On Satisfiability in ATL with Strategy Contexts

Nicolas Troquard[1] and Dirk Walther[2]

[1] Laboratory for Applied Ontology (ISTC-CNR), Trento, Italy
[2] Universidad Politécnica de Madrid, Spain

Abstract. This paper is a study of Brihaye *et al.*'s ATL with strategy contexts. We focus on memory-less strategies and establish that the resulting logic is undecidable. An immediate corollary follows that the problem of satisfiability checking of every variant of ATL with strategy context introduced by Brihaye *et al.* is undecidable. We also relate ATL_{sc} with memory-less strategies with ATL with explicit strategies, providing a decidable fragment.

1 Introduction

With Alternating-time Temporal Logic $\mathsf{ATL}^{(*)}$ ([2,14]) one can reason about the *ability* of a coalition to ensure something *whatever the other agents do.* It is the logic of sentences like "The monitoring units $u_1, \ldots, u_l$ can ensure that the system stays in a failsafe state." In this paper, we consider the recent variant of ATL with strategy contexts [4,6]. A *strategy context* is the actual current strategy of some committed set of agents. The truth value of an ATL_{sc}-formula is evaluated in a concurrent game structure, at a state, and *wrt. a strategy context.* Informally, the formula $\langle\cdot A\cdot\rangle\psi$ states that A has a strategy to ensure the property ψ in the context of the current strategy commitment. Like in ATL, the formula ψ typically represents a temporal property, but unlike the ATL path quantifier, the modality $\langle\cdot A\cdot\rangle$ *commits* the members of A to their chosen strategy F_A. Henceforth, the commitment is used for the evaluation of ψ. That is, ψ is evaluated wrt. to a strategy context consisting in the initial strategy context updated with F_A. The operator $\cdot\rangle A\langle\cdot$ *releases* this commitment. Under the common assumptions of ATL, the ATL path quantifier is trivially captured by

$$\langle\langle A\rangle\rangle\psi \stackrel{\text{def}}{=} \cdot\rangle\Sigma\langle\cdot\langle\cdot A\cdot\rangle\psi,$$

where Σ is the set of all agents.

The notion of ability of a coalition in ATL_{sc} is their ability given the context of the strategies that the coalition is actually committed to. *Actual agency*, the property of some agentive entity in the act of doing something, is ubiquitous in our everyday life: "Unit u_1 is inspecting the register $0x12345678$." It is all the more important in a multi-agent framework where agents strategise given some input (observation, expectation, belief, etc.) about the strategies followed by the other players, and their abilities depend on it: "If units $u_1, \ldots, u_{i-1}, u_{i+1}, \ldots, u_l$ do not know which register u_i is inspecting, they cannot ensure that no system

L. Fariñas del Cerro, A. Herzig, and J. Mengin (Eds.): JELIA 2012, LNCS 7519, pp. 398–410, 2012.

failure will occur." Actual agency is also central to game theory, where for instance, a Nash equilibrium occurs when every agent is playing his best response to the current strategy of the other agents. With the advent of the Internet and service-oriented computing, system designers in industry and in academia rely increasingly on the multi-agent paradigm. As we seek after the 'next generation' of logics for the specification of properties of societies of agents, and for the verification of their designs, it appears important to be able to talk and reason about actual agency of coalitions of agents, and their contextualised ability.

ATL_{sc} and ATL^*_{sc} can capture a variety of notions of *strategic actual agency* that lie beyond the mere ability of coalitions as captured by ATL. For instance, a type of STIT modality ([11,5]) can be defined as

$$[A\ \mathsf{sstit}]\psi \stackrel{\text{def}}{=} \cdot\rangle \Sigma \setminus A\langle\cdot\langle\cdot\emptyset\cdot\rangle\psi,$$

reading "A is seeing to it that ψ." (See the earlier report [18] for a detailed discussion about ATL and STIT modalities.)

In the pure tradition of knowledge representation it is also useful to be able to talk about strategies in a more explicit manner. Practically, they can serve, e.g., as explicit delegation instruction between agents. We will contrast the use of strategy contexts with *explicit strategies.* ATL_{sc} and ATLES ([19]) capture the notions of commitment to, release and recall of strategies, as well as irrevocable strategies ([1]). We introduce ATLES on concurrent game structures in Section 3 and relate ATL_{sc} with ATLES, determining a decidable fragment of ATL_{sc}.

Originating from theoretical computer science and verification, the focus of ATL_{sc} has been on model checking so far, and not satisfiability. In Section 4, we establish that the satisfiability problem for both ATL_{sc} and ATL^*_{sc} is undecidable in general, emphasising the significance of the fragment previously identified.

In the next section we define rigorously the syntax and semantics of ATL_{sc} and ATL^*_{sc} that we have informally presented in this introduction.

2 ATL with Strategy Contexts

We fix a countable set of *atomic propositions* $\mathbf{\Pi}$ and a finite set of *agents* (or *players*) $\mathbf{\Sigma}$. The following grammar was given for ATL^*_{sc} in [6].

Definition 1 (ATL^*_{sc} syntax). *The following grammar defines state formulas φ and path formulas ψ, where p ranges over $\mathbf{\Pi}$ and A over finite subsets of $\mathbf{\Sigma}$. The language of ATL^*_{sc} consists of the state formulas.*

$$\begin{aligned}\varphi &::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \cdot\rangle A\langle\cdot\varphi \mid \langle\cdot A\cdot\rangle\psi \\ \psi &::= \varphi \mid \neg\psi \mid \psi \vee \psi \mid \bigcirc\varphi \mid \varphi\,\mathcal{U}\,\varphi\end{aligned}$$

The remaining Boolean operators $\wedge$, $\rightarrow$ and $\leftrightarrow$ as well as the logical constants $\top$ and $\bot$ can be defined as usual in terms of the operators given. The linear temporal logic operators 'sometime' and 'forever' can be defined as path formulas $\Diamond\varphi = (\top\,\mathcal{U}\,\varphi)$ and $\Box\varphi = \neg(\top\,\mathcal{U}\,\neg\varphi)$.

The language of ATL_{sc} consists only of some formulas from ATL^*_{sc}. The syntax of the path formulas ψ is restricted as follows (where φ refers to the state formulas in Def. 1):

$$\psi ::= \neg\psi \mid \bigcirc\varphi \mid \varphi\,\mathcal{U}\,\varphi$$

We evaluate the formulas on Concurrent Game Structures (CGSs), which are defined as follows.

