Alternating-time Temporal Logics with Irrevocable Strategies

Thomas Ågotnes
Dept. of Computer Engineering
Bergen University College,
Bergen, Norway
tag@hib.no

Valentin Goranko
School of Mathematics
Univ. of the Witwatersrand
Johannesburg, South Africa
goranko@maths.wits.ac.za

Wojciech Jamroga
Department of Informatics
Clausthal Univ. of Technology
Clausthal, Germany
wjamroga@in.tu-clausthal.de

Abstract

In Alternating-time Temporal Logic (ATL), one can express statements about the strategic ability of an agent (or a coalition of agents) to achieve a goal ϕ such as: "agent i can choose a strategy such that, if i follows this strategy then, no matter what other agents do, ϕ will always be true". However, strategies in ATL are *revocable* in the sense that in the evaluation of the goal ϕ the agent i is no longer restricted by the strategy she has chosen in order to reach the state where the goal is evaluated. In this paper we consider alternative variants of ATL where strategies, on the contrary, are *irrevocable.* The difference between revocable and irrevocable strategies shows up when we consider the ability to achieve a goal which, again, involves (nested) strategic ability. Furthermore, unlike in the standard semantics of ATL, memory plays an essential role in the semantics based on irrevocable strategies.

1 Introduction

Logics for game-like scenarios have received much interest recently [1, 5, 6, 8], for example as a part of the foundations of multi-agent systems. Alternating-time Temporal Logic ATL [1] is probably the most popular logic of this kind now. ATL is an extension of the Computational Tree Logic CTL [2], one of the most successful temporal logics in computer science. The main semantic assumption behind ATL is that a system at a given time is in one of several possible states, and that the next state of the system is determined by the current state and the actions chosen by each of κ *agents* present in the system. ATL involves strategic quantifiers (called *cooperation modalities*), such as $\langle\langle C\rangle\rangle\mathcal{X}$ and $\langle\langle C\rangle\rangle\mathcal{G}$ where C is a set of agents. Formula $\langle\langle C\rangle\rangle\mathcal{X}\phi$ is intended to mean that coalition C can achieve ϕ in the ne$\mathcal{X}$t state of the system, or, in more detail, that the agents in C can choose their strategies so that, if they use these strategies then ϕ will be true in the next state – no matter what the agents outside C do. Similarly, $\langle\langle C\rangle\rangle\mathcal{G}\phi$ means that C can force ϕ to be true in all future states ($\mathcal{G}$lobally).

An interesting feature of ATL is that strategies in the logic are *revocable*, in the sense that in the evaluation of the goal ϕ an agent $i \in C$ is no longer restricted by the strategy she has chosen. That is, if ϕ includes a nested cooperation modality for a coalition including i, then i is again free to choose any strategy to demonstrate the truth of ϕ. This is very much in agreement with the semantics of CTL path quantifiers, where it is natural to express facts like "there is a path, such that the system can always deviate from the path to another path which satisfies ϕ" ($\mathbf{E}\mathcal{G}\mathbf{E}\phi$). On a more general level, this reflects the way in which quantifiers are treated in classical mathematical logic: in the formula $\exists x(x = 1 \wedge \exists x\ x = 0)$, the second occurrence of $\exists x$ supersedes the first one in its scope of binding, and in consequence the formula is true in any sensible arithmetics. The semantics of the strategic quantifiers in ATL works similarly, and there are many scenarios, where one would like to talk about strategies and abilities exactly in this way, too. However, it somehow contradicts the usual game-theoretical view of a strategy as a conditional plan that *completely* specifies the agent's future behavior. In this paper, we focus on the latter view, and consider a variant of ATL, in which strategies are *irrevocable* – they are chosen once and forever.

We begin by recalling the language and semantics of ATL, and an informal discussion on possible semantics of strategic quantifiers. Then, we present and study our alternative semantics of cooperation modalities in a formal way. Validity, satisfiability, and model checking problems are discussed in the subsequent sections. Irrevocable strategies are also used in the context of

ATL in [7], but in a different way. The relationship is discussed in Section 9.

2 Preliminaries: Alternating-time Temporal Logic

Here we recall briefly the semantic and syntactic basics of ATL. For full details, see any of [1, 3, 4].

