Satisfiability of ATL with strategy contexts

François Laroussinie
LIAFA – Univ. Paris Diderot & CNRS
francoisl@liafa.univ-paris-diderot.fr

Nicolas Markey
LSV – ENS Cachan & CNRS
markey@lsv.ens-cachan.fr

Various extensions of the temporal logic ATL have recently been introduced to express rich properties of multi-agent systems. Among these, ATL_{sc} extends ATL with *strategy contexts*, while Strategy Logic has *first-order quantification* over strategies. There is a price to pay for the rich expressiveness of these logics: model-checking is non-elementary, and satisfiability is undecidable.

We prove in this paper that satisfiability is decidable in several special cases. The most important one is when restricting to *turn-based* games. We prove that decidability also holds for concurrent games if the number of moves available to the agents is bounded. Finally, we prove that restricting strategy quantification to memoryless strategies brings back undecidability.

1 Introduction

Temporal logics are a convenient tool to reason about computerised systems, in particular in the setting of verification [Pnu77, CE82, QS82]. When systems are interactive, the models usually involve several agents (or players), and relevant properties to be checked often question the existence of *strategies* for these agents to achieve their goals. To handle these, *alternating-time temporal logic* was introduced, and its algorithmic properties were studied: model checking is PTIME-complete [AHK02], while satisfiability was settled EXPTIME-complete [WLWW06].

While model checking is tractable, ATL still suffers from a lack of expressiveness. Over the last five years, several extensions or variants of ATL have been developed, among which ATL *with strategy contexts* [BDLM09] and *Strategy Logic* [CHP07, MMV10]. The model-checking problem for these logics has been proved non-elementary [DLM10, DLM12], while satisfiability is undecidable, both when looking for finite-state or infinite-state models [MMV10, TW12]. Several fragments of these logics have been defined and studied, with the aim of preserving a rich expressiveness and at the same time lowering the complexity of the decision problems [WHY11, MMPV12, HSW13].

In this paper we prove that satisfiability is decidable (though with non-elementary complexity) for the full logic ATL_{sc} (and SL) in two important cases: first, when satisfiability is restricted to turn-based games (this solves a problem left open in [MMV10] for SL), and second, when the number of moves available to the players is bounded. We also consider a third variation, where quantification is restricted to *memoryless* strategies; in that setting, the satisfiability problem is proven undecidable, even for turn-based games.

Our results heavily rely on a tight connection between ATL_{sc} and QCTL [DLM12], the extension of CTL with quantification over atomic propositions. For instance, the QCTL formula $\exists p.\ \varphi$ states that it is possible to label the unwinding of the model under consideration with proposition p in such a way that φ holds. This labeling with additional proposition allows us to mark the strategies of the agents and the model-checking problem for ATL_{sc} can then be reduced to the model-checking problem for QCTL.

This work was partly supported by ERC Starting grant EQualIS (308087) and by European project Cassting (FP7-ICT-601148).

Gabriele Puppis, Tiziano Villa (Eds.): Fourth International Symposium on Games, Automata, Logics and Formal Verification
EPTCS 119, 2013, pp. 208–223, doi:10.4204/EPTCS.119.18

However, in this transformation, the resulting QCTL formula depends both on the ATL_{sc} formula to be checked and on the game where the formula is being checked. This way, the procedure does not extend to satisfiability, which is actually undecidable. We prove here that this difficulty can be overcome when considering turn-based games, or when the number of available moves is fixed. The satisfiability problem for ATL_{sc} is then reduced to the satisfiability problem for QCTL, which we proved decidable (with non-elementary complexity) in [LM13]. When restricting to memoryless strategies, a similar reduction to QCTL exists, but in a setting where the quantified atomic propositions directly label the model, instead of its unwinding. The satisfiability problem for QCTL under that semantics is undecidable [Fre01, LM13], and we adapt the proof of that result to show that satisfiability of ATL^0_{sc} (in which quantification is restricted to memoryless strategies) is also undecidable.

2 Definitions

2.1 ATL with strategy contexts

In this section, we define the framework of concurrent game structures, and define the logic ATL with strategy contexts. We fix once and for all a set AP of atomic propositions.

Definition 1. *A* Kripke structure $\mathscr{S}$ *is a* 3*-tuple* $\langle Q,R,\ell\rangle$ *where* Q *is a countable set of states,* $R\subseteq Q^2$ *is a total relation (*i.e.*, for all* $q\in Q$*, there is* $q'\in Q$ *s.t.* $(q,q')\in R$*) and* $\ell\colon Q\to 2^{\mathsf{AP}}$ *is a labelling function.*

A path in a Kripke structure $\mathscr{S}$ is a mapping $\rho\colon \mathbb{N}\to Q$ such that $(\rho(i),\rho(i+1))\in R$ for all i. We write $\mathsf{first}(\rho)=\rho(0)$. Given a path ρ and an integer i, the i-th suffix of ρ, is the path $\rho_{\geq i}\colon n\mapsto \rho(i+n)$; the i-th prefix of ρ, denoted $\rho_{\leq i}$, is the finite sequence made of the $i+1$ first state of ρ. We write $\mathsf{Exec}^{\mathsf{f}}(q)$ for the set of finite prefixes of paths (or *histories*) with first state q. We write $\mathsf{last}(\pi)$ for the last state of a history π. Given a history $\rho_{\leq i}$ and a path π such that $\mathsf{last}(\rho_{\leq i})=\mathsf{first}(\pi)$, the concatenation $\lambda=\rho_{\leq i}\cdot\pi$ is defined by $\lambda(j)=\rho(j)$ when $j\leq i$ and $\lambda(j)=\pi(j-i)$ when $j>i$.

Definition 2 ([AHK02])**.** *A* Concurrent Game Structure *(*CGS*)* $\mathscr{C}$ *is a* 7*-tuple* $\langle Q,R,\ell,\mathsf{Agt},\mathscr{M},\mathsf{Mov},\mathsf{Edge}\rangle$ *where:* $\langle Q,R,\ell\rangle$ *is a (possibly infinite-state) Kripke structure,* $\mathsf{Agt}=\{a_1,\ldots,a_p\}$ *is a finite set of* agents*,* $\mathscr{M}$ *is a non-empty set of moves,* $\mathsf{Mov}\colon Q\times\mathsf{Agt}\to\mathscr{P}(\mathscr{M})\smallsetminus\{\varnothing\}$ *defines the set of available moves of each agent in each state, and* $\mathsf{Edge}\colon Q\times\mathscr{M}^{\mathsf{Agt}}\to R$ *is a transition table associating, with each state* q *and each set of moves of the agents, the resulting transition departing from* q*.*

The size of a CGS $\mathscr{C}$ is $|Q|+|\mathsf{Edge}|$. For a state $q\in Q$, we write $\mathsf{Next}(q)$ for the set of all states reachable by the possible moves from q, and $\mathsf{Next}(q,a_j,m_j)$, with $m_j\in\mathsf{Mov}(q,a_j)$, for the restriction of $\mathsf{Next}(q)$ to possible transitions from q when player a_j plays move m_j. We extend Mov and Next to coalitions (*i.e.*, sets of agents) in the natural way. We say that a CGS is *turn-based* when each state q is controlled by a given agent, called the owner of q (and denoted $\mathsf{Own}(q)$). In other terms, for every $q\in Q$, for any two move vectors m and m' in which $\mathsf{Own}(q)$ plays the same move, it holds $\mathsf{Edge}(q,m)=\mathsf{Edge}(q,m')$ (which can be achieved by letting the sets $\mathsf{Mov}(q,a)$ be singletons for every $a\neq\mathsf{Own}(q)$).

A *strategy* for some player $a_i\in\mathsf{Agt}$ in a CGS $\mathscr{C}$ is a function f_i that maps any history to a possible move for a_i, *i.e.*, satisfying $f_i(\pi)\in\mathsf{Mov}(\mathsf{last}(\pi),a_i)$. A strategy f_i is *memoryless* if $f_i(\pi)=f_i(\pi')$ whenever $\mathsf{last}(\pi)=\mathsf{last}(\pi')$. A strategy for a coalition A is a mapping assigning a strategy to each agent in A. The set of strategies for A is denoted $\mathsf{Strat}(A)$. The *domain* $\mathsf{dom}(F_A)$ of $F_A\in\mathsf{Strat}(A)$ is A. Given a coalition B, the strategy $(F_A)_{|B}$ (resp. $(F_A)_{\smallsetminus B}$) denotes the restriction of F_A to the coalition $A\cap B$ (resp. $A\smallsetminus B$). Given two strategies $F\in\mathsf{Strat}(A)$ and $F'\in\mathsf{Strat}(B)$, we define $F\circ F'\in\mathsf{Strat}(A\bigcup B)$ as $(F\circ F')_{|a_j}(\rho)=F_{|a_j}(\rho)$ (resp. $F'_{|a_j}(\rho)$) if $a_j\in A$ (resp. $a_j\in B\smallsetminus A$).

Let ρ be a history. A strategy $F_A = (f_j)_{a_j \in A}$ for some coalition A induces a set of paths from ρ, called the *outcomes* of F_A after ρ, and denoted $\mathsf{Out}(\rho, F_A)$: an infinite path $\pi = \rho \cdot q_1 q_2 \ldots$ is in $\mathsf{Out}(\rho, F_A)$ if, and only if, writing $q_0 = \mathsf{last}(\rho)$, for all $i \geq 0$ there is a set of moves $(m_k^i)_{a_k \in \mathsf{Agt}}$ such that $m_k^i \in \mathsf{Mov}(q_i, a_k)$ for all $a_k \in \mathsf{Agt}$, $m_k^i = f_k(\pi_{|\rho|+i})$ if $a_k \in A$, and q_{i+1} is the unique element of $\mathsf{Next}(q_i, \mathsf{Agt}, (m_k^i)_{a_k \in \mathsf{Agt}})$. Also, given a history ρ and a strategy $F_A = (f_j)_{a_j \in A}$, the strategy F_A^ρ is the sequence of strategies $(f_j^\rho)_{a_j \in A}$ such that $f_j^\rho(\pi) = f_j(\rho \cdot \pi)$, assuming $\mathsf{last}(\rho) = \mathsf{first}(\pi)$.