Definition 2 (Concurrent Game Structure). *Let $\Sigma = \{1,\ldots,n\} \subset \mathbf{\Sigma}$, with $n \geq 1$, be a finite set of agents, and $\Pi \subset \mathbf{\Pi}$ be a finite set of atomic propositions. A* Concurrent Game Structure *(CGS) $\mathcal{C}$ for $\langle\Sigma,\Pi\rangle$ is a tuple $\mathcal{C} = \langle W, V, \Sigma, M, Mov, E\rangle$, where:*

- *W is a non-empty set of worlds (or game positions);*
- *$V : W \to 2^{\Pi}$ is a valuation function;*
- *M is a finite, non-empty set of moves;*
- *$Mov : W \times \Sigma \to 2^M \setminus \emptyset$ maps every world w and agent a to the non-empty set $Mov(w,a)$ of moves available to a at w; and*
- *$E : W \times M^{\Sigma} \to W$ is a transition function mapping a world w and a move profile $\boldsymbol{m} = \langle m_1,\ldots,m_n\rangle$ (one move for each agent) to the world $E(w,\boldsymbol{m})$.*

Let $\mathcal{C}$ be a CGS. The component Mov determines which of the moves from M are available for an agent at a world w. Let $\mathsf{prof}(w)$ be the set of available move profiles at world w, i.e.,

$$\mathsf{prof}(w) = \{\langle m_1,\ldots,m_n\rangle \mid m_i \in Mov(w,i)\}.$$

A move profile is used to determine a successor of a world using the transition function E. Let $\mathsf{succ}(w)$ be the set of possible successors of w, formally

$$\mathsf{succ}(w) = \{E(w,\boldsymbol{m}) \mid \boldsymbol{m} \in \mathsf{prof}(w)\}.$$

An infinite sequence $\lambda = x_0x_1x_2\cdots \in W^{\omega}$ of worlds is called a *play* or *computation* if $x_{i+1} \in \mathsf{succ}(x_i)$ for all positions $i \geq 0$. Denote with $\lambda[i]$ the i-th component x_i in λ, and with $\lambda[0,i]$ the initial sequence $x_0\cdots x_i$ of λ.

A *strategy for an agent* $a \in \Sigma$ is a function f_a that maps a world w from W to a move profile $f_a(w) \in Mov(w,a)$ available to a at w. A *strategy for a coalition* $A \subseteq \Sigma$ is a set F_A of strategies with $F_A = \{\sigma_a \mid a \in A\}$ containing one strategy for every agent in A. We refer to a strategy also as *strategy context.* We denote with $\mathsf{strat}(A)$ the set of strategies available to coalition A. The strategies considered here are *memoryless* as they are functions from worlds to move profiles and, thus, do not take previously visited states into account.

We define two operations on strategies: upgrade and release of strategies. Let F_A and F be strategies for sets of agents, where F_A contains strategies for the agents in A. The *upgrade* of F with the strategies in F_A is the result of *overwriting* F with strategies for the agents in $A \cap \mathsf{dom}(F)$ and *supplementing* F with strategies for agents for which F does not already provide a strategy (i.e., for agents in $A \setminus \mathsf{dom}(F)$). We will use $\circ$ as a strategy upgrade operator. Formally,

$$F_A \circ F = F_A \cup \{f_a \in F \mid a \notin A\}.$$

The *release* of the strategies for the agents in B from F is the *restriction* of F to strategies for agents that do not occur in B (i.e., for agents in $\Sigma \setminus B$). Formally, for $C = \Sigma \setminus B$,

$$F|_C = \{f_a \in F \mid a \in C\}.$$

The set $\mathsf{out}(w, F_A)$ of *outcomes* of a strategy F_A for the agents in A starting at a world w is the set of all plays $\lambda = x_0x_1x_2\cdots \in W^\omega$ such that $x_0 = w$ and, for every $i \geq 0$, there is a move profile $\boldsymbol{m} = \langle m_1, \ldots, m_n\rangle \in \mathsf{prof}(x_i)$ such that:

(i) $m_a = f_a(x_i)$, for all $a \in A$; and
(ii) $x_{i+1} = E(x_i, \boldsymbol{m})$.

The truth values of ATL^*_{sc}-formulas over CGSs is given as follows, where state formulas are evaluated at worlds (or game positions) and path formulas over infinite paths in a CGS.

Definition 3 (ATL^*_{sc} Semantics). *Given a CGS $\mathcal{C} = \langle W, R, V, \Sigma, M, Mov, E\rangle$ for $\langle \Sigma, \Pi\rangle$ and a strategy context F, the consequence relation $\models$ is inductively defined as follows:*

- $\mathcal{C}, w \models_F p$ *iff* $p \in V(w)$, *for all atomic propositions* $p \in \Pi$;
- $\mathcal{C}, w \models_F \neg\varphi$ *iff* $\mathcal{C}, w \not\models_F \varphi$;
- $\mathcal{C}, w \models_F \varphi_1 \vee \varphi_2$ *iff* $\mathcal{C}, w \models_F \varphi_1$ *or* $\mathcal{C}, w \models_F \varphi_2$;
- $\mathcal{C}, w \models_F \cdot\rangle A\langle\cdot\varphi$ *iff* $\mathcal{C}, w \models_S \varphi$, *where* $S = F|_{\Sigma\setminus A}$;
- $\mathcal{C}, w \models_F \langle\cdot A\cdot\rangle\psi$ *iff there is* $F_A \in \mathsf{strat}(A)$ *such that for all plays* $\lambda \in \mathsf{out}(w, S)$, *it holds that* $\mathcal{C}, \lambda \models_S \psi$, *where* $S = F_A \circ F$;
- $\mathcal{C}, \lambda \models_F \varphi$ *iff* $\mathcal{C}, \lambda[0] \models_F \varphi$, *when* φ *is a state formula;*
- $\mathcal{C}, \lambda \models_F \neg\psi$ *iff* $\mathcal{C}, \lambda \not\models_F \psi$;
- $\mathcal{C}, \lambda \models_F \psi_1 \vee \psi_2$ *iff* $\mathcal{C}, \lambda \models_F \psi_1 \vee \psi_2$;
- $\mathcal{C}, \lambda \models_F \bigcirc\varphi$ *iff* $\mathcal{C}, \lambda[1] \models_F \varphi$;
- $\mathcal{C}, \lambda \models_F (\varphi_1 \,\mathcal{U}\, \varphi_2)$ *iff there is an* $i \geq 0$ *such that* $\mathcal{C}, \lambda[i] \models_F \varphi_2$ *and* $\mathcal{C}, \lambda[j] \models_F \varphi_1$ *for all* j *with* $0 \leq j < i$.

A formula φ is satisfiable *if $\mathcal{C}, w \models_F \varphi$ for some CGS $\mathcal{C}$, some world w in $\mathcal{C}$ and some strategy context F in $\mathcal{C}$; a formula is called* valid *if $\mathcal{C}, w \models_F \varphi$ for all $\mathcal{C}$, all w and all F.*

In this paper, we do not assume agents being capable of perfect recall. In fact, we use a semantics for ATL_{sc} and ATL^*_{sc} that is based on memoryless strategies. This means that agents use strategies that prescribe for every world which move to take. The history of previously visited worlds is not taken into account. In [4,6], these logics are denoted with $\mathsf{ATL}_{\mathrm{sc},0}$ and $\mathsf{ATL}^*_{\mathrm{sc},0}$.