A *concurrent game structure* (CGS) is a tuple $M = (\Sigma, \Pi, Q, \pi, \mathit{Act}, d, \delta)$ where $\Sigma = \{1, \ldots, \kappa\}$, for some $\kappa > 0$, is the set of *agents (players)*; Π is a set of *atomic propositions*; Q is a set of *states*; $\pi : Q \to \wp(\Pi)$ is the *labeling* function; Act is a (usually finite) set of *actions*; d is a mapping such that for each player $i \in \Sigma$ and state $q \in Q$, $d_i(q) \subseteq \mathit{Act}$ is the non-empty set of actions available to player i in q; and δ is the *transition function*, mapping every pair $(q \in Q, \alpha \in D(q))$, where $D(q) = d_1(q) \times \cdots \times d_\kappa(q)$ is the set of *joint actions* at q, to an *outcome state* $\delta(q, \alpha) \in Q$.

For $\alpha \in D(q)$, let α_i denote the i-th component of α. Likewise, $\alpha(C)$ denotes the projection of α onto $C \subseteq \Sigma$, and $D(q, C)$ denotes the projection of $D(q)$ onto C. Whenever necessary, we may write $D(M, q)$ and $D(M, q, C)$ in order to indicate the CGS explicitly. A *pointed CGS* is a pair (M, q) where M is a CGS and q is a state in M.

Given a CGS M, $C \subseteq \Sigma$, a state q in M, and a tuple of actions $\alpha(C) \in D(q, C)$, one for each agent in C, we denote by $\mathit{out}(M, q, \alpha(C))$ the set of outcome states of all joint actions extending $\alpha(C)$. Formally,

$$\mathit{out}(M, q, \alpha(C)) = \\ \{\delta(q, \alpha') \mid \alpha' \in D(q), \text{ and } \alpha'(C) = \alpha(C)\}.$$

Thus, in particular, $\mathit{out}(M, q, \alpha) = \{\delta(q, \alpha)\}$. Also, we simply write $\mathit{out}(M, q)$ for the set $\mathit{out}(M, q, \alpha(\emptyset))$ of all possible outcome states from q; when M is fixed, we simply write $\mathit{out}(q)$.

A *computation* λ is an infinite sequence of states; $\lambda = q_0 q_1 \cdots$, where for each $j \geq 0$ there is a joint action $\alpha \in D(q_j)$ such that $\delta(q_j, \alpha) = q_{j+1}$. By $\lambda[j]$ we will denote the element (q_j) in λ with index j ; respectively $\lambda|j$ will denote the initial segment of λ ending with $\lambda[j]$. Such an initial segment will be called a *finite computation.* The last state of a finite computation σ will be denoted by $l(\sigma)$. Without risk of confusion, whenever suitable we will regard a state q as a one step computation, and then $l(q) = q$.

A simple (or memoryless) *strategy* for a player i is a function $f_i : Q \to \mathit{Act}$ with $f_i(q) \in d_i(q)$ for each $q \in Q$. That is, the strategy maps each state to an action for player i. A *memory-based strategy* for a player i is a function $f_i : Q^+ \to \mathit{Act}$ with $f_i(\sigma) \in d_i(l(\sigma))$, i.e., it maps possible histories of the play to i's choices. Clearly, memoryless strategies can be seen as special cases of memory-based strategies, where $f_i(q_1 \ldots q_n)$ only depends on the last state q_n.

A *joint strategy* for $C \subseteq \Sigma$ is a tuple of strategies, one per $i \in C$; by $\mathit{Str}(M, C)$ we denote the set of joint memoryless strategies for C in M. We then denote by $f_C(q)$ the tuple of respective actions $f_i(q)$ for $i \in C$, and adopt the notation $\mathit{out}(M, q, f_C)$ for $\mathit{out}(M, q, f_C(q))$. Given a state q and a joint memoryless strategy f_C for C, $\mathit{comp}(M, q, f_C)$ denotes the set of possible computations starting in state q where the agents in C use the strategies f_C. Formally, $\lambda \in \mathit{comp}(M, q, f_C)$ iff $\lambda[0] = q$ and for all $j \geq 0$, $\lambda[j+1] \in \mathit{out}(M, \lambda[j], f_C)$. For the set $\mathit{comp}(M, q, \emptyset)$ of possible computations starting in state q of CGS M, we simply write $\mathit{comp}(M, q)$. The sets of *finite* computations starting from q, $\mathit{fincomp}(M, q, f_C)$, in particular $\mathit{fincomp}(M, q)$, are defined likewise.