We now introduce the extension of ATL with strategy contexts [BDLM09, DLM10]:

Definition 3. *Given a set of atomic propositions* AP *and a set of agents* Agt, *the syntax of* ATL^*_{sc} *is defined as follows (where p ranges over AP and A over* 2^{Agt}*):*

$$\mathsf{ATL}^*_{sc} \ni \varphi_{\mathsf{state}}, \psi_{\mathsf{state}} ::= p \mid \neg\,\varphi_{\mathsf{state}} \mid \varphi_{\mathsf{state}} \vee \psi_{\mathsf{state}} \mid \cdot\rangle A \langle\cdot\, \varphi_{\mathsf{state}} \mid \langle\cdot A \cdot\rangle\, \varphi_{\mathsf{path}}$$
$$\varphi_{\mathsf{path}}, \psi_{\mathsf{path}} ::= \varphi_{\mathsf{state}} \mid \neg\,\varphi_{\mathsf{path}} \mid \varphi_{\mathsf{path}} \vee \psi_{\mathsf{path}} \mid \mathbf{X}\,\varphi_{\mathsf{path}} \mid \varphi_{\mathsf{path}}\,\mathbf{U}\,\psi_{\mathsf{path}}.$$

That a (state or path) formula φ is satisfied at a position i of a path ρ of a CGS $\mathscr{C}$ under a strategy context $F \in \mathsf{Strat}(B)$ (for some coalition B), denoted $\mathscr{C}, \rho, i \models_F \varphi$, is defined as follows (omitting atomic propositions and Boolean operators):

$$\begin{array}{rcl}
\mathscr{C}, \rho, i \models_F \cdot\rangle A \langle\cdot\, \varphi_{\mathsf{state}} & \text{iff} & \mathscr{C}, \rho, i \models_{F_{\setminus A}} \varphi_{\mathsf{state}} \\
\mathscr{C}, \rho, i \models_F \langle\cdot A \cdot\rangle\, \varphi_{\mathsf{path}} & \text{iff} & \exists F_A \in \mathsf{Strat}(A).\ \forall \rho' \in \mathsf{Out}(\rho_{\leq i}, F_A \circ F).\ \mathscr{C}, \rho', i \models_{F_A \circ F} \varphi_{\mathsf{path}} \\
\mathscr{C}, \rho, i \models_F \mathbf{X}\,\varphi_{\mathsf{path}} & \text{iff} & \mathscr{C}, \rho, i+1 \models_F \varphi_{\mathsf{path}} \\
\mathscr{C}, \rho, i \models_F \varphi_{\mathsf{path}}\,\mathbf{U}\,\psi_{\mathsf{path}} & \text{iff} & \exists j \geq 0.\ \mathscr{C}, \rho, i+j \models_F \psi_{\mathsf{path}} \text{ and } \forall 0 \leq k < j.\ \mathscr{C}, \rho, i+k \models_F \varphi_{\mathsf{path}}
\end{array}$$

Notice how the (existential) strategy quantifier contains an implicit universal quantification over the set of outcomes of the selected strategies. Also notice that state formulas do not really depend on the selected path: indeed one can easily show that

$$\mathscr{C}, \rho, i \models_F \varphi_{\mathsf{state}} \quad \text{iff} \quad \mathscr{C}, \rho', j \models_{F'} \varphi_{\mathsf{state}}$$

where we assume $\rho(i) = \rho'(j)$ and where F and F' verifies: $F(\rho_{\leq i} \cdot \rho'') = F'(\rho'_{\leq j} \cdot \rho'')$ for any finite ρ'' starting in $\rho(i)$. In particular this is the case when the $\rho_{\leq i} = \rho'_{\leq j}$ and $F = F'$.

In the sequel we equivalently write $\mathscr{C}, \pi(0) \models_F \varphi_{\mathsf{state}}$ in place of $\mathscr{C}, \pi, 0 \models_F \varphi_{\mathsf{state}}$ when dealing with state formulas.

For convenience, in the following we allow the construct $\langle\cdot A \cdot\rangle\, \varphi_{\mathsf{state}}$, defining it as a shorthand for $\langle\cdot A \cdot\rangle \perp \mathbf{U}\, \varphi_{\mathsf{state}}$. We also use the classical modalities $\mathbf{F}$ and $\mathbf{G}$, which can be defined using $\mathbf{U}$. Also, $[\cdot A \cdot]\, \varphi_{\mathsf{path}} = \neg\, \langle\cdot A \cdot\rangle\, \neg\, \varphi_{\mathsf{path}}$ expresses that any A-strategy has at least one outcome where φ_{path} holds.

The fragment ATL_{sc} of ATL^*_{sc} is defined as usual, by restricting the set of path formulas to

$$\varphi_{\mathsf{path}}, \psi_{\mathsf{path}} ::= \neg\, \varphi_{\mathsf{path}} \mid \mathbf{X}\, \varphi_{\mathsf{state}} \mid \varphi_{\mathsf{state}}\, \mathbf{U}\, \psi_{\mathsf{state}}.$$

It was proved in [BDLM09] that ATL_{sc} is actually as expressive as ATL^*_{sc}. Moreover, for any given set of players, any ATL_{sc} formula can be written without using negation in path formulas, replacing for instance $\langle\cdot A \cdot\rangle\, \mathbf{G}\, \varphi$ with $\langle\cdot A \cdot\rangle\, \neg\, \langle\cdot \mathsf{Agt} \setminus (A \cup B) \cdot\rangle\, \mathbf{F}\, \neg\, \varphi$, where B is the domain of the context in which the formula is being evaluated. While this is not a generic equivalence (it depends on the context and on the set of agents), it provides a way of removing negation from any given ATL_{sc} formula.

2.2 Quantified CTL

In this section, we introduce QCTL, and define its *tree semantics*.

Definition 4. *Let Σ be a finite alphabet, and S be a (possibly infinite) set of directions. A* Σ-labelled S-tree *is a pair $\mathscr{T} = \langle T, l \rangle$, where $T \subseteq S^*$ is a non-empty set of finite words on S s.t. for any non-empty word $n = m \cdot s$ in T with $m \in S^*$ and $s \in S$, the word m is also in T; and $l \colon T \to \Sigma$ is a labelling function.*

The *unwinding* (or *execution tree*) of a Kripke structure $\mathscr{S} = \langle Q, R, \ell \rangle$ from a state $q \in Q$ is the 2^{AP}-labelled Q-tree $\mathscr{T}_{\mathscr{S}}(q) = \langle \mathsf{Exec}^{\mathsf{f}}(q), \ell_{\mathscr{T}} \rangle$ with $\ell_{\mathscr{T}}(q_0 \cdots q_i) = \ell(q_i)$. Note that $\mathscr{T}_{\mathscr{S}}(q) = \langle \mathsf{Exec}^{\mathsf{f}}(q), \ell_{\mathscr{T}} \rangle$ can be seen as an (infinite-state) Kripke structure where the set of states is $\mathsf{Exec}^{\mathsf{f}}(q)$, labelled according to $\ell_{\mathscr{T}}$, and with transitions $(m, m \cdot s)$ for all $m \in \mathsf{Exec}^{\mathsf{f}}(q)$ and $s \in Q$ s.t. $m \cdot s \in \mathsf{Exec}^{\mathsf{f}}(q)$.

Definition 5. *For $P \subseteq \mathsf{AP}$, two 2^{AP}-labelled trees $\mathscr{T} = \langle T, \ell \rangle$ and $\mathscr{T}' = \langle T', \ell' \rangle$ are* P-equivalent *(denoted by $\mathscr{T} \equiv_P \mathscr{T}'$) whenever $T = T'$, and $\ell(n) \cap P = \ell'(n) \cap P$ for any $n \in T$.*

In other terms, $\mathscr{T} \equiv_P \mathscr{T}'$ if $\mathscr{T}'$ can be obtained from $\mathscr{T}$ by modifying the labelling function of $\mathscr{T}$ for propositions not in P. We now define the syntax and semantics of QCTL^*:

Definition 6. *The syntax of* QCTL^* *is defined by the following grammar:*

$$\mathsf{QCTL}^* \ni \varphi_{\mathsf{state}}, \psi_{\mathsf{state}} ::= p \mid \neg\, \varphi_{\mathsf{state}} \mid \varphi_{\mathsf{state}} \vee \psi_{\mathsf{state}} \mid \mathbf{E}\varphi_{\mathsf{path}} \mid \mathbf{A}\varphi_{\mathsf{path}} \mid \exists p.\ \varphi_{\mathsf{state}}$$
$$\varphi_{\mathsf{path}}, \psi_{\mathsf{path}} ::= \varphi_{\mathsf{state}} \mid \neg\, \varphi_{\mathsf{path}} \mid \varphi_{\mathsf{path}} \vee \psi_{\mathsf{path}} \mid \mathbf{X}\, \varphi_{\mathsf{path}} \mid \varphi_{\mathsf{path}}\, \mathbf{U}\, \psi_{\mathsf{path}}.$$

QCTL^* is interpreted here over Kripke structures through their unwindings[1]: given a Kripke structure $\mathscr{S}$, a state q and a formula $\varphi \in \mathsf{QCTL}^*$, that φ holds at q in $\mathscr{S}$, denoted with $\mathscr{S}, q \models_t \varphi$, is defined by the truth value of $\mathscr{T}_{\mathscr{S}}(q) \models \varphi$ that uses the standard inductive semantics of CTL^* over trees extended with the following case:

$$\mathscr{T} \models \exists p.\varphi_{\mathsf{state}} \quad \text{iff} \quad \exists \mathscr{T}' \equiv_{\mathsf{AP}\setminus\{p\}} \mathscr{T} \text{ s.t. } \mathscr{T}' \models \varphi_{\mathsf{state}}.$$

Universal quantification over atomic propositions, denoted with the construct $\forall p.\ \varphi$, is obtained by dualising this definition. We refer to [LM13] for a detailed study of QCTL^* and QCTL. Here we just recall the following important properties of these logics. First note that QCTL is actually as expressive as QCTL^* (with an effective translation) [Fre01, DLM12]. Secondly model checking and satisfiability are decidable but non elementary. More precisely given a QCTL formula φ and a (finite) set of degrees $\mathscr{D} \subseteq \mathbb{N}$, one can build a tree automaton $\mathscr{A}_{\varphi,\mathscr{D}}$ recognizing the $\mathscr{D}$-trees satisfying φ. This provides a decision procedure for model checking as the Kripke structure $\mathscr{S}$ fixes the set $\mathscr{D}$, and it remains to check whether the unwinding of $\mathscr{S}$ is accepted by $\mathscr{A}_{\varphi,\mathscr{D}}$. For satisfiability the decision procedure is obtained by building a formula φ_2 from φ such that φ_2 is satisfied by some $\{1,2\}$-tree iff φ is satisfied by some finitely-branching tree. Finally it remains to notice that a QCTL formula is satisfiable iff it is satisfiable in a finitely-branching tree (as QCTL is as expressive as MSO) to get the decision procedure for QCTL satisfiability. By consequence we also have that a QCTL formula is satisfiable iff it is satisfied by a regular tree (corresponding to the unwinding of some finite Kripke structure).