3 Strategy Contexts and Explicit Strategies

In this section, we contrast the notion of strategy contexts with explicit strategies. Many notions relevant to strategies come into the picture and our principal aim is to discuss them informally. We first present ATLES, the extension of ATL with explicit strategies from [19] (Section 3.1). We introduce it over CGSs while its original presentation was in terms of alternating transition systems. We then translate a fragment of ATL_{sc} into ATLES (Section 3.2).

3.1 ATLES

The language of ATL is enriched with symbols for strategies and commitment functions that assign agents to strategies they are committed to play. Thus ATLES allows to reason explicitly about strategies. This is not possible with any of ATL and ATL_{sc} (and their respective LTL-extensions) as strategies are pure semantic constructs and they do not occur in the object language.

Formally, the signature of the language is extended by a set $\mathbf{\Upsilon}$ of *strategy terms*, where $\mathbf{\Upsilon} = \bigcup_{a\in\mathbf{\Sigma}} \mathbf{\Upsilon}_a$ and $\mathbf{\Upsilon}_a$ is a countably infinite set of *strategy terms* σ_a for agent a in $\mathbf{\Sigma}$. A *commitment function* is a partial function $\rho : \mathbf{\Sigma} \to \mathbf{\Upsilon}$ with a finite domain mapping an agent $a \in \mathbf{\Sigma}$ to a strategy term $\rho(a) \in \mathbf{\Upsilon}_a$ for a. Note that a commitment function ρ is a finite object and as such it is used to additionally parameterise path-quantifiers as $\langle\!\langle A\rangle\!\rangle_\rho$. The set $\mathsf{dom}(\rho)$ consists of the committed agents. If $\rho(a)$ is defined, then ρ contains a mapping of the form $a \mapsto \sigma_a$ which is called a *commitment* of agent a (or a *commits*) to play the strategy denoted by the strategy term σ_a. On the other hand, if $\rho(a)$ is undefined, then a does not commit to any strategy and, thus, a can quantify freely over the strategies available to a. The reading of an ATL-path quantifier $\langle\!\langle A\rangle\!\rangle$ with commitment function ρ is as follows:

> $\langle\!\langle A\rangle\!\rangle_\rho\varphi$ states that, given the commitment of any agent b in $\mathsf{dom}(\rho)$ to use the strategy denoted by $\rho(b)$, the agents in $A \setminus \mathsf{dom}(\rho)$ have a strategy to ensure the temporal property φ, no matter what the agents in $\Sigma \setminus (\mathsf{dom}(\rho) \cup A)$ do.

Notice that the committed agents in $\mathsf{dom}(\rho)$ do not take part in the quantification over strategies in $\langle\!\langle A\rangle\!\rangle_\rho$.

We remark that $\langle\!\langle A\rangle\!\rangle_\rho$ is not how the path quantifier really looks like when used in a formula. The symbol ρ is merely a meta-logical reference to an actual commitment function, which is a collection of mappings of the form $a \mapsto \sigma_a$, where σ_a is a strategy term for agent a. This should be considered when analysing the length of a formula.

The notion of commitment to strategies requires the same strategies to be played again at a later stage. This means, in formulas of the form $\langle\!\langle A\rangle\!\rangle_\rho\Psi$, the same commitment $a \mapsto \sigma_a$ from ρ occurs in a commitment function ξ of a nested path quantifier $\langle\!\langle B\rangle\!\rangle_\xi$ in Ψ. Both ρ and ξ prescribe the strategy term σ_a for agent a (or, in both cases, a commits to σ_a). We have that $\rho(a) = \xi(a)$. Release of commitment to σ_a is modelled as easily as committing to it in the first place. This is achieved by having a commitment function χ of a nested path quantifier not include the commitment $a \mapsto \sigma_a$, i.e., either $\chi(a) \neq \sigma_a$ or χ is undefined for a. In case release of commitment is not desired, the notion of irrevocable strategies is used. It can be modelled explicitly in ATLES by only allowing commitment functions ρ to extend conservatively the commitment functions ξ under whose range they occur, i.e., ρ and ξ agree for all agents in $\mathsf{dom}(\xi)$. Thus, IATL can be defined in ATLES while avoiding the update semantics employed in [1].

The language of ATLES is defined over the extended signature $\langle \mathbf{\Pi}, \mathbf{\Sigma}, \mathbf{\Upsilon}\rangle$.

Definition 4 (ATLES **Syntax).** *The following grammar defines state formulas* φ *and path formulas* ψ*, where* p *ranges over* $\mathbf{\Pi}$*,* A *ranges over finite subsets of* $\mathbf{\Sigma}$ *and* ρ *over commitment functions. The language of* ATLES *consists of state formulas.*

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle\langle A \rangle\rangle_{\rho}\psi$$
$$\psi ::= \bigcirc\varphi \mid \Box\varphi \mid \varphi\,\mathcal{U}\,\varphi$$

The language of ATLES could easily be extended to allow for negation of the temporal operators next-time and until. We refrain from extending the syntax in this paper as we use the established complexity result of the satisfiability problem for ATLES from [19] in order to use ATLES to determine a decidable fragment of ATL_{sc} whose satisfiability can be solved in ExpTime.

Strategy terms in $\mathbf{\Upsilon}$ are interpreted as strategies in a CGS via assignments. An *assignment* $\mathfrak{a}$ in $\mathcal{C}$ is a function mapping strategy terms σ_a in $\mathbf{\Upsilon}_a$ for any agent a in $\mathbf{\Sigma}$ to a strategy $\mathfrak{a}(\sigma_a)$ in $\mathsf{strat}(a)$ for a in $\mathcal{C}$. Note that the assignment $\mathfrak{a}$ in a CGS acts like an assignment in First-order Logic with the difference that in ATLES strategy terms are mapped to actual strategies in the CGS instead of domain elements as in FOL. In [19] an assignment is called denotation function, which comes as a component of an ATS.

To define the semantics of ATLES, we use the notions of a strategy and outcome as in Section 2. We lift the notion of assignment to commitment functions as follows. The application of an assignment $\mathfrak{a}$ to a commitment function ρ is the set $\mathfrak{a}(\rho)$ of strategies for the agents in $\mathsf{dom}(\rho)$. Formally,

$$\mathfrak{a}(\rho) = \{f_a \in \mathsf{strat}(a) \mid f_a = \mathfrak{a}(\rho(a)), a \in \mathsf{dom}(\rho)\}.$$

It is readily checked that $\mathfrak{a}(\rho)$ is indeed a set of strategies, one for each agent in $\mathsf{dom}(\rho)$. To see this, recall that ρ is functional, i.e., it yields exactly one strategy term $\rho(a)$ for every agent for which ρ is defined.