A language for ATL is determined by the set of atomic propositions Π and the set of agents Σ, and will be denoted by $\text{ATL}(\Pi, \Sigma)$. The formulae of $\text{ATL}(\Pi, \Sigma)$ are defined recursively as follows:

$$\phi ::= \\ \top \mid p \mid \neg\phi \mid \phi \wedge \phi \mid \langle\langle C \rangle\rangle \mathcal{X}\phi \mid \langle\langle C \rangle\rangle \mathcal{G}\phi \mid \langle\langle C \rangle\rangle \phi\mathcal{U}\phi$$

where $p \in \Pi$ and $C \subseteq \Sigma$.

We use the standard derived propositional connectives, in addition to $\langle\langle C \rangle\rangle \mathcal{F}\phi$ for $\langle\langle C \rangle\rangle (\top\mathcal{U}\phi)$, and sometimes write $\mathbf{A}\mathcal{X}\phi$, $\mathbf{A}\mathcal{G}\phi$, $\mathbf{A}(\phi_1\mathcal{U}\phi_2)$ respectively for $\langle\langle\emptyset\rangle\rangle\mathcal{X}\phi$, $\langle\langle\emptyset\rangle\rangle\mathcal{G}\phi$ and $\langle\langle\emptyset\rangle\rangle(\phi_1\mathcal{U}\phi_2)$, and $\mathbf{E}\mathcal{X}\phi$, $\mathbf{E}\mathcal{G}\phi$, $\mathbf{E}(\phi_1\mathcal{U}\phi_2)$ respectively for $\langle\langle\Sigma\rangle\rangle\mathcal{X}\phi$, $\langle\langle\Sigma\rangle\rangle\mathcal{G}\phi$ and $\langle\langle\Sigma\rangle\rangle(\phi_1\mathcal{U}\phi_2)$. We also write $[\![C]\!]\mathcal{X}\phi$ and $[\![C]\!]\mathcal{G}\phi$ respectively for the duals $\neg\langle\langle C \rangle\rangle\mathcal{X}\neg\phi$ and $\neg\langle\langle C \rangle\rangle\mathcal{F}\neg\phi$.

Truth of a formula ψ in a state q of a CGS M is defined via the standard clauses for the Boolean connectives and the following clauses for the strategic temporal operators:[1]

$$M, q \models_{\text{ATL}} \langle\langle C \rangle\rangle \mathcal{X}\phi \Leftrightarrow \\ \exists f_C \in \mathit{Str}(M, C) \forall\lambda \in \mathit{comp}(M, q, f_C) \\ (M, \lambda[1] \models_{\text{ATL}} \phi)$$

$$M, q \models_{\text{ATL}} \langle\langle C \rangle\rangle \mathcal{G}\phi \Leftrightarrow \\ \exists f_C \in \mathit{Str}(M, C) \forall\lambda \in \mathit{comp}(M, q, f_C) \forall j \geq 0 \\ (M, \lambda[j] \models_{\text{ATL}} \phi)$$

$$M, q \models_{\text{ATL}} \langle\langle C \rangle\rangle \phi_1\mathcal{U}\phi_2 \Leftrightarrow \\ \exists f_C \in \mathit{Str}(M, C) \forall\lambda \in \mathit{comp}(M, q, f_C) \exists j \geq 0 \\ ((M, \lambda[j] \models_{\text{ATL}} \phi_2) \text{ and} \\ \forall 0 \leq k < j (M, \lambda[k] \models_{\text{ATL}} \phi_1)).$$

[1] We deviate from the original semantics of ATL [1] in that we use memoryless rather than memory-based strategies. However, both types of strategies yield equivalent semantics for "pure" ATL [6].

We further say that: ϕ is ATL-*valid in* M (denoted $M \models_{\text{ATL}} \phi$) if $M, q \models_{\text{ATL}} \phi$ for every $q \in Q$; ϕ is ATL-*valid* ($\models_{\text{ATL}} \phi$) if $M \models_{\text{ATL}} \phi$ for every CGS M.

3 Revocable vs. Irrevocable Strategies

The CTL heritage in particular – and compositionality of the ATL semantics on a more abstract level – imply that agents' strategies are *revocable* in ATL, as the following example demonstrates.