3 From ATL_{sc} to QCTL

The main results of this paper concern the satisfiability problem for ATL_{sc}: given a formula in ATL_{sc}, does there exists a CGS $\mathscr{C}$ and a state q such that $\mathscr{C}, q \models_\varnothing \varphi$ (with empty initial context)? Before we

[1] Note that several semantics are possible for QCTL^* and the one we use here is usually called the *tree semantics*.

present these results in the next sections, we briefly explain how we reduce the model-checking problem for ATL_{sc} (which consists in deciding whether a given state q of a given CGS $\mathscr{C}$ satisfies a given ATL_{sc} formula φ) to the model-checking problem for QCTL. This reduction will serve as a basis for proving our main result.

3.1 Model checking

Let $\mathscr{C} = \langle Q, R, \ell, \mathsf{Agt}, \mathscr{M}, \mathsf{Mov}, \mathsf{Edge}\rangle$ be a finite-state CGS, with a finite set of moves $\mathscr{M} = \{m_1, \ldots, m_k\}$. We consider the following sets of fresh atomic propositions: $\mathsf{P}_Q = \{\mathsf{p}_q \mid q \in Q\}$, $\mathsf{P}^j_{\mathscr{M}} = \{\mathsf{m}^j_1, \ldots, \mathsf{m}^j_k\}$ for every $a_j \in \mathsf{Agt}$, and write $\mathsf{P}_{\mathscr{M}} = \bigcup_{a_j \in \mathsf{Agt}} \mathsf{P}^j_{\mathscr{M}}$. Let $\mathscr{S}_{\mathscr{C}}$ be the Kripke structure $\langle Q, R, \ell_+\rangle$ where for any state q, we have: $\ell_+(q) = \ell(q) \cup \{\mathsf{p}_q\}$. A strategy for an agent a_j can be seen as a function $f_j \colon \mathsf{Exec}^{\mathsf{f}}(q) \to \mathsf{P}^j_{\mathscr{M}}$ labeling the execution tree of $\mathscr{S}_{\mathscr{C}}$ with propositions in $\mathsf{P}^j_{\mathscr{M}}$.

Let $F \in \mathsf{Strat}(C)$ be a strategy context and $\Phi \in \mathsf{ATL}_{sc}$. We reduce the question whether $\mathscr{C}, q \models_F \Phi$ to a model-checking instance for QCTL^* over $\mathscr{S}_{\mathscr{C}}$. For this, we define a QCTL^* formula $\overline{\Phi}^C$ inductively: for non-temporal formulas,

$$\overline{\cdot\rangle A\langle\cdot\, \varphi}^C = \overline{\varphi}^{C\setminus A} \qquad \overline{\varphi \wedge \psi}^C = \overline{\varphi}^C \wedge \overline{\psi}^C \qquad \overline{\neg\psi}^C = \neg\,\overline{\varphi}^C \qquad \overline{p}^C = p$$

For a formula of the form $\langle\cdot A\cdot\rangle\, \mathbf{X}\, \varphi$ with $A = \{a_{j_1}, \ldots, a_{j_l}\}$, we let:

$$\overline{\langle\cdot A\cdot\rangle\, \mathbf{X}\, \varphi}^C = \exists \mathsf{m}^{j_1}_1 \ldots \mathsf{m}^{j_1}_k \ldots \mathsf{m}^{j_l}_1 \ldots \mathsf{m}^{j_l}_k.\ \bigwedge_{a_j \in A} \mathbf{AG}\left(\Phi_{\mathsf{strat}}(a_j)\right) \wedge \mathbf{A}\left(\Phi^{[C\cup A]}_{\mathsf{out}} \Rightarrow \mathbf{X}\, \overline{\varphi}^{C\cup A}\right)$$

where:

$$\Phi_{\mathsf{strat}}(a_j) = \bigvee_{q \in Q} \Big(\mathsf{p}_q \wedge \bigvee_{m_i \in \mathsf{Mov}(q, a_j)} (\mathsf{m}^j_i \wedge \bigwedge_{l \neq i} \neg\, \mathsf{m}^j_l)\Big)$$

$$\Phi^{[A]}_{\mathsf{out}} = \mathbf{G}\left[\bigwedge_{\substack{q \in Q \\ m \in \mathsf{Mov}(q,A)}} \Big((\mathsf{p}_q \wedge m) \Rightarrow \mathbf{X}\big(\bigvee_{q' \in \mathsf{Next}(q,A,m)} \mathsf{p}_{q'}\big)\Big)\right]$$

where m is a move $(m^j)_{a_j \in A} \in \mathsf{Mov}(q, A)$ for A and P_m is the propositional formula $\bigwedge_{a_j \in A} m^j$ characterizing m. Formula $\Phi_{\mathsf{strat}}(a_j)$ ensures that the labelling of propositions m^j_i describes a feasible strategy for a_j. Formula $\Phi^{[A]}_{\mathsf{out}}$ characterizes the outcomes of the strategy for A that is described by the atomic propositions in the model. Note that $\Phi^{[A]}_{\mathsf{out}}$ is based on the transition table Edge of $\mathscr{C}$ (via $\mathsf{Next}(q, A, m)$). For a formula of the form $\langle\cdot A\cdot\rangle\, (\varphi\, \mathbf{U}\, \psi)$ with $A = \{a_{j_1}, \ldots, a_{j_l}\}$, we let:

$$\overline{\langle\cdot A\cdot\rangle\, (\varphi\, \mathbf{U}\, \psi)}^C = \exists \mathsf{m}^{j_1}_1 \ldots \mathsf{m}^{j_1}_k \ldots \mathsf{m}^{j_l}_1 \ldots \mathsf{m}^{j_l}_k.\ \bigwedge_{a_j \in A} \mathbf{AG}\left(\Phi_{\mathsf{strat}}(a_j)\right) \wedge \mathbf{A}\left(\Phi^{[C\cup A]}_{\mathsf{out}} \Rightarrow (\overline{\varphi}^{C\cup A}\, \mathbf{U}\, \overline{\psi}^{C\cup A})\right)$$

Then:

Theorem 7. *[DLM12] Let q be a state in a CGS $\mathscr{C}$. Let Φ be an ATL_{sc} formula and F be a strategy context for some coalition C. Let $\mathscr{T}'$ be the execution tree $\mathscr{T}_{\mathscr{S}_{\mathscr{C}}}(q)$ with a labelling function ℓ' s.t. for every $\pi \in \mathsf{Exec}^{\mathsf{f}}(q)$ of length i and any $a_j \in C$, $\ell'(\pi) \cap \mathsf{P}^j_{\mathscr{M}} = \mathsf{m}^j_i$ if, and only if, $F(\pi)_{|a_j} = m_i$. Then $\mathscr{C}, q \models_F \Phi$ if, and only if, $\mathscr{T}', q \models_t \overline{\Phi}^C$.*

Combined with the (non-elementary) decision procedure for QCTL^* model checking, we get a model-checking algorithm for model checking ATL_{sc}. Notice that our reduction above is into QCTL^*, but as explained before every QCTL^* formula can be translated into QCTL. Finally note that model checking is non elementary (k-EXPTIME-hard for any k) both for QCTL and ATL_{sc} [DLM12].

3.2 Satisfiability

We now turn to satisfiability. The reduction to QCTL we just developed for model checking does not extend to satisfiability, because the QCTL formula we built depends both on the formula and on the structure. Actually, satisfiability is undecidable for ATL_{sc}, both for infinite CGS and when restricting to finite CGS [TW12]. It is worth noticing that both problems are relevant, as ATL_{sc} does not have the finite-model property (nor does it have the finite-branching property). This can be derived from the fact that the modal logic $S5^n$ does not have the finite-model property [Kur02], and from the elegant reduction of satisfiability of $S5^n$ to satisfiability of ATL_{sc} given in [TW12] [2].

In what follows, we prove decidability of satisfiability in two different settings: first in the setting of turn-based games, and then in the setting of a bounded number of actions allowed to the players. A consequence of our decidability proofs is that in both cases (based on automata constructions), ATL_{sc} does have the finite-model property (thanks to Rabin's regularity theorem). We also consider the setting where quantification is restricted to memoryless strategies, but prove that then satisfiability is undecidable (even on turn-based games and with a fixed number of actions).

Before we proceed to the algorithms for satisfiability, we prove a generic result [3] about the number of agents needed in a CGS to satisfy a formula involving a given set of agents. This result has already been proved for ATL (*e.g.* in [WLWW06]). Given a formula $\Phi \in \mathsf{ATL}_{sc}$, we use $\mathsf{Agt}(\Phi)$ to denote the set of agents involved in the strategy quantifiers in Φ.