An assignment $\mathfrak{a}$ acts as an interpretation of the commitment function ρ (i.e. the strategy terms in ρ). We can view a strategy term $\sigma_a = \rho(a)$, for any a in $\mathsf{dom}(\rho)$, as a constant rather than a variable. As we will see below in the semantics of ATLES, the assignment $\mathfrak{a}$ does not change during the evaluation of a formula and, thus, the strategy $\mathfrak{a}(\sigma_a)$ is fixed. We can think of the strategy term σ_a as being existentially quantified in the sense that there exists a strategy for a that is referenced by σ_a and provided by $\mathfrak{a}$. ATLES does not provide references to universally quantified strategies.

Using the notion of assignments, we can now define how to interpret the formulas of ATLES over CGSs.

Definition 5 (ATLES **Semantics).** *Given a CGS* $\mathcal{C} = \langle W, R, V, \Sigma, M, Mov, E\rangle$ *for* $\langle \Sigma, \Pi\rangle$ *and an assignment* $\mathfrak{a}$*, the consequence relation* $\models$ *is inductively defined as follows, and the notions of validity and satisfiability are defined as usual:*

- $\mathcal{C}, w \models^{\mathfrak{a}} p$ *iff* $w \in V(p)$*, for all atomic propositions* $p \in \Pi$*;*
- $\mathcal{C}, w \models^{\mathfrak{a}} \neg\varphi$ *iff* $\mathcal{C}, w \not\models^{\mathfrak{a}} \varphi$*;*
- $\mathcal{C}, w \models^{\mathfrak{a}} \varphi_1 \vee \varphi_2$ *iff* $\mathcal{C}, w \models^{\mathfrak{a}} \varphi_1$ *or* $\mathcal{C}, w \models^{\mathfrak{a}} \varphi_2$*;*

- $\mathcal{C}, w \models^{\mathfrak{a}} \langle\langle A \rangle\rangle_\rho \psi$ *iff there is a strategy* F_A *in* $\mathsf{strat}(A)$ *such that for all plays* $\lambda \in \mathsf{out}(w, S)$*, it holds that* $\mathcal{C}, \lambda \models^{\mathfrak{a}} \psi$*, where* $S = \mathfrak{a}(\rho) \circ F_A$*;*
- $\mathcal{C}, \lambda \models^{\mathfrak{a}} \bigcirc\varphi$ *iff* $\mathcal{C}, \lambda[1] \models^{\mathfrak{a}} \varphi$*;*
- $\mathcal{C}, \lambda \models^{\mathfrak{a}} \Box\varphi$ *iff* $\mathcal{C}, \lambda[i] \models^{\mathfrak{a}} \varphi$ *for all positions* $i \geq 0$*;*
- $\mathcal{C}, \lambda \models^{\mathfrak{a}} (\varphi_1 \mathcal{U} \varphi_2)$ *iff there is an* $i \geq 0$ *such that* $\mathcal{C}, \lambda[i] \models^{\mathfrak{a}} \varphi_2$ *and* $\mathcal{C}, \lambda[j] \models^{\mathfrak{a}} \varphi_1$ *for all positions* j *with* $0 \leq j < i$.

The ATLES semantics of $\langle\langle A \rangle\rangle_\rho$ is similar to the semantics of $\langle \cdot A \cdot \rangle$ in ATL_{sc}, which facilitates comparison. We recall that the operator $\circ$ from Section 2 yields $\mathfrak{a}(\rho) \circ F_A = \mathfrak{a}(\rho) \cup \{f_a \in F_A \mid a \notin \mathsf{dom}(\rho)\}$. Intuitively, $\mathfrak{a}(\rho) \circ F_A$ states that commitments of agents are respected as prescribed in ρ, all other agents in A play their just selected strategies.

3.2 Comparing ATL_{sc} and ATLES

Obvious differences between ATL_{sc} and ATLES are that, while the former includes a separate release operator $\cdot\rangle A \langle\cdot$ and a strategy context in the semantics, the latter allows to specify commitments in the form of $a \mapsto \sigma_a$ in the syntax which are interpreted using assignments. However, commitments and assignments in ATLES can play the roles of strategy release and strategy contexts in ATL_{sc}. A crucial difference between the logics is the semantics of the path quantifiers $\langle \cdot A \cdot \rangle$ and $\langle\langle A \rangle\rangle_\rho$. For $\langle \cdot A \cdot \rangle$, the strategies F_A selected by A upgrade or overwrite the strategy context F_{context} (cf. Def. 3), whereas, for $\langle\langle A \rangle\rangle_\rho$, the strategies $\mathfrak{a}(\rho)$ specified by the commitment ρ are supplemented by F_A (cf. Def. 5). The set of plays considered for further evaluation depends on the upgraded context $F_A \circ F_{\mathsf{context}}$ or the supplemented commitments $\mathfrak{a}(\rho) \circ F_A$. Both are not necessarily equivalent. The following proposition states under which conditions $\langle \cdot A \cdot \rangle$ and $\langle\langle A \rangle\rangle_\rho$ determine the same set $\mathsf{out}(x, S)$ of plays, where S is defined as $S = F_A \circ F_{\mathsf{context}}$ in the former case, and $S = \mathfrak{a}(\rho) \circ F_A$ in the latter.

Proposition 1. *It holds that* $F_A \circ F_{\mathsf{context}} = \mathfrak{a}(\rho) \circ F_A$ *if one of the following three conditions is satisfied:*

(i) $F_{\mathsf{context}} = \mathfrak{a}(\rho) = \emptyset$*;*
(ii) $F_A = \emptyset$ *and* $F_{\mathsf{context}} = \mathfrak{a}(\rho)$*; or*
(iii) $F_A = F_{\mathsf{context}} = \mathfrak{a}(\rho)$*.*

The proposition can be shown by using the fact that the strategy upgrade operator $\circ$ forms an idempotent semigroup on the set strat of strategies, and that $\circ$ is not commutative.

Proposition 1 makes clear that a strategy context F_{context} in ATL_{sc} corresponds to the strategy commitment $\mathfrak{a}(\rho)$ in ATLES with the difference that F_{context} is a purely semantic object, whereas $\mathfrak{a}(\rho)$ consists of a syntactic component ρ and a semantic component $\mathfrak{a}$. This means we can explicitly describe strategy contexts in the language of ATLES, whereas in ATL_{sc} we have to make use of $\langle \cdot A \cdot \rangle$ and $\cdot\rangle A \langle\cdot$ that describe that strategies for A are either pushed into the context or released from it. Notice how using strategy commitments in the syntax is more flexible than the strategy context model as every path quantifier in ATLES can be parameterised with a different commitment function, which describes explicitly which agent is using what strategy. In particular, this does not require a dedicated release operator.

The notion of *irrevocable strategies* is captured in ATL_{sc} by carefully avoiding quantification over strategies of committed agents. In ATLES, irrevocability can be made explicit in the syntax.

