

[24] C. Elgot and M. Rabin, *Decidability and undecidability of extensions of second (first) order theory of (generalized) successor*. J. Symbolic Log. **31** (1966), *2*, 169–181.

[25] K. Engelhard, R. van der Meyden and Y. Moses, *Knowledge and the logic of local propositions*. Proceedings of TARK'98, Morgan Kaufman, 1998, pp. 29–41.

[26] R. Fagin, J. Halpem, Y. Moses and M. Vardi, *Reasoning about knowledge*. The MIT Press, 2004.

[27] P. Gammie and R. van der Meyden, *MCK: Model checking the logic of knowledge*. Proceedings of CAV'04, **3114**, Lecture Notes in Comput. Sci., Springer Verlag, 2004, pp. 479–483.

[28] V. Goranko and W. Jamroga, *Comparing semantics of logics for multi-agent systems*. Synthese **139** (2004), *2*, 241–280.

[29] E. Gradel, W. Thomas and Th. Wilke, *Automata, Logics and Infinite Games*. **2500** LNCS. Springer Verlag, 2002.

[30] D. Guelev and C. Dima, *Model-checking strategic ability and knowledge of the past of communicating coalitions*. Proceedings of DALT 2008, **5397**, Lecture Notes in Comput. Sci., Springer, 2008, pp. 75–90.

[31] D. Guelev, C. Dima and C. Enea, *An alternating-time temporal logic with knowledge, perfect recall and past: axiomatisation and model-checking*. J. Appl. Non-Class. Log., 2011, pp. 93–131.

[32] D.P. Guelev and C. Dima, *Epistemic atl with perfect recall, past and strategy contexts*. Proceedings of the 13th International Workshop Computational Logic in Multi-Agent Systems (CLIMA'12) **7486**, Lecture Notes in Comput. Sci., Springer, 2012, pp. 77–93.

[33] J. Halpem and R. Fagin, *A formal model of knowledge, action and communication in distributed systems: Preliminary report.* Proceedings of PODC'84, 1985, pp. 224–236.

[34] J. Halpem and K. O'Neill, *Anonymity and information hiding in multiagent systems.* J. Comput. Security **13** (2005), *3*, 483–512.

[35] J. Halpem, R. van der Meyden and M. Vardi, *Complete axiomatizations for reasoning about knowledge and time.* SIAM J. Comput. **33** (2004), *3*, 674–703.

[36] J. Halpem and M. Vardi, *The complexity of reasoning about knowledge and time: Extended abstract.* Proceedings of STOC, 1986, pp. 304–315.

[37] J. Halpem and M. Vardi, *The complexity of reasoning about knowledge and time. I. Lower bounds.* J. Comput. System Sci. **38** (1989), *1*, 195–237.

[38] W. Jamroga and Th. Ågotnes, *Constructive knowledge: What agents can achieve under imperfect information.* J. Appl. Non-Class. Log. **17** (2007), *4*, 423–475.

[39] W. Jamroga and W. van der Hoek, *Agents that know how to play.* Fund. lnform. **63** (2004), *2–3*, 185–219.

[40] H. Jonker and W. Pieters, *Receipt-freeness as a special case of anonymity in epistemic logic*, June 2006. LAVoSS Workshop On Trustworthy Elections (WOTE 2006), Cambridge.

[41] M. Kacprzak, A. Lomuscio, A. Niewiadomski, W. Penczek, Fr. Raimondi and M. Szreter, *Comparing BDD and SAT based techniques for model checking Chaum's Dining Cryptographers Protocol.* Fund. lnform. **72** (2006), *1–3*, 215–234.

[42] M. Kacprzak and W. Penczek, *Fully symbolic unbounded model checking for alternating-time temporal logic.* Autonomous Agents and Multi-Agent Systems **11** (2005), *1*, 69–89.

[43] R. Ladner and J. Reif, *The logic of distributed protocols.* Proceedings of TARK, Morgan Kaufmann, 1986, 207–222.

[44] Fr. Laroussinie, N. Markey and Gh. Oreiby, *On the expressiveness and complexity of ATL.* Log. Methods Comput. Sci. **4** (2008), *2*.

[45] A. Lomuscio and W. Penczek, *LDYIS: A framework for model checking security protocols.* Fund. Inform., **85** (2008), *1–4*, 359–375.

[46] A. Lomuscio and Fr. Raimondi, *The complexity of model checking concurrent programs against CTLK specifications.* Proceedings of DALT'06, **4327**, Lecture Notes in Comput. Sci., Springer, 2006, pp. 29–42.