Proposition 8. *An ATL_{sc} formula Φ is satisfiable iff, it is satisfiable in a CGS with $|\mathsf{Agt}(\Phi)|+1$ agents.*

Proof. Assume Φ is satisfied in a CGS $\mathscr{C} = \langle Q, R, \ell, \mathsf{Agt}, \mathscr{M}, \mathsf{Mov}, \mathsf{Edge}\rangle$. If $|\mathsf{Agt}| \leq \mathsf{Agt}(\Phi)$, one can easily add extra players in $\mathscr{C}$ in such a way that they play no role in the behavior of the game structure. Otherwise, if $|\mathsf{Agt}| > \mathsf{Agt}(\Phi)+1$, we can replace the agents in Agt that do not belong to $\mathsf{Agt}(\Phi)$ by a unique agent mimicking the action of the removed players. For example, a coalition $A = \{a_1, \ldots, a_k\}$ can be replaced by a player a whose moves are k-tuples in $\mathscr{M}^k$. □

4 Turn-based case

Let Φ be an ATL_{sc} formula, and assume $\mathsf{Agt}(\Phi)$ is the set $\{a_1, \ldots, a_n\}$. Following Prop. 8, let Agt be the set of agents $\mathsf{Agt}(\Phi) \cup \{a_0\}$, where a_0 is an additional player. In the following, we use an atomic propositions $(\mathsf{turn}_j)_{a_j \in \mathsf{Agt}}$ to specify the owner of the states. A strategy for an agent a_j can be encoded by an atomic proposition mov_j: indeed it is sufficient to mark one *successor* of every a_j-state (notice that this is a crucial difference with CGS). The outcomes of such a strategy are the runs in which every a_j-state is followed by a state labelled with mov_j; this is the main idea of the reduction below.

Given a coalition C (which we intend to represent the agents that have a strategy in the current context), we define a QCTL^* formula $\widehat{\Phi}^C$ inductively:

- for non-temporal formulas we let:

$$\widehat{\cdot\rangle A\langle\cdot\,\varphi}^C = \widehat{\varphi}^{C\smallsetminus A} \qquad \widehat{\varphi\wedge\psi}^C = \widehat{\varphi}^C \wedge \widehat{\psi}^C \qquad \widehat{\neg\psi}^C = \neg\,\widehat{\varphi}^C \qquad \widehat{P}^C = P$$

- for path formulas, we define:

$$\widehat{\mathbf{X}\varphi}^C = \mathbf{X}\,\widehat{\varphi}^C \qquad \widehat{\varphi\,\mathbf{U}\,\psi}^C = \widehat{\varphi}^C\,\mathbf{U}\,\widehat{\psi}^C$$

[2] Indeed the finite-branching property for ATL_{sc} would imply the finite-model property for $S5^n$.
[3] Note that it still holds true when restricting to turn-based games.

- for formulas of the form $\langle\cdot A\cdot\rangle\,\varphi$ with $A=\{a_{j_1},\ldots,a_{j_l}\}$, we let:

$$\widehat{\langle\cdot A\cdot\rangle\,\varphi}^{C} = \exists \mathsf{mov}_{j_1}\ldots\mathsf{mov}_{j_l}.\left[\mathbf{AG}\bigwedge_{a_j\in A}(\mathsf{turn}_j\Rightarrow \mathbf{EX}_1\,\mathsf{mov}_j)\wedge \mathbf{A}\Big[\mathbf{G}\Big(\bigwedge_{a_j\in A\cup C}(\mathsf{turn}_j\Rightarrow \mathbf{X}\,\mathsf{mov}_j)\Big)\Rightarrow\widehat{\varphi}^{C\cup A}\Big]\right]$$

where $\mathbf{EX}_1\,\alpha$ is a shorthand for $\mathbf{EX}\,\alpha\wedge\forall p.\Big(\mathbf{EX}\,(\alpha\wedge p)\Rightarrow\mathbf{AX}\,(\alpha\Rightarrow p)\Big)$, specifying the existence of a unique successor satisfying α.

Now we have the following proposition, whose proof is done by structural induction over the formula:

Proposition 9. *Let $\Phi\in\mathsf{ATL}_{sc}$, and $\mathsf{Agt}=\mathsf{Agt}(\Phi)\cup\{a_0\}$ as above. Let $\mathscr{C}$ be a turn-based CGS, q be a state of $\mathscr{C}$, and F be a strategy context. Let $\mathscr{T}_{\mathscr{C}}(q)=\langle T,\ell\rangle$ be the execution tree of the underlying Kripke structure of $\mathscr{C}$ (including a labelling with propositions $(\mathsf{turn}_j)_{a_j\in\mathsf{Agt}}$). Let ℓ_F be the labelling extending ℓ such that for every node ρ of T belonging to some $a_j\in\mathsf{dom}(F)$ (i.e., such that $\mathsf{last}(\rho)\in Own(a_j)$), its successor $\rho\cdot q$ according to F (i.e., such that $F_j(\rho)=q$) is labelled with mov_j. Then we have:*

$$\mathscr{C},q\models_F\Phi\quad\textit{iff}\quad\langle T,\ell_F\rangle\models\widehat{\Phi}^{\mathsf{dom}(F)}$$

Proof. The proof is by structural induction over Φ. The cases of atomic propositions and Boolean operators are straightforward.

- $\Phi=\langle\cdot A\cdot\rangle\,(\varphi\,\mathbf{U}\,\psi)$: assume $\mathscr{C},q\models_F\Phi$. Then there exists $F_A\in\mathsf{Strat}(A)$ s.t. for any $\rho\in\mathsf{Out}(q,F_A\circ F)$, there exists $i\geq 0$ s.t. $\mathscr{C},\rho(i)\models_{(F_A\circ F)^{\rho_{\leq i}}}\psi$ and $\forall 0\leq j<i$, we have $\mathscr{C},\rho(j)\models_{(F_A\circ F)^{\rho_{\leq j}}}\varphi$. Let $\ell_{F_A\circ F}$ be the extension of ℓ labelling T with propositions $(\mathsf{mov}_j)_{a_j\in\mathsf{Agt}}$ according to the strategy context $F_A\circ F$. By induction hypothesis, the following two statements hold true:
  - $\langle T,\ell_{F_A\circ F}\rangle_{\rho_{\leq i}}\models\widehat{\psi}^{\mathsf{dom}(F)\cup A}$, and
  - $\langle T,\ell_{F_A\circ F}\rangle_{\rho_{\leq j}}\models\widehat{\varphi}^{\mathsf{dom}(F)\cup A}$ for any $0\leq j<i$.

  (where $\langle U,l\rangle_\pi$ is the subtree of $\langle U,l\rangle$ rooted at node $\pi\in U$). As this is true for every ρ in the outcomes induced by $F_A\circ F$, it holds for every path in the execution tree satisfying the constraint over the labelling of $(\mathsf{turn}_j)_{a_j\in\mathsf{Agt}}$ and $(\mathsf{mov}_j)_{a_j\in\mathsf{Agt}}$. It follows that

$$\langle T,\ell_{F_A\circ F}\rangle\models\mathbf{A}\Big[\mathbf{G}\Big(\bigwedge_{a_j\in A\cup C}(\mathsf{turn}_j\Rightarrow\mathbf{X}\,\mathsf{mov}_j)\Big)\Rightarrow\widehat{\varphi}^{\mathsf{dom}(F)\cup A}\Big]$$

  Moreover we also know that $\mathbf{AG}\bigwedge_{a_j\in A}(\mathsf{turn}_j\Rightarrow\mathbf{EX}_1\mathsf{mov}_j)$ holds true in $\langle T,\ell_{F_A\circ F}\rangle$ since the labelling $\ell_{F_A\circ F}$ includes the strategy F_A. Hence $\langle T,\ell_F\rangle\models\widehat{\Phi}^{\mathsf{dom}(F)}$, with the labelling for $(\mathsf{mov}_j)_{a_j\in A}$ being obtained from F_A.

  Now assume $\langle T,\ell_F\rangle\models\widehat{\Phi}^{\mathsf{dom}(F)}$. Write $A=\{a_{j_1},\ldots,aj_l\}$. Then we have:

$$\langle T,\ell_F\rangle\models\exists\mathsf{mov}_{j_1}\ldots\mathsf{mov}_{j_l}.\Big[\mathbf{AG}\bigwedge_{a_j\in A}(\mathsf{turn}_j\Rightarrow\mathbf{EX}_1\mathsf{mov}_j)\wedge \mathbf{A}\Big[\mathbf{G}\Big(\bigwedge_{a_j\in A\cup C}(\mathsf{turn}_j\Rightarrow\mathbf{X}\,\mathsf{mov}_j)\Big)\Rightarrow(\widehat{\varphi}^{\mathsf{dom}(F)\cup A}\,\mathbf{U}\,\widehat{\psi}^{\mathsf{dom}(F)\cup A})\Big]\Big]$$

The first part of the formula, namely $\mathbf{AG} \bigwedge_{a_j \in A}(\mathsf{turn}_j \Rightarrow \mathbf{EX}_1 \mathsf{mov}_j)$, ensures that the labeling with $(\mathsf{mov}_j)_{a_j \in A}$ defines a strategy for the coalition A. The second part states that every run belonging to the outcomes of $F_A \circ F$ (remember that ℓ_F already contains the strategy context F) satisfies $(\widehat{\varphi}^{\mathsf{dom}(F)\cup A}\, \mathbf{U}\, \widehat{\psi}^{\mathsf{dom}(F)\cup A})$. Finally it remains to use the induction hypothesis over states along the execution to deduce $\mathscr{C}, q \models_F \langle\cdot A\cdot\rangle\, (\varphi\, \mathbf{U}\, \psi)$.