Once a strategy in the strategy context is overwritten with a new strategy or released, it cannot be recovered in ATL_{sc}, because any reference to it is lost. This could be described with the notion of *forgetting forever*. Not so in ATLES, where 'forgetting forever' can be modelled explicitly in the language, but it is no restriction of the logic as in ATL_{sc}. In fact, an agent in ATLES may *resume* a commitment after releasing it, which also captures a notion of agents having a *strategy memory.*

A strength of ATL_{sc} is to push *any* strategy that is available to an agent into the context. This is achieved with formulas of the form $\neg\langle\cdot A\cdot\rangle\psi$, where the agents in A quantify universally over their strategies F_A. In the semantics, before we continue with the evaluation of the path formula ψ, the strategies F_A are used to upgrade the strategy context (cf. Def. 3). This is another crucial difference to ATLES, which is restricted to existential quantification over commitments. To make more precise the relationship between ATL_{sc} and ATLES, we present an equivalence preserving mapping from a fragment of ATL_{sc} into ATLES. The fragment under consideration is represented by the set of ATL_{sc}-formulas where (i) a negated path subformula can only have the form $\neg(\top\,\mathcal{U}\,\varphi)$, and (ii) every $\langle\cdot A\cdot\rangle$ (for any A) is under the scope of an even number of negations. Let us denote $L(e)$ this language. Let us also denote $L(o)$ the language satisfying (i) but such that every $\langle\cdot A\cdot\rangle$ (for any A) is under the scope of an odd number of negations. We define the translation $tr(\cdot,\cdot)$ as follows:

$$
\begin{aligned}
tr(p,\xi) &\stackrel{\text{def}}{=} p;\\
tr(\neg\varphi_o,\xi) &\stackrel{\text{def}}{=} \neg tr(\varphi_o,\xi);\\
tr(\varphi_1\vee\varphi_2,\xi) &\stackrel{\text{def}}{=} tr(\varphi_1,\xi)\vee tr(\varphi_2,\xi);\\
tr(\cdot\rangle A\langle\cdot\varphi,\xi) &\stackrel{\text{def}}{=} tr(\varphi,\chi),\ \text{ where } \chi=\xi|_{\Sigma\setminus A};\\
tr(\langle\cdot A\cdot\rangle\bigcirc\varphi,\xi) &\stackrel{\text{def}}{=} \langle\!\langle A\rangle\!\rangle_\rho\bigcirc tr(\varphi,\rho);\\
tr(\langle\cdot A\cdot\rangle\Box\varphi,\xi) &\stackrel{\text{def}}{=} \langle\!\langle A\rangle\!\rangle_\rho\Box\, tr(\varphi,\rho);\\
tr(\langle\cdot A\cdot\rangle(\varphi_1\,\mathcal{U}\,\varphi_2),\xi) &\stackrel{\text{def}}{=} \langle\!\langle A\rangle\!\rangle_\rho(tr(\varphi_1,\rho)\,\mathcal{U}\,tr(\varphi_2,\rho)),
\end{aligned}
$$

where φ_o is in $L(o)$, φ, φ_1 and φ_2 are in $L(e)$, and where the commitment function ρ overwrites/updates ξ at A with fresh strategy terms. Formally,

$$\rho=\xi|_{\mathsf{dom}(\xi)\setminus A}\cup\{a\mapsto\sigma_a\mid a\in A,\sigma_a\text{ is fresh}\}.$$

The following proposition states that $tr(\cdot,\cdot)$ is indeed equivalence preserving. The proof works by induction on the structure of ATL_{sc}-formulas that are translated.

Proposition 2. *Let φ be a formula in $L(e)$, $\mathcal{C}$ a CGS, x a world in $\mathcal{C}$ and F a strategy in $\mathcal{C}$. The following are equivalent:*

(a) $\mathcal{C}, x \models_F \varphi$*;*
(b) $\mathcal{C}, x \models^{\mathfrak{a}} tr(\varphi, \rho_F)$*, for some* $\langle \rho_F, F\rangle$*-compatible assignment* $\mathfrak{a}$*,*

where $\rho_F = \{a \mapsto \sigma_a \mid f_a \in F, \sigma_a \text{ is fresh}\}$ *and an assignment* $\mathfrak{a}$ *is* $\langle \rho_F, F\rangle$-compatible *if* $\mathfrak{a}(\rho_F(a)) = f_a$*, for every* $a \in \mathsf{dom}(\rho_F)$ *and* $f_a \in F$*.*

The satisfiability checking problem for $L(e)$ can be solved in ExpTime by Proposition 2 and the fact that ATLES is in ExpTime [19]. This is in contrast with the complexity of full ATL_{sc}, which we establish in the following section.

4 Complexity

This section is devoted to investigating the computational complexity of ATL_{sc} and ATL^*_{sc}.

Generally, high expressiveness tends to come with the price of high computational complexity of reasoning problems. While the model checking problem was already considered in [6,4] (and shown to be between 2ExpTime-hard and non-elementary for ATL_{sc}, while it is 2ExpTime-complete for ATL^*, cf. [2]), we focus here on the satisfiability problem. The lower complexity bounds carry over to ATL_{sc} and ATL^*_{sc} from their respective fragments ATL and ATL^*. It turns out, however, that extending ATL with strategy contexts comes with a much higher price. In the following we show that ATL_{sc} is undecidable. In fact, we show this for the release-free fragment of ATL_{sc}. We use a reduction of the satisfiability problem for the product logic $\mathsf{S5}^n$, which is known to be undecidable. As we have hinted upon in the introduction, ATL_{sc} can capture some type of STIT actual group agency. Thus the undecidability of ATL_{sc} may not come as a surprise considering the undecidability of Chellas' STIT logic of group agency ([10]).

4.1 Product Logic S5

The language of $\mathsf{S5}^n$ is the basic propositional n-modal language given by the following grammar, where p ranges over $\mathbf{\Pi}$, and $i \in \{1, \ldots, n\}$:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \Diamond_i \varphi.$$

The semantic structures for $\mathsf{S5}^n$ are as follows. A *universal product S5*^{n}*-frame* is a tuple $\mathfrak{F} = (W_1 \times \cdots \times W_n, R_1, \ldots, R_n)$, where for every $i \in \{1, \ldots, n\}$, W_i is a non-empty set of worlds and R_i is the universal relation on W_i. As the relations R_i are determined by the sets W_i, we also denote such frames by $(W_1 \times \cdots \times W_n)$. An S5^n-model is a pair $\mathfrak{M} = (\mathfrak{F}, V)$, where $\mathfrak{F} = (W_1 \times \cdots \times W_n)$ is a universal product S5^n-frame and V a valuation in $\mathfrak{F}$ mapping every propositional variable p to a subset $V(p)$ of $W_1 \times \cdots \times W_n$. The consequence relation $\models_{\mathrm{S5}^n}$ is defined inductively between S5^n-models $\mathfrak{M}$, worlds $\boldsymbol{x} = \langle x_1, \ldots, x_n\rangle$ in $\mathfrak{M}$ and formulas of S5^n as follows:

- $(\mathfrak{M}, \boldsymbol{x}) \models p$ iff $\boldsymbol{x} \in V(p)$;
- $(\mathfrak{M}, \boldsymbol{x}) \models \neg\varphi$ iff $(\mathfrak{M}, \boldsymbol{x}) \not\models_{\mathrm{S5}^n} \varphi$;

– $(\mathfrak{M}, \boldsymbol{x}) \models \varphi \vee \psi$ iff $(\mathfrak{M}, \boldsymbol{x}) \models_{S5^n} \varphi$ or $(\mathfrak{M}, \boldsymbol{x}) \models_{S5^n} \psi$;
– $(\mathfrak{M}, \boldsymbol{x}) \models \Diamond_i \varphi$ iff there is a $y_i \in W_i$ such that $(\mathfrak{M}, \boldsymbol{y}) \models \varphi$, where $\boldsymbol{y} = \langle x_1, \ldots, x_{i-1}, y_i, x_{i+1}, \ldots, x_n \rangle$.