[47] A. Lomuscio and Fr. Raimondi, *Mcmas: A model checker for multi-agent systems.* Proceedings of TACAS'2006, **3920**, Lecture Notes in Comput. Sci., Springer, 2006, pp. 450–454.

[48] A. Lomuscio and M. Ryan, *On the relation between interpreted systems and Kripke models.* Proceedings of AAMAS'97, **1441**, Lecture Notes in Comput. Sci., Springer, 1998, pp. 46–59.

[49] A. Lomuscio and B. Wozna, *A complete and decidable security-specialised logic and its application to the TESLA protocol.* Proceedings of the 5^{th} International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS), ACM, 2006, pp. 145–152.

[50] S. Nain and M.Y. Vardi, *Branching vs. linear time: Semantical perspective.* Proceedings of ATVA'07, **4762**, Lecture Notes in Comput. Sci., Springer Verlag, 2007, pp. 19–34.

[51] P. Pandya and D. V. Hung, *Duration calculus of weakly monotonic time.* Proceedings of FTRTFT, **1486**, Lecture Notes in Comput. Sci., Springer, 1998, pp. 55–64.

[52] Fr. Raimondi and A. Lomuscio, *Automatic verification of multi-agent systems by model checking via ordered binary decision diagrams.* J. Appl. Log. **5** (2005), *2*, 235–251.

[53] P.-Y. Schobbens, *Alternating-lime logic with imperfect recall.* Electron. Notes Theor. Comput. Sci. **85** (2004), *2*, 82–93.

[54] N. Shilov and N. Garanina, *Model checking knowledge and fixpoints.* Proceedings of FICS, Extended version available as Preprint 98, Ershov Institute of Informatics, Novosibirsk, 2002, pp. 25–39.

[55] K. Su, *Model checking temporal logics of knowledge in distributed systems.* Proceedings of AAAI'2004, AAAI Press/The MIT Press, 2004, pp. 98–103.

[56] K. Su, Q. Chen, A. Sattar, W. Yue, G. Lv and X. Zheng, *Verification of authentication protocols for epistemic goals via SAT compilation.* J. Comput. Sci. Tech. **21** (2006), *6*, 932–943.

[57] K. Su, G. Lv and Q. Chen, *Knowledge theoretic approach to formal verification of authentication protocols.* Sci. China Inf. Sci. **48** (2006), *4*, 513–532.

[58] W. Thomas, *Infinite trees and automation-definable relations over ω-words.* Theoret. Comput. Sci. **103** (1992), *1*, 143–159.

[59] W. Thomas, *Languages, automata and logic.* In: G. Rozenberg and A. Salomaa (Eds.), *Handbook of Formal Languages* **3**, Beyond Words, Springer Verlag, 1997, pp. 389–455.

[60] J. van Benthem and E. Pacuit, *The tree of knowledge in action: Towards a common perspective.* Proceedings of AiML, College Publications, 2006, pp. 87–106.

[61] W. van der Hoek, A. Lomuscio and M. Wooldridge, *On the complexity of practical ATL model checking.* Proceedings of AAMAS, ACM, 2006, pp. 201–208.

[62] W. van der Hoek and M. Wooldridge, *Cooperation, knowledge and time: Alternating-time temporal epistemic logic and its applications.* Studia Logica **75** (2003), *1*, 125–157.

[63] R. van der Meyden, *Common knowledge and update in finite environments.* Inform. and Comput. **140** (1998), *2*, 115–157.

[64] R. van der Meyden and N. Shilov, *Model checking knowledge and time in systems with perfect recall (extended abstract).* Proceedings of FSTTCS, **1738**, LNCS, 1999, 432–445.

[65] R. van der Meyden and K. Su, *Symbolic model checking the knowledge of the dining cryptographers.* Proceedings o f the 17$^{\text{th}}$ IEEE Computer Security Foundations Workshop, (CSFW-17), IEEE Computer Society, 2004.

[66] H. van Ditmarsch, W. van der Hoek, R. van der Meyden and Ji Ruan, *Model checking russian cards.* Electron. Notes Theor. Comput. Sci. **149** (2006), *2*, 105–123.

[67] S. van Otterloo, W. van der Hoek and M. Wooldridge, *Model checking a knowledge exchange scenario.* Appl. Artificial Intelligence **18** (2004), *9–10*, 937–952.

[68] B. Wozna and A. Lomuscio, *A logic for knowledge, correctness and real time.* Proceedings of CLIMA V, Lecture Notes in Comput. Sci. **3487**, Springer, 2005, pp. 1–15.

Received 1 August 2016

LACL, Université Paris Est-Créteil,
61 av. du G-ral de Gaulle,
94010 Créteil, France
dima@u-pec.fr