- $\Phi = \cdot\rangle A \langle\cdot\, \psi$: assume $\mathscr{C}, q \models_F \Phi$. Then $\mathscr{C}, q \models_{F_{\mathsf{dom}(F)\setminus A}} \psi$. Applying the induction hypothesis, we get $\langle T, \ell_{F_{\mathsf{dom}(F)\setminus A}}\rangle \models \widehat{\psi}^{\mathsf{dom}(F)\setminus A}$. And it follows that $\langle T, \ell_F\rangle \models \widehat{\psi}^{\mathsf{dom}(F)\setminus A}$ because the labeling of strategies for coalition A in F is not used for evaluating $\widehat{\psi}^{\mathsf{dom}(F)\setminus A}$. Conversely, assume $\langle T, \ell_F\rangle \models \widehat{\psi}^{\mathsf{dom}(F)\setminus A}$. Then we have $\langle T, \ell_{F_{\mathsf{dom}(F)\setminus A}}\rangle \models \widehat{\psi}^{\mathsf{dom}(F)\setminus A}$ (again the labeling of A strategies in F is not used for evaluating the formula). Applying induction hypothesis, we get $\mathscr{C}, q \models_{F_{\mathsf{dom}(F)\setminus A}} \psi$ and then $\mathscr{C}, q \models_F \Phi$.
- $\Phi = \langle\cdot A\cdot\rangle\, \mathbf{X}\, \varphi$ and $\Phi = \cdot\rangle A \langle\cdot\, \mathbf{X}\, \varphi$: the proofs are similar to the previous ones. □

Finally, let Φ_{tb} be the following formula, used to make the game turn-based:

$$\Phi_{tb} = \mathbf{AG}\Big[\bigvee_{a_j \in \mathsf{Agt}} \Big(\mathsf{turn}_j \wedge \bigwedge_{a_l \neq a_j} \neg \mathsf{turn}_l\Big)\Big]$$

and let $\widetilde{\Phi}$ be the formula $\Phi_{tb} \wedge \widehat{\Phi}^{\varnothing}$. Then we have:

Theorem 10. *Let Φ be an ATL_{sc} formula and $\widetilde{\Phi}$ be the QCTL^* formula defined as above. Φ is satisfiable in a turn-based CGS if, and only if, $\widetilde{\Phi}$ is satisfiable (in the tree semantics).*

Proof. If Φ is satisfiable in a turn-based structure, then there exists such a structure $\mathscr{C}$ with $|\mathsf{Agt}(\Phi)|+1$ agents. Assume $\mathscr{C}, q \models \Phi$. Now consider the execution tree $\mathscr{T}_{\mathscr{C}}(q)$ with the additional labelling to mark states with the correct propositions $(\mathsf{turn}_j)_{a_j \in \mathsf{Agt}}$, indicating the owner of each state. From Proposition 9, we have $\mathscr{T}_{\mathscr{C}}(q) \models \widehat{\Phi}^{\varnothing}$. Thus clearly $\mathscr{T}_{\mathscr{C}}(q) \models \widetilde{\Phi}$.

Conversely assume $\mathscr{T} \models \widetilde{\Phi}$. As explained in Section 2, we can assume that $\mathscr{T}$ is regular. Thus $\mathscr{T} \models \Phi_{tb} \wedge \widehat{\Phi}^{\varnothing}$: the first part of the formula ensures that every state of the underlying Kripke structure can be assigned to a unique agent, hence defining a turn-based CGS. The second part ensures that Φ holds for the corresponding game, thanks to Proposition 9. □

The above translation from ATL_{sc} into QCTL^* transforms a formula with k strategy quantifiers into a formula with at most $k+1$ nested blocks of quantifiers; satisfiability of a QCTL^* formula with $k+1$ blocks of quantifiers is in $(\mathsf{k}+3)$-$\mathsf{EXPTIME}$ [LM13]. Hence the algorithm has non-elementary complexity. We now prove that this high complexity cannot be avoided:

Theorem 11. *Satisfiability of ATL_{sc} formulas in turn-based CGS is non-elementary (i.e., it is* k-$\mathsf{EXPTIME}$*-hard, for all k).*

Proof (sketch). Model checking ATL_{sc} over turn-based games is non-elementary [DLM12], and it can easily be encoded as a satisfiability problem. Let $\mathscr{C} = \langle Q, R, \ell, \mathsf{Agt}, \mathscr{M}, \mathsf{Mov}, \mathsf{Edge}\rangle$ be a turn-based CGS, and Φ be an ATL_{sc} formula. Let P_q be a fresh atomic proposition for every $q \in Q$. Now we define an ATL_{sc} formula $\Psi_{\mathscr{C}}$ to describe the game $\mathscr{C}$ as follows:

$$\Psi_{\mathscr{C}} = \mathbf{AG}\Big(\bigvee_{q \in Q} (\mathsf{P}_q \wedge \bigwedge_{q' \neq q} \neg \mathsf{P}_{q'} \wedge \bigwedge_{P \in \ell(q)} P \wedge \bigwedge_{P' \notin \ell(q)} \neg P')\Big) \wedge$$
$$\mathbf{AG}\Big[\bigvee_{q \in Q} \Big(\mathsf{P}_q \Rightarrow (\bigwedge_{q \to q'} \langle\!\langle \mathsf{Own}(q) \rangle\!\rangle\, \mathbf{X}\, \mathsf{P}_{q'} \wedge \bigwedge_{q'.\ q \not\to q'} \neg \langle\!\langle \mathsf{Own}(q) \rangle\!\rangle\, \mathbf{X}\, \mathsf{P}_{q'})\Big)\Big]$$

where $q \to q'$ denotes the existence of a transition from q to q' in $\mathscr{C}$. Any turn-based CGS satisfying $\Psi_{\mathscr{C}}$ corresponds to some unfolding of $\mathscr{C}$, and then has the same execution tree. Finally we clearly have that $\mathscr{C}, q \models \Phi$ if, and only if, $\Psi_{\mathscr{C}} \wedge \mathsf{P}_q \wedge \Phi$ is satisfiable in a turn-based structure. $\square$

5 Bounded action alphabet

We consider here another setting where the reduction to QCTL^* can be used to solve the satisfiability of ATL_{sc}: we assume that each player has a bounded number of available actions. Formally, it corresponds to the following satisfiability problem:

Problem: $(\mathsf{Agt}, \mathscr{M})$**-satisfiability**
Input: a finite set of moves $\mathscr{M}$, a set of agents Agt, and an ATL_{sc} formula Φ involving the agents in Agt;
Question: does there exist a CGS $\mathscr{C} = \langle Q, R, \ell, \mathsf{Agt}, \mathscr{M}, \mathsf{Mov}, \mathsf{Edge}\rangle$ and a state $q \in Q$ such that $\mathscr{C}, q \models \Phi$.

Assume $\mathscr{M} = \{1, \ldots, \alpha\}$ and $\mathsf{Agt} = \{a_1, \ldots, a_n\}$. With this restriction, we know that we are looking for a CGS whose execution tree has nodes with degrees in the set $\mathscr{D} = \{1, 2, \ldots, \alpha^n\}$. We consider such $\mathscr{D}$-trees where the transition table is encoded as follows: for every agent a_i and move m in $\mathscr{M}$, we use the atomic proposition mov_i^m to specify that agent a_i has played move m in the *previous* node. Any execution tree of such a CGS satisfies formula

$$\Phi_{\mathsf{Edge}} = \mathbf{AG}\Big[\Big(\bigwedge_{\bar{m} \in \mathscr{M}^n} \mathbf{EX}_1 \mathsf{mov}^{\bar{m}}\Big) \wedge \mathbf{AX}\Big(\bigvee_{\bar{m} \in \mathscr{M}^n} \mathsf{mov}^{\bar{m}}\Big)\Big]$$

where $\mathsf{mov}^{\bar{m}}$ stands for $\bigwedge_{a_j \in \mathsf{Agt}} \mathsf{mov}_j^{\bar{m}_j}$. Notice that the second part of the formula is needed because of the way we handle the *implicit* universal quantification associated with the strategy quantifiers of ATL_{sc}.

Given a coalition C, we define a QCTL^* formula $\widehat{\Phi}^C$ inductively as follows:

- for non-temporal formulas we let

$$\widehat{\cdot\rangle A \langle\cdot\, \varphi}^{C} = \widehat{\varphi}^{C \smallsetminus A} \qquad \widehat{\varphi \wedge \psi}^{C} = \widehat{\varphi}^{C} \wedge \widehat{\psi}^{C} \qquad \widehat{\neg \psi}^{C} = \neg\, \widehat{\varphi}^{C} \qquad \widehat{P}^{C} = P$$

- for temporal modalities, we define

$$\widehat{\mathbf{X}\varphi}^{C} = \mathbf{X}\,\widehat{\varphi}^{C} \qquad \widehat{\varphi\, \mathbf{U}\, \psi}^{C} = \widehat{\varphi}^{C}\, \mathbf{U}\, \widehat{\psi}^{C}.$$

- finally, for formulas of the form $\langle\cdot A \cdot\rangle\, \varphi$ with $A = \{a_{j_1}, \ldots, a_{j_l}\}$, we let:

$$\widehat{\langle\cdot A\cdot\rangle\, \varphi}^{C} = \exists \mathsf{choose}_{j_1}^{1} \ldots \mathsf{choose}_{j_1}^{\alpha} \ldots \mathsf{choose}_{j_l}^{1} \ldots \mathsf{choose}_{j_l}^{\alpha}. \Big[\mathbf{AG}\Big(\bigwedge_{a_j \in A} \bigvee_{m = 1 \ldots \alpha} (\mathsf{choose}_j^m \wedge \bigwedge_{n \neq m} \neg\, \mathsf{choose}_j^n)\Big) \wedge \mathbf{A}\Big[\mathbf{G}\Big(\bigwedge_{a_j \in A \cup C} \bigwedge_{m = 1 \ldots \alpha} (\mathsf{choose}_j^m \Rightarrow \mathbf{X}\, \mathsf{mov}_j^m)\Big) \Rightarrow \widehat{\varphi}^{C \cup A}\Big]\Big].$$

The first part of this formula requires that the atomic propositions choose_j^m describe a strategy, while the second part expresses that every execution following the labelled strategies (including those for C) satisfies the path formula $\widehat{\varphi}^{C \cup A}$.