We make use of the following results.

Theorem 1. *The satisfiability problem for* $\mathsf{S5}^n$ *over finite models is*

(i) NExpTime-complete for $n = 2$*; and*
(ii) undecidable for all $n \geq 3$*.*

As $S5^2$ has the finite model property ([17]), (i) follows from Marx's result on the NExpTime-hardness of $S5^2$ ([16]). Undecidability of $\mathsf{S5}^n$, for $n \geq 3$, over arbitrary models was shown by Maddux ([15]) in an algebraic setting, via a reduction of the undecidable word problem of semigroups. As the word problem of all finite semigroups is also undecidable ([8]), Maddux's original proof actually shows the undecidability of $S5^n$ reasoning restricted to finite models (even though $S5^n$ lacks the finite model property for $n \geq 3$, cf. [13]). Another way of showing the undecidability of finite model reasoning with $S5^n$, for $n = 3$, is using Trakhtenbrot's theorem ([3, Section 2.1.2]). He showed how to encode the $\omega \times \omega$ grid and halting Turing machines in finite models, using the first-order language having binary predicates and 3 variables only. This language can be translated to $S5^3$ while keeping the models finite, using the Halmos-Johnson technique ([9,12], see also [7, Section 8.1]).[1]

4.2 Satisfiability of ATL_{sc}

Theorem 2. *The satisfiability problem for* ATL_{sc} *is*

(i) NP-hard for formulas with $n = 1$ *agent;*
(ii) NExpTime-hard for formulas with $n = 2$ *agents; and*
(iii) undecidable for formuals with $n \geq 3$ *agents.*

We show the lower complexity bounds in Theorem 2 by a reduction of the satisfiability problem for $\mathsf{S5}^n$ to the problem for ATL_{sc}. We leave the matching upper bounds for (i) and (ii) as open problems. Define inductively a translation $tr(\cdot)$ mapping $S5^n$-formulas to formulas of ATL_{sc} as follows:

$$\begin{aligned} tr(p) &\stackrel{\text{def}}{=} \langle\!\cdot \emptyset \cdot\!\rangle \bigcirc p; \\ tr(\neg\varphi) &\stackrel{\text{def}}{=} \neg tr(\varphi); \\ tr(\varphi \vee \psi) &\stackrel{\text{def}}{=} tr(\varphi) \vee tr(\psi); \\ tr(\Diamond_i \varphi) &\stackrel{\text{def}}{=} \langle\!\cdot i \cdot\!\rangle (\bot\, \mathcal{U}\, tr(\varphi)). \end{aligned}$$

Notice that the translation does not make use of the strategy release operator $\cdot\rangle A \langle\cdot$ of ATL_{sc}. Thus Theorem 2 holds already for the $\cdot\rangle A \langle\cdot$-free fragment of ATL_{sc}.

[1] We are grateful to Agi Kurucz for referencing and summarising these details for us.

Lemma 1. *Let φ be an $\mathsf{S5}^n$-formula and let Σ_φ be the set of agents that occur in φ. The following are equivalent:*

(i) φ is satisfiable wrt. $\mathsf{S5}^n$ in a finite model;
(ii) $\langle\cdot\Sigma_\varphi\cdot\rangle\bot\,\mathcal{U}\,tr(\varphi)$ is satisfiable wrt. ATL_{sc}.

Proof. "$(i) \Rightarrow (ii)$": Given a finite $\mathsf{S5}^n$-model $\mathfrak{M} = (\mathfrak{F}, V)$ with $\mathfrak{F} = (W_1 \times \cdots \times W_n)$, we construct a CGS $\mathcal{C}_\mathfrak{M} = \langle W_\mathfrak{M}, V_\mathfrak{M}, \Sigma_\mathfrak{M}, M_\mathfrak{M}, Mov_\mathfrak{M}, E_\mathfrak{M}\rangle$ as follows. Set:

- $W_\mathfrak{M} = W_1 \times \cdots \times W_n$;
- $V_\mathfrak{M}(\boldsymbol{w}) = \{p \mid \boldsymbol{w} \in V(p)\}$, for all $\boldsymbol{w} \in W_\mathfrak{M}$;
- $\Sigma_\mathfrak{M} = \{1, \ldots, n\}$;
- $Mov_\mathfrak{M}(\boldsymbol{w}, i) = \{\{\langle x_1, \ldots, x_n\rangle \mid x_j \in W_j \text{ for all } j \neq i\} \mid x_i \in W_i\}$, for all $\boldsymbol{w} \in W_\mathfrak{M}$ and all $i \in \Sigma_\mathfrak{M}$;
- $M_\mathfrak{M} = \bigcup_{i=1,\ldots,n} Mov_\mathfrak{M}(\boldsymbol{w}, i)$, for some arbitrary $\boldsymbol{w} \in W_\mathfrak{M}$; and
- $E_\mathfrak{M}(\boldsymbol{w}, \boldsymbol{m}) \in \bigcap_{i=1,\ldots,n} m_i$, for all $\boldsymbol{w} \in W_\mathfrak{M}$ and all $\boldsymbol{m} = \langle m_1, \ldots, m_n\rangle \in \mathsf{prof}(\boldsymbol{w})$,

where $\mathsf{prof}(\boldsymbol{w}) = \{\langle m_1, \ldots, m_n\rangle \mid m_i \in Mov_\mathfrak{M}(\boldsymbol{w}, i)\}$. It is readily checked that $\mathcal{C}_\mathfrak{M}$ is indeed a CGS. To see this, note that for all $\boldsymbol{m} = \langle m_1, \ldots, m_n\rangle \in \mathsf{prof}(w)$, each m_i is a subset of $W_\mathfrak{M}$, and verify that the intersection $\bigcap_{i=1,\ldots,n} m_i$ is a singleton set.