Now, letting $\widehat{\Phi}$ be the formula $\Phi_{\mathsf{Edge}} \wedge \widehat{\widehat{\Phi}}^{\varnothing}$, we have the following theorem (similar to Theorem 10):

Theorem 12. *Let Φ be an ATL_{sc} formula, $\mathsf{Agt} = \{a_1, \ldots, a_n\}$ be a finite set of agents, $\mathscr{M} = \{1, \ldots, \alpha\}$ be a finite set of moves, and $\widehat{\Phi}$ be the formula defined above. Then Φ is $(\mathsf{Agt}, \mathscr{M})$-satisfiable in a CGS if, and only if, the QCTL^* formula $\widehat{\Phi}$ is satisfiable (in the tree semantics).*

We end up with a non-elementary algorithm (in $(\mathsf{k}+2)$-$\mathsf{EXPTIME}$ for a formula involving k strategy quantifiers) for solving satisfiability of an ATL_{sc} formula for a bounded number of moves, both for a fixed or for an unspecified set of agents (we can infer the set of agents using Prop. 8). Since ATL_{sc} model checking is non-elementary even for a fixed number of moves (the crucial point is the alternation of strategy quantifiers), we deduce:

Corollary 13. *$(\mathsf{Agt}, \mathscr{M})$-satisfiability for ATL_{sc} formulas is non-elementary (i.e., k-$\mathsf{EXPTIME}$-hard, for all k).*

6 Memoryless strategies

Memoryless strategies are strategies that only depend on the present state (as opposed to general strategies, whose values can depend on the whole history). Restricting strategy quantifiers to memoryless strategies in the logic makes model checking much easier: in a finite game, there are only finitely many memoryless strategies to test, and applying a memoryless strategy just amounts to removing some transitions in the graph. Still, quantification over memoryless strategies is not possible in plain ATL_{sc}, and this additional expressive power turns out to make satifiability undecidable, even when restricting to turn-based games. One should notice that the undecidability proof of [TW12] for satisfiability in concurrent games uses one-step games (*i.e.*, they only involve one $\mathbf{X}$ modality), and hence also holds for memoryless strategies.

Theorem 14. *Satisfiability of ATL^0_{sc} (with memoryless-strategy quantification) is undecidable, even when restricting to turn-based games.*

Proof. We prove the result for infinite-state turn-based games, by adapting the corresponding proof for QCTL under the structure semantics [Fre01], which consists in encoding the problem of tiling a quadrant. The result for finite-state turn-based games can be obtained using similar (but more involved) ideas, by encoding the problem of tiling all finite grids (see [LM13] for the corresponding proof for QCTL).

We consider a finite set T of tiles, and two binary relations H and V indicating which tile(s) may appear on the right and above (respectively) a given tile. Our proof consists in writing a formula that is satisfiable only on a grid-shaped (turn-based) game structure representing a tiling of the quadrant (*i.e.*, of $\mathbb{N} \times \mathbb{N}$). The reduction involves two players: Player 1 controls square states (which are labelled with $\square$), while Player 2 controls circle states (labelled with $\bigcirc$). Each state of the grid is intended to represent one cell of the quadrant to be tiled. For technical reasons, the reduction is not that simple, and our game structure will have three kinds of states (see Fig. 1):

- the "main" states (controlled by Player 2), which form the grid. Each state in this main part has a *right* neighbour and a *top* neighbour, which we assume we can identify: more precisely, we make use of two atomic propositions v_1 and v_2 which alternate along the horizontal lines of the grid. The *right* successor of a v_1-state is labelled with v_2, while its *top* successor is labelled with v_1;
- the "tile" states, labelled with one item of T (seen as atomic propositions). Each tile state only has outgoing transition(s) to a tile state labelled with the same tile;

- the "choice" states, which appear between "main" states and "tile" states: there is one choice state associated with each main state, and each choice state has a transition to each tile state. Choice states are controlled by Player 1.

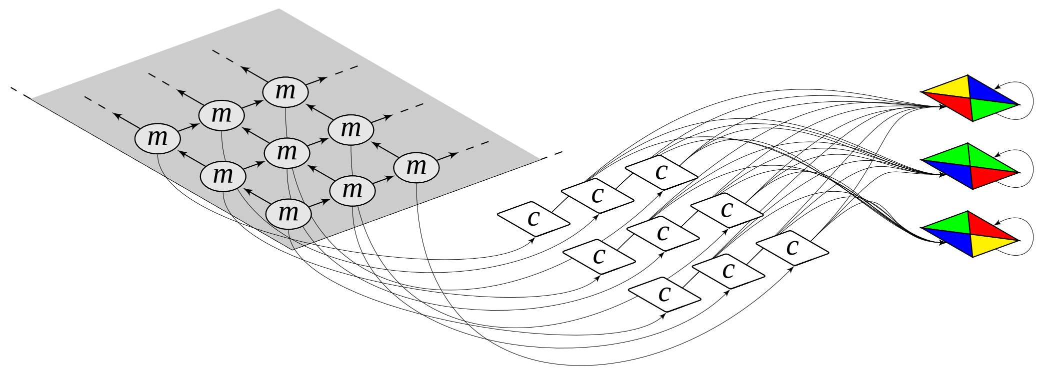

Fig. 1: The turn-based game encoding the tiling problem

Assuming that we have such a structure, a tiling of the grid corresponds to a *memoryless* strategy of Player 1 (who only plays in the "choice" states). Once such a memoryless strategy for Player 1 has been selected, that it corresponds to a valid tiling can be expressed easily: for instance, in any cell of the grid (assumed to be labelled with v_1), there must exist a pair of tiles $(t_1,t_2) \in H$ such that $v_1 \wedge \langle\cdot 2\cdot\rangle_0 \mathbf{X}\mathbf{X} t_1 \wedge \langle\cdot 2\cdot\rangle_0 \mathbf{X}(v_2 \wedge \mathbf{X}\mathbf{X} t_2)$. This would be written as follows:

$$\langle\cdot 1\cdot\rangle_0 \mathbf{G} \left[\begin{array}{cc} & v_1 \Rightarrow \bigvee_{(t_1,t_2)\in H} \langle\cdot 2\cdot\rangle_0 \mathbf{X}\mathbf{X} t_1 \wedge \langle\cdot 2\cdot\rangle_0 \mathbf{X}(v_2 \wedge \mathbf{X}\mathbf{X} t_2) \\ \wedge & \\ & v_2 \Rightarrow \bigvee_{(t_1,t_2)\in H} \langle\cdot 2\cdot\rangle_0 \mathbf{X}\mathbf{X} t_1 \wedge \langle\cdot 2\cdot\rangle_0 \mathbf{X}(v_1 \wedge \mathbf{X}\mathbf{X} t_2) \end{array} \right].$$

The same can be imposed for vertical constraints, and for imposing a fairness constraint on the base line (under the same memoryless strategy for Player 1).

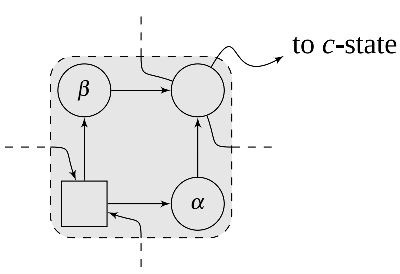

Fig. 2: The cell gadget

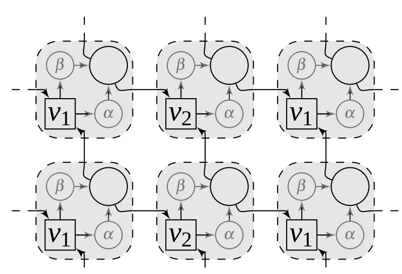

Fig. 3: Several cells forming (part of) a grid

It remains to build a formula characterising an infinite grid. This requires a slight departure from the above description of the grid: each main state will in fact be a gadget composed of four states, as depicted on Fig. 2. The first state of each gadget will give the opportunity to Player 1 to *color* the state with either α or β. This will be used to enforce "confluence" of several transitions to the same state (which we need to express that the two successors of any cell of the grid share a common successor).

We now start writing our formula, which we present as a conjunction of several subformulas. We require that the main states be labelled with m, the choice states be labelled with c, and the tile states be labelled with the names of the tiles. We let $\mathsf{AP}' = \{m,c\} \cup T$ and $\mathsf{AP} = \mathsf{AP}' \cup \{v_1, v_2, \alpha, \beta, \Box, \bigcirc\}$. The first part of the formula reads a follows (where universal path quantification can be encoded, as long as the context is empty, using $\langle\cdot\varnothing\cdot\rangle_0$):

$$\mathbf{AG}\left[\bigvee_{p\in\mathsf{AP}'} p \wedge \bigwedge_{p'\in\mathsf{AP}'\setminus\{p\}} \neg p'\right] \wedge \mathbf{A}(m\,\mathbf{W}\,c) \wedge \mathbf{AG}\left[c \Rightarrow \left(\Box \wedge \bigwedge_{t\in T} \langle\cdot 1\cdot\rangle_0\,\mathbf{X}\,t \wedge \mathbf{AX}\left(\bigvee_{t\in T} \mathbf{AG}\,t\right)\right)\right] \wedge$$
$$\mathbf{AG}\left[(\Box \Leftrightarrow \neg\bigcirc) \wedge \left(\Box \Rightarrow \bigwedge_{p\in\mathsf{AP}} (\mathbf{EX}\,p \Leftrightarrow \langle\cdot 1\cdot\rangle_0\,\mathbf{X}\,p)\right) \wedge \left(\bigcirc \Rightarrow \bigwedge_{p\in\mathsf{AP}} (\mathbf{EX}\,p \Leftrightarrow \langle\cdot 2\cdot\rangle_0\,\mathbf{X}\,p)\right)\right] \quad (1)$$

This formula enforces that each state is labelled with exactly one proposition from AP'. It also enforces that any path will wander through the main part until it possibly goes to a choice state (this is expressed as $\mathbf{A}(m\,\mathbf{W}\,c)$, where $m\,\mathbf{W}\,c$ means $\mathbf{G}\,m \vee m\,\mathbf{U}\,c$, and can be expressed a negated-until formula). Finally, the second part of the formula enforces the witnessing structures to be turn-based.

Now we have to impose that the m-part has the shape of a grid: intuitively, each cell has three successors: one "to the right" and one "to the top" in the main part of the grid, and one c-state which we will use for associating a tile with this cell. For technical reasons, the situation is not that simple, and each cell is actually represented by the gadget depicted on Fig. 2. Each state of the gadget is labelled with m. We constrain the form of the cells as follows:

$$\mathbf{AG}\Big[m \Rightarrow ((\Box \wedge \neg\alpha \wedge \neg\beta) \vee (\bigcirc \wedge \neg(\alpha \wedge \beta)))\Big] \wedge \mathbf{AG}\Big[\big((m \wedge \Box) \Rightarrow (v_1 \Leftrightarrow \neg v_2)\big) \wedge \big((v_1 \vee v_2) \Rightarrow (m \wedge \Box)\big)\Big] \wedge$$
$$\mathbf{AG}\Big[(m \wedge \Box) \Rightarrow \Big[\mathbf{AX}\big(m \wedge \bigcirc \wedge (\alpha \vee \beta) \wedge \mathbf{AX}\,(m \wedge \bigcirc \wedge \neg\alpha \wedge \neg\beta)\big) \wedge \langle\cdot 1\cdot\rangle_0\,\mathbf{X}\,\alpha \wedge \langle\cdot 1\cdot\rangle_0\,\mathbf{X}\,\beta\Big]\Big] \quad (2)$$

This says that there are four types of states in each cell, and specifies the possible transitions within such cells. We now express constraints on the transitions leaving a cell:

$$\mathbf{AG}\Big[(\mathbf{EX}\,c \vee \mathbf{EX}\,v_1 \vee \mathbf{EX}\,v_2) \Rightarrow (m \wedge \bigcirc \wedge \neg\alpha \wedge \neg\beta)\Big] \wedge$$
$$\mathbf{AG}\Big[(m \wedge \bigcirc \wedge \neg\alpha \wedge \neg\beta) \Rightarrow (\mathbf{EX}\,c \wedge \mathbf{EX}\,v_1 \wedge \mathbf{EX}\,v_2 \wedge \mathbf{AX}\,(c \vee v_1 \vee v_2)\Big] \quad (3)$$

It remains to enforce that the successor of the α and β states are the same. This is obtained by the following formula:

$$\mathbf{AG}\Big[(m \wedge \Box) \Rightarrow [\cdot 2\cdot]_0\,\big(\langle\cdot\varnothing\cdot\rangle_0\,\mathbf{X}^3(c \vee v_1) \vee \langle\cdot\varnothing\cdot\rangle_0\,\mathbf{X}^3(c \vee v_2)\big)\Big] \quad (4)$$

Indeed, assume that some cell has two different "final" states; then there would exist a strategy for Player 2 (consisting in playing differently in those two final states) that would violate Formula (4). Hence each cell as a single final state.

We now impose that each cell in the main part has exactly two m-successors, and these two m-successors have an m-successor in common. For the former property, Formula (3) already imposes that each cell has at least two m-successors (one labelled with v_1 and one with v_2). We enforce that there cannot be more that two:

$$\mathbf{AG}\Big[(m \wedge \Box) \Rightarrow [\cdot 1\cdot]_0\,[(\langle\cdot 2\cdot\rangle_0\,\mathbf{X}^3(v_1 \wedge \mathbf{X}\,\alpha) \wedge \langle\cdot 2\cdot\rangle_0\,\mathbf{X}^3(v_2 \wedge \mathbf{X}\,\alpha)) \Rightarrow [\cdot 2\cdot]_0\,\langle\cdot\varnothing\cdot\rangle_0\,\mathbf{X}^3\mathbf{X}\,\alpha]\Big]. \quad (5)$$

Notice that $[\cdot 2\cdot]_0\, \langle\cdot\varnothing\cdot\rangle_0\, \varphi$ means that φ has to hold along any outcome of any *memoryless* strategy of Player 2. Assume that a cell has three (or more) successor cells. Then at least one is labelled with v_1 and at least one is labelled with v_2. There is a strategy for Player 1 to color one v_1-successor cell and one v_2-successor cell with α, and a third successor cell with β, thus violating Formula (5) (as Player 2 has a strategy to reach a successor cell colored with β)

For the latter property (the two successors have a common successor), we add the following formula (as well as its v_2-counterpart):

$$[\cdot 1\cdot]_0\, \langle\cdot\varnothing\cdot\rangle_0\, \mathbf{G}\Big[(m\wedge\Box\wedge v_1)\Rightarrow\Big([\langle\cdot 2\cdot\rangle_0\, \mathbf{X}^3(v_1\wedge [\cdot 2\cdot]_0\, \mathbf{X}^3\mathbf{X}\,\alpha)]\Rightarrow[\langle\cdot 2\cdot\rangle_0\, \mathbf{X}^3(\neg v_1\wedge \mathbf{X}^3(\neg v_1\wedge \mathbf{X}\,\alpha))]\Big)\Big] \tag{6}$$

In this formula, the initial (universal) quantification over strategies of Player 1 fixes a color for each cell. The formula claims that whatever this choice, if we are in some v_1-cell and can move to another v_1-cell whose two successors have color α, then also we can move to a v_2-cell having one α successor (which we require to be a v_2-cell). As this must hold for any coloring, both successors of the original v_1-cell share a common successor. Notice that this does not prevent the grid to be collapsed: this would just indicate that there is a *regular* infinite tiling.

We conclude by requiring that the initial state be in a square state of a cell in the main part. $\square$

7 Results for Strategy Logic

In this section, we extend the previous results to Strategy Logic (SL). This logic has been initially introduced in [CHP07] for two-player turn-based games. It has then been extended to n-players concurrent games in [MMV10]. As explained in the introduction, satisfiability has been shown undecidable when considering infinite structures [MMV10], and the proof in [TW12] for finite satisfiability of ATL_{sc} straightforwardly extends to SL. Here we show that satisfiability is decidable when considering turn-based games and when fixing a finite alphabet, and that it remains undecidable when only considering memoryless strategies.

Strategy Logic in a nutshell. We start by briefly recalling the main ingredients of SL. The syntax is given by the following grammar:

$$\varphi,\psi ::= p \mid \varphi\wedge\psi \mid \neg\,\varphi \mid \mathbf{X}\,\varphi \mid \varphi\,\mathbf{U}\,\psi \mid \langle\!\langle x\rangle\!\rangle\,\varphi \mid (a,x)\varphi$$

where $a\in\mathsf{Agt}$ is an agent and x is a (strategy) variable (we use Var to denote the set of these variables). Formula $\langle\!\langle x\rangle\!\rangle\,\varphi$ expresses the existence of a strategy, which is stored in variable x, under which formula φ holds. In φ, the *agent binding* operator (a,x) can be used to bind agent a to follow strategy x. An assignment χ is a partial function from $\mathsf{Agt}\cup\mathrm{Var}$ to Strat. SL formulas are interpreted over pairs (χ,q) where q is a state of some CGS and χ is an assignment such that every free strategy variable/agent[4] occurring in the formula belongs to $\mathsf{dom}(\chi)$. Note that we have $\mathsf{Agt}\subseteq\mathsf{dom}(\chi)$ when temporal modalities $\mathbf{X}$ and $\mathbf{U}$ are interpreted: this implies that the set of outcomes is restricted to a unique execution generated by all the strategies assigned to players in Agt, and the temporal modalities are therefore interpreted over this execution. Here we just give the semantics of the main two constructs (see [MMV10] for a complete definition of SL):

$$\begin{aligned}\mathscr{C},\chi,q\models\langle\!\langle x\rangle\!\rangle\,\varphi \quad &\text{iff}\quad \exists F\in\mathsf{Strat}\text{ s.t. }\mathscr{C},\chi[x\mapsto F],q\models\varphi\\ \mathscr{C},\chi,q\models(a,x)\varphi \quad &\text{iff}\quad \mathscr{C},\chi[a\mapsto\chi(x)],q\models\varphi\end{aligned}$$

[4] We use the standard notion of freedom for the strategy variables with the hypothesis that $\langle\!\langle x\rangle\!\rangle$ binds x, and for the agents with the hypothesis that (a,x) binds a and that every agent in Agt is free in temporal subformula (*i.e.*, with $\mathbf{U}$ or $\mathbf{X}$ as root).

In the following we assume w.l.o.g. that every quantifier $\langle\!\langle x\rangle\!\rangle$ introduces a fresh strategy variable x: this allows us to permanently use variable x to denote the selected strategy for a.

Turn-based case. The approach we used for ATL_{sc} can be adapted for SL. Given an SL formula Φ and a mapping $V\colon \mathsf{Agt} \to \mathrm{Var}$, we define a QCTL^* formula $\widehat{\Phi}^V$ inductively as follows (Boolean cases omitted):

$$\widehat{\langle\!\langle x\rangle\!\rangle\,\varphi}^V = \exists \mathsf{mov}_x. \Big[\mathbf{AG}\Big(\mathbf{EX}_1 \mathsf{mov}_x\Big) \wedge \widehat{\varphi}^V\Big] \qquad\qquad \widehat{(a,x)\varphi}^V = \widehat{\varphi}^{V[a\to x]}$$

Note that in this case we require that *every* reachable state has a (unique) successor labeled with mov_x: indeed when one quantifies over a strategy x, the agent(s) who will use this strategy are not known yet. However, in the turn-based case, a given strategy should be dedicated to a single agent: there is no natural way to share a strategy for two different agents (or the other way around, any two strategies for two different agents can be seen as a single strategy), as they are not playing in the same states. When the strategy x is assigned to some agent a, only the choices made in the a-states are considered.