Given a world $\boldsymbol{x} = \langle x_1, \ldots, x_n\rangle$ in $\mathfrak{M}$, set the strategy $F_{\boldsymbol{x}} = \{f_1^{\boldsymbol{x}}, \ldots, f_n^{\boldsymbol{x}}\}$ for the agents in $\Sigma_\mathfrak{M}$ as follows: For all $i = 1, \ldots, n$ and all $\boldsymbol{w} \in W_\mathfrak{M}$,

$$f_i^{\boldsymbol{x}}(\boldsymbol{w}) = \{\langle y_1, \ldots, y_n\rangle \mid y_i = x_i, y_j \in W_j \text{ for all } j \neq i\}.$$

$F_{\boldsymbol{x}}$ is indeed a strategy for $\Sigma_\mathfrak{M}$ as $f_i^{\boldsymbol{x}}(\boldsymbol{w}) \in Mov(\boldsymbol{w}, i)$, for all $i = 1, \ldots, n$ and $\boldsymbol{w} \in W_\mathfrak{M}$. Note that $F_{\boldsymbol{x}}$ specifies the same complete move profile $\langle f_1^{\boldsymbol{x}}(\boldsymbol{w}), \ldots, f_n^{\boldsymbol{x}}(\boldsymbol{w})\rangle$ at every state $\boldsymbol{w}$ in $\mathcal{C}_\mathfrak{M}$.

To show this direction of the lemma, it is sufficient to show that, for all $\mathsf{S5}^n$-formulas φ, all $\mathsf{S5}^n$-models and worlds $\boldsymbol{x}$ in $\mathfrak{M}$ and states $\boldsymbol{w}$ in $\mathcal{C}_\mathfrak{M}$:

$$\mathfrak{M}, \boldsymbol{x} \models_{\mathsf{S5}^n} \varphi \text{ iff } \mathcal{C}_\mathfrak{M}, \boldsymbol{w} \models_{F_{\boldsymbol{x}}} tr(\varphi). \tag{1}$$

$F_{\boldsymbol{x}}$ specifies the same state $E_\mathfrak{M}(\boldsymbol{w}, F_{\boldsymbol{x}}) = \boldsymbol{x}$ as successor of any state $\boldsymbol{w}$ in $\mathcal{C}_\mathfrak{M}$. It follows that the set $\mathsf{out}(\boldsymbol{w}, F_{\boldsymbol{x}})$ consists of exactly one play λ such that $\lambda[i] = \boldsymbol{x}$, for all positions $i \geq 0$. Together with the right-hand side of (1), this implies that $\mathcal{C}_\mathfrak{M}, \boldsymbol{w} \models_S \langle\cdot\Sigma_\varphi\cdot\rangle\Box tr(\varphi)$ and $\mathcal{C}_\mathfrak{M}, \boldsymbol{w} \models_S \langle\cdot\Sigma_\varphi\cdot\rangle\bot\,\mathcal{U}\,tr(\varphi)$, for any strategy S for $\Sigma_\mathfrak{M}$. Hence, the left-to-right direction of the lemma follows.

To show (1), we proceed by induction on the structure of φ. In the induction base, φ is a proposition p. The following equivalences hold: $\mathfrak{M}, \boldsymbol{x} \models_{\mathsf{S5}^n} p$ iff $\boldsymbol{x} \in V(p)$ iff $p \in V_\mathfrak{M}(\boldsymbol{x})$ iff, for every state $\boldsymbol{w}$ in $\mathcal{C}_\mathfrak{M}$, $\mathcal{C}_\mathfrak{M}, \boldsymbol{w} \models_{F_{\boldsymbol{x}}} \langle\cdot\emptyset\cdot\rangle\bigcirc p$. For the induction step, assume that we have already shown the induction hypothesis for φ. Consider the following case of the induction step (we omit the Boolean cases):

- $\varphi = \Diamond_i\psi$. Then: $\mathfrak{M}, \boldsymbol{x} \models_{\mathsf{S5}^n} \Diamond_i\psi$ iff there is a $y_i \in W_i$ such that $\mathfrak{M}, \boldsymbol{x'} \models_{\mathsf{S5}^n} \psi$, where $\boldsymbol{x'} = \langle x_1, \ldots, x_{i-1}, y_i, x_{i+1}, \ldots, x_n\rangle$. By the induction hypothesis, this is equivalent to $\mathcal{C}_{\mathfrak{M}}, \boldsymbol{w} \models_{F_{\boldsymbol{x}'}} tr(\psi)$ (for all $\boldsymbol{w}$). The strategy $f_i^{\boldsymbol{x}'}(\boldsymbol{w})$ is a move in $Mov_{\mathfrak{M}}(\boldsymbol{w}, a)$ available to agent i at any state $\boldsymbol{w}$ in $W_{\mathfrak{M}}$. Since $F_{\boldsymbol{x}}$ and $F_{\boldsymbol{x}'}$ differ at most in their i-th component, we have that $\mathcal{C}_{\mathfrak{M}}, \boldsymbol{w} \models_{F_{\boldsymbol{x}}} \langle\!\cdot i \cdot\!\rangle \bot \,\mathcal{U}\, tr(\psi)$, which is equivalent to $\mathcal{C}_{\mathfrak{M}}, \boldsymbol{w} \models_{F_{\boldsymbol{x}}} tr(\Diamond_i \psi)$.

This finishes the induction and, thus, this direction of the proof.

"$(ii) \Rightarrow (i)$": Given a CGS $\mathcal{C} = \langle W, V, \Sigma, M, Mov, E\rangle$ for $\langle \Sigma, \Pi\rangle$ with $\Sigma = \{1, \ldots, n\}$ and a world x in $\mathcal{C}$, construct an $\mathsf{S5}^n$-model $\mathfrak{M}_{(\mathcal{C},x)} = (\mathfrak{F}_{(\mathcal{C},x)}, V_{(\mathcal{C},x)})$ with $\mathfrak{F}_{(\mathcal{C},x)} = (W_1^{(\mathcal{C},x)} \times \cdots \times W_n^{(\mathcal{C},x)})$ as follows. Set:

- $W_i^{(\mathcal{C},x)} = Mov(x, i)$ for all $i \in \Sigma$; and
- $V_{(\mathcal{C},x)}(p) = \{\boldsymbol{m} \in W_1^{(\mathcal{C},x)} \times \cdots \times W_n^{(\mathcal{C},x)} \mid p \in V(E(x, \boldsymbol{m}))\}$, for all $p \in \Pi$.

It is readily checked that $\mathfrak{M}_{(\mathcal{C},x)}$ is indeed a finite $\mathsf{S5}^n$-model. While a move profile determines a unique successor $E(x, \boldsymbol{m})$ at a state x, two move profiles $\boldsymbol{m}_1 \neq \boldsymbol{m}_2$ may be mapped to the same successor, i.e. $E(x, \boldsymbol{m}_1) = E(x, \boldsymbol{m}_2)$. However, in the product model $\mathfrak{M}_{(\mathcal{C},x)}$ the move profiles $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ are different worlds. Let $F_\Sigma = \{f_1, \ldots, f_n\}$ be a strategy in $\mathcal{C}$ for the agents in Σ. A world $\boldsymbol{m}$ in $\mathfrak{M}_{(\mathcal{C},x)}$ is called an $F_{\Sigma,x}$-*world* if $E(x, \boldsymbol{m}) = \lambda[1]$ with $\{\lambda\} = \mathsf{out}(x, F_{\Sigma,x})$.

To show this direction of the lemma, it is sufficient to show that, for all S5^n-formulas φ, for all CGSs $\mathcal{C}$, all worlds x in $\mathcal{C}$ and all strategies F for Σ, and all F-worlds $\boldsymbol{w}_F$ in $\mathfrak{M}_{(\mathcal{C},x)}$:

$$\mathcal{C}, x \models_F tr(\varphi) \text{ iff } \mathfrak{M}_{(\mathcal{C},x)}, \boldsymbol{w}_F \models_{\mathsf{S5}^n} \varphi. \tag{2}$$

Then, the right-to-left direction of the lemma follows from (2) together with the fact that $\mathcal{C}, x \models_S \langle\!\cdot \Sigma_\varphi \cdot\!\rangle tr(\varphi)$ implies $\mathcal{C}, x \models_F tr(\varphi)$, for any strategy S.

To show (2), we proceed by induction on the structure of φ. In the induction base, φ is a proposition p. The following equivalences hold: $\mathcal{C}, x \models_F tr(p)$ iff $\mathcal{C}, x \models_F \langle\!\cdot \emptyset \cdot\!\rangle \bigcirc p$ iff $\mathcal{C}, y \models_F p$, where $y = \mathsf{out}(x, F)$ iff $p \in V(y)$ iff $\boldsymbol{w}_F \in V_{(\mathcal{C},x)}(p)$ iff $\mathfrak{M}_{(\mathcal{C},x)}, \boldsymbol{w}_F \models_{\mathsf{S5}^n} p$. For the induction step, assume that we have already shown the induction hypothesis for φ. Again, we skip the Boolean cases and proceed with the interesting case:

- $\varphi = \Diamond_i \psi$. Then: $\mathcal{C}, x \models_F tr(\Diamond_i \psi)$ iff $\mathcal{C}, x \models_F \langle\!\cdot i \cdot\!\rangle \bot \,\mathcal{U}\, tr(\psi)$ iff there is a strategy f_i such that it holds that $\mathcal{C}, x \models_S tr(\psi)$, with $S = \{f_i\} \cup \{f_b \in F \mid b \neq i\}$. By the induction hypothesis, we obtain $\mathfrak{M}_{(\mathcal{C},x)}, \boldsymbol{w}_S \models_{\mathsf{S5}^n} \psi$. Since F and S are identical with the possible exception of the strategy f_i for agent i, the worlds $\boldsymbol{w}_S$ and $\boldsymbol{w}_F$ differ at most in their i-th component. We have that $\mathfrak{M}_{(\mathcal{C},x)}, \boldsymbol{w}_F \models_{\mathsf{S5}^n} \Diamond_i \psi$. The other direction of this case can be shown similarly.

□

Corollary 1. *The satisfiability problem of any variant of* ATL *with strategy contexts in [4] is undecidable.*

Acknowledgements. We thank Agi Kurucz for her input on the undecidability of finite model reasoning of the product logic $\mathsf{S5}^n$. We are indebted to the reviewers of LAMAS 2012 and the workshop "Modeling Strategic Reasoning" at the Lorentz Center in Leiden, The Netherlands, 20–24 February 2012. The first author is supported by a Marie Curie COFUND fellowship, and the second author by a Juan de la Cierva fellowship of Spain and the project Agreement Technologies (Grant CONSOLIDER CSD2007-0022, INGENIO 2010), and the MICINN projects TIN2006-15455 and TIN2009-14562-CO5.

References

1. Ågotnes, T., Goranko, V., Jamroga, W.: Alternating-time temporal logics with irrevocable strategies. In: Proceedings of TARK 2007, pp. 15–24. ACM (2007)
2. Alur, R., Henzinger, T.A., Kupferman, O.: Alternating-time temporal logic. The Journal of the ACM 49(5), 672–713 (2002)
3. Börger, E., Grädel, E., Gurevich, Y.: The Classical Decision Problem. In: Perspectives Mathematical Logic. Springer (1997)
4. Brihaye, T., Da Costa, A., Laroussinie, F., Markey, N.: ATL with Strategy Contexts and Bounded Memory. In: Artemov, S., Nerode, A. (eds.) LFCS 2009. LNCS, vol. 5407, pp. 92–106. Springer, Heidelberg (2008)
5. Broersen, J., Herzig, A., Troquard, N.: A STIT-Extension of ATL. In: Fisher, M., van der Hoek, W., Konev, B., Lisitsa, A. (eds.) JELIA 2006. LNCS (LNAI), vol. 4160, pp. 69–81. Springer, Heidelberg (2006)
6. Da Costa, A., Laroussinie, F., Markey, N.: ATL with strategy contexts: Expressiveness and model checking. In: Proceedings of FSTTCS 2010. LIPIcs, vol. 8, pp. 120–132. Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik (2010)
7. Gabbay, D., Kurucz, A., Wolter, F., Zakharyaschev, M.: Many-dimensional modal logics: theory and applications. Studies in Logic, vol. 148. Elsevier Science (2003)
8. Gurevich, Y., Lewis, H.R.: The word problem for cancellation semigroups with zero. The Journal of Symbolic Logic 49(1), 184–191 (1984)
9. Halmos, P.: Algebraic logic, IV. Transactions of the AMS 86, 1–27 (1957)
10. Herzig, A., Schwarzentruber, F.: Properties of logics of individual and group agency. In: Proceedings of AiML 2008, pp. 133–149. College Publications (2008)
11. Horty, J.F.: Agency and Deontic Logic. Oxford University Press, Oxford (2001)
12. Johnson, J.: Nonfinitizability of classes of representable polyadic algebras. The Journal of Symbolic Logic 34(3), 344–352 (1969)
13. Kurucz, A.: S5 x S5 x S5 lacks the finite model property. In: Proceedings of AiML 2002, pp. 321–327. World Scientific (2002)
14. Laroussinie, F., Markey, N., Oreiby, G.: On the expressiveness and complexity of ATL. Logical Methods in Computer Science 4(2) (2008)
15. Maddux, R.: The equational theory of CA_3 is undecidable. The Journal of Symbolic Logic 45(2), 311–316 (1980)
16. Marx, M.: Complexity of products of modal logics. Journal of Logic and Computation 9(2), 197–214 (1999)
17. Mortimer, M.: On languages with two variables. Zeitschrift für Mathematische Logik und Grundlagen der Mathematik 21, 135–140 (1975)
18. Troquard, N., Walther, D.: ATL with contexts: agency and explicit strategies. In: Proceedings of LAMAS@AAMAS 2012 (2012)
19. Walther, D., van der Hoek, W., Wooldridge, M.: Alternating-time temporal logic with explicit strategies. In: Proceedings of TARK 2007, pp. 269–278. ACM (2007)