The temporal modalities are treated as follows:

$$\widehat{\varphi\,\mathbf{U}\,\psi}^V = \mathbf{A}\Big[\mathbf{G}\Big(\bigwedge_{a_j\in\mathsf{Agt}} (\mathsf{turn}_j \Rightarrow \mathbf{X}\,\mathsf{mov}_{V(a_j)})\Big) \Rightarrow \widehat{\varphi}^V\,\mathbf{U}\,\widehat{\psi}^V\Big]$$
$$\widehat{\mathbf{X}\varphi}^V = \mathbf{A}\Big[\mathbf{G}\Big(\bigwedge_{a_j\in\mathsf{Agt}} (\mathsf{turn}_j \Rightarrow \mathbf{X}\,\mathsf{mov}_{V(a_j)})\Big) \Rightarrow \mathbf{X}\,\widehat{\varphi}^V\Big]$$

Now let $\widetilde{\Phi}$ be the formula $\Phi_{tb} \wedge \widehat{\Phi}^{V_\varnothing}$. Then we have the following theorem:

Theorem 15. *Let Φ be an SL formula and $\widetilde{\Phi}$ be the QCTL^* formula defined as above. Then Φ is satisfiable in a turn-based CGS if, and only if, $\widetilde{\Phi}$ is satisfiable (in the tree semantics).*

Bounded action alphabet Let $\mathscr{M}$ be $\{1,\ldots,\alpha\}$. The reduction carried out for ATL_{sc} can also be adapted for SL in this case. Given an SL formula Φ and a partial function $V\colon \mathsf{Agt} \to \mathrm{Var}$, we define the QCTL^* formula $\widehat{\widehat{\Phi}}^V$ inductively as follows:

$$\widehat{\widehat{\langle\!\langle x\rangle\!\rangle\,\varphi}}^V = \exists \mathsf{choose}_x^1 \ldots \exists \mathsf{choose}_x^\alpha. \mathbf{AG}\Big(\bigvee_{1\le m\le\alpha} \mathsf{choose}_x^m \wedge \bigwedge_{n\neq m} \neg\,\mathsf{choose}_x^n\Big) \wedge \widehat{\widehat{\varphi}}^V \qquad \widehat{\widehat{(a,x)\varphi}}^V = \widehat{\widehat{\varphi}}^{V[a\mapsto x]}$$

The temporal modalities are handled as follows:

$$\widehat{\widehat{\varphi\,\mathbf{U}\,\psi}}^V = \mathbf{A}\Big[\Big(\mathbf{G}\bigwedge_{a_j\in\mathsf{Agt}}\bigwedge_{1\le m\le\alpha}\Big(\mathsf{choose}^m_{V(a_j)} \Rightarrow \mathbf{X}\,\mathsf{mov}^m_j\Big)\Big) \Rightarrow \Big(\widehat{\widehat{\varphi}}^V\,\mathbf{U}\,\widehat{\widehat{\psi}}^V\Big)\Big]$$
$$\widehat{\widehat{\mathbf{X}\varphi}}^C = \mathbf{A}\Big[\Big(\mathbf{G}\bigwedge_{a_j\in\mathsf{Agt}}\bigwedge_{1\le m\le\alpha}\Big(\mathsf{choose}^m_{V(a_j)} \Rightarrow \mathbf{X}\,\mathsf{mov}^m_j\Big)\Big) \Rightarrow \Big(\mathbf{X}\,\widehat{\widehat{\varphi}}^V\Big)\Big]$$

Remember that in this case, mov^m_j labels the possible successors of a state where agent a_j plays m.

Finally, let $\widehat{\Phi}$ be the formula $\Phi_{\mathrm{move}} \wedge \widehat{\widehat{\Phi}}^V_\varnothing$. We have:

Theorem 16. *Let Φ be an SL formula based on the set $\mathsf{Agt} = \{a_1,\ldots,a_n\}$, let $\mathscr{M} = \{1,\ldots,\alpha\}$ be a finite set of moves, and $\widehat{\Phi}$ be the QCTL^* formula defined as above. Then Φ is $(\mathsf{Agt},\mathscr{M})$-satisfiable if, and only if, $\widehat{\Phi}$ is satisfiable (in the tree semantics).*

7.1 Memoryless strategies

We now extend the undecidability result of ATL^0_{sc} to SL with memoryless-strategy quantification. Notice that there is an important difference between ATL^0_{sc} and SL^0 (the logic obtained from SL by quantifying only on memoryless strategies): the ATL_{sc}-quantifier $\langle\cdot A\cdot\rangle_0$ still has an implicit quantification over *all* the strategies of the other players (unless their strategy is fixed by the context), while in SL^0 all strategies must be explicitly quantified. Hence SL^0 and ATL^0_{sc} have uncomparable expressiveness. Still:

Theorem 17. SL^0 *satisfiability is undecidable, even when restricting to turn-based game structures.*

Proof (sketch). The proof uses a similar reduction as for the proof for ATL^0_{sc}. The difference is that the implicitly-quantified strategies in ATL^0_{sc} are now explicitly quantified, hence memoryless. However, most of the properties that our formulas impose are "local" conditions (involving at most four nested "next" modalities) imposed in all the reachable states. Such properties can be enforced even when considering only the ultimately periodic paths that are outcomes of memoryless strategies. The only subformula not of this shape is formula $\mathbf{A} m \mathbf{W} c$, but imposing this property along the outcomes of memoryless strategies is sufficient to have the formula hold true along any path. □

8 Conclusion

While satisfiability for ATL_{sc} and SL is undecidable, we proved in this paper that it becomes decidable when restricting the search to turn-based games. We also considered the case where strategy quantification in those logics is restricted to memoryless strategies: while this makes model checking easier, it makes satisfiability undecidable, even for turn-based structures. These results have been obtained by following the tight and natural link between those temporal logics for games and the logic QCTL, which extends CTL with quantification over atomic propositions. This witnesses the power and usefulness of QCTL, which we will keep on studying to derive more results about temporal logics for games.

Acknowledgement. We thank the anonymous reviewers for their numerous suggestions, which helped us improve the presentation of the paper.

References

[AHK02] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *J. ACM*, 49(5):672–713, 2002. doi:10.1145/585265.585270.

[BDLM09] Th. Brihaye, A. Da Costa, F. Laroussinie, and N. Markey. ATL with strategy contexts and bounded memory. In *Proceedings of the International Symposium Logical Foundations of Computer Science (LFCS'09)*, LNCS 5407, p. 92–106. Springer, 2009. doi:10.1007/978-3-540-92687-0_7.

[CE82] E. M. Clarke and E. A. Emerson. Design and synthesis of synchronization skeletons using branching-time temporal logic. In *Proceedings of the 3rd Workshop on Logics of Programs (LOP'81)*, LNCS 131, p. 52–71. Springer, 1982. doi:10.1007/BFb0025774.

[CHP07] K. Chatterjee, T. A. Henzinger, and N. Piterman. Strategy logic. In *Proceedings of the 18th International Conference on Concurrency Theory (CONCUR'07)*, LNCS 4703, p. 59–73. Springer, 2007. doi:10.1007/978-3-540-74407-8_5.

[DLM10] A. Da Costa, F. Laroussinie, and N. Markey. ATL with strategy contexts: Expressiveness and model checking. In *Proceedings of the 30th Conferentce on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'10)*, LIPIcs 8, p. 120–132. Leibniz-Zentrum für Informatik, 2010. doi:10.4230/LIPIcs.FSTTCS.2010.120.

[DLM12] A. Da Costa, F. Laroussinie, and N. Markey. Quantified CTL: Expressiveness and model checking. In *Proceedings of the 23rd International Conference on Concurrency Theory (CONCUR'12)*, LNCS 7454, p. 177–192. Springer, 2012. doi:10.1007/978-3-642-32940-1_14.

[Fre01] T. French. Decidability of quantified propositional branching time logics. In *Proceedings of the 14th Australian Joint Conference on Artificial Intelligence (AJCAI'01)*, LNCS 2256, p. 165–176. Springer, 2001. doi:10.1007/3-540-45656-2_15.

[HSW13] C.-H. Huang, S. Schewe, and F. Wang. Model-checking iterated games. In *Proceedings of the 19th International Conference on Tools and Algorithms for Construction and Analysis of Systems (TACAS'13)*, LNCS 7795, p. 154–168. Springer, 2013. doi:10.1007/978-3-642-36742-7_11.

[Kur02] A. Kurucz. $S5 \times S5 \times S5$ lacks the finite model property. In *Proceedings of the 3rd Workshop on Advances in Modal Logic (AIML'00)*, p. 321–327. World Scientific, 2002.

[LM13] F. Laroussinie and N. Markey. Quantified CTL: expressiveness and complexity. Research Report LSV-13-07, Lab. Spécification & Vérification, ENS Cachan, France, 2013.

[MMPV12] F. Mogavero, A. Murano, G. Perelli, and M. Y. Vardi. What makes ATL* decidable? a decidable fragment of strategy logic. In *Proceedings of the 23rd International Conference on Concurrency Theory (CONCUR'12)*, LNCS 7454, p. 193–208. Springer, 2012. doi:10.1007/978-3-642-32940-1_15.

[MMV10] F. Mogavero, A. Murano, and M. Y. Vardi. Reasoning about strategies. In *Proceedings of the 30th Conferentce on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'10)*, LIPIcs 8, p. 133–144. Leibniz-Zentrum für Informatik, 2010. doi:10.4230/LIPIcs.FSTTCS.2010.133.

[Pnu77] A. Pnueli. The temporal logic of programs. In *Proceedings of the 18th Annual Symposium on Foundations of Computer Science (FOCS'77)*, p. 46–57. IEEE Comp. Soc. Press, 1977. doi:10.1109/SFCS.1977.32.

[QS82] J.-P. Queille and J. Sifakis. Specification and verification of concurrent systems in CESAR. In *Proceedings of the 5th International Symposium on Programming (SOP'82)*, LNCS 137, p. 337–351. Springer, 1982. doi:10.1007/3-540-11494-7_22.

[TW12] N. Troquard and D. Walther. On satisfiability in atl with strategy contexts. In *Proceedings of the 13th European Conference in Logics in Artificial Intelligence (JELIA'12)*, LNCS 7519, p. 398–410. Springer, 2012. doi:10.1007/978-3-642-33353-8_31.

[WHY11] F. Wang, C.-H. Huang, and F. Yu. A temporal logic for the interaction of strategies. In *Proceedings of the 22nd International Conference on Concurrency Theory (CONCUR'11)*, LNCS 6901, p. 466–481. Springer, 2011. doi:10.1007/978-3-642-23217-6_31.

[WLWW06] D. Walther, C. Lutz, F. Wolter, and M. Wooldridge. ATL satisfiability is indeed EXPTIME-complete. *Journal of Logic and Computation*, 16(6):765–787, 2006. doi:10.1093/logcom/exl009.