Example 1 *We are given a system with a shared resource, and are interested in reasoning about whether agent a has access to the resource. Let p denote the fact that agent a controls the resource. The* ATL *formula $\langle\langle a\rangle\rangle\mathcal{X}p$ expresses the fact that a is able to obtain control of the resource in the next moment, if she chooses to. Now imagine that agent a does not need to access the resource all the time, but she would like to be able to control the resource any time she needs it. This can be expressed in* ATL *by formula $\langle\langle a\rangle\rangle\mathcal{G}\langle\langle a\rangle\rangle\mathcal{X}p$, saying that a has a strategy which guarantees that, in any future state of the system, a can always force the next state to be one where a controls the resource.*

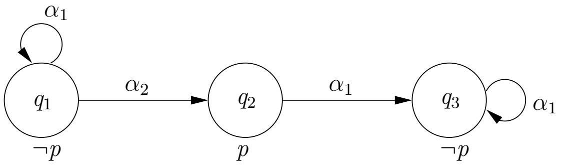

Figure 1: System M_0 with a single agent a. The transitions between states are labeled by the actions chosen by a.

Consider system M_0 from Figure 1. We have that $M_0, q_1 \models \langle\langle a\rangle\rangle\mathcal{X}p$: a can choose action α_2, which guarantees that p is true next. But we also have that $M_0, q_1 \models \langle\langle a\rangle\rangle\mathcal{G}\langle\langle a\rangle\rangle\mathcal{X}p$: a's strategy in this case is to always choose α_1, which guarantees that the system will stay in q_1 forever and, as we have seen, $M_0, q_1 \models \langle\langle a\rangle\rangle\mathcal{X}p$. However, this system does not have the exact property we had in mind because, by following that strategy, the agent a dooms herself to never access the resource *– in which case it is maybe counterintuitive that $\langle\langle a\rangle\rangle\mathcal{X}p$ should be true. In other words, a can ensure that she is forever* able *to access the resource – but only by never* actually *accessing it. Indeed, while a can force the* possibility *of achieving p to be true forever, the actual achievement of p destroys that possibility.*

The above example shows that strategies can be revoked by agents in ATL – and they are bound to be this way, due to the compositionality of ATL semantics. That is, the set of states that satisfy $\langle\langle a\rangle\rangle\mathcal{X}\phi$ depends only on the extensions of its parts (ϕ in this case), and the way in which those parts are combined ($\langle\langle a\rangle\rangle\mathcal{X}$ in this case). In particular, it does *not* depend on the semantic choices made to evaluate subformula ϕ, or a larger formula, of which $\langle\langle a\rangle\rangle\mathcal{X}\phi$ might be a part itself. Philosphically, this corresponds to strategies that are not "committed"; the agents merely *intend* to execute them, but they are free to change their minds as soon as the next choice point (i.e., next cooperation modality) is encountered.

On the other hand, a strategy in game theory is usually understood as a *complete* plan that prescribes the player's behaviour in all conceivable situations, and for all future moments. In this view, it is somehow counterintuitive to maintain that an agent behaves according to one strategy, and yet she may start behaving in a different way. Under this interpretation, $\langle\langle a\rangle\rangle\mathcal{G}\langle\langle a\rangle\rangle\mathcal{X}p$ *cannot* hold in M_0, q_1, because a can only play according to the first or the second strategy, but not to both of them at the same time. Irrevocable strategies are often naturally assumed, not only in game theory, but also in controller synthesis (the controller is an irrevocable strategy), and in planning in AI, where the actual achievement of certain subgoals may affect the possibility of achieving other subgoals.

Note that the first interpretation (strategies as intentions) views strategies as entities internal to, and fully controllable by the agent. The second interpretation corresponds to an objective view of the agent's behaviour, as perceived by a fully omniscient observer. Alternatively, we can see such strategies as ones that the agents committed to execute; then, they are internal to the players, but not controllable by them any more.

We do not imply that the ATL meaning of formulae like $\langle\langle a\rangle\rangle\mathcal{G}\langle\langle a\rangle\rangle\mathcal{X}p$ in ATL is "wrong" or not useful. However, we believe that there are many situations in which the "committed" interpretation of strategies is more appropriate. In this paper, we introduce alternative semantics for strategic quantifiers based on this assumption, and study properties of the resulting logics.

Remark 2 *It turns out that, unlike the standard semantics for* ATL*, memory plays essential role in the semantics based on irrevocable strategies. Thus, two natural variations of* ATL *emerge:* IATL*, referring to memoryless irrevocable strategies, and* MIATL*, referring to memory-based irrevocable strategies. There is a range of other interesting cases, e.g. one with strategies based on* finite memory*, and another, where agents must only adhere to their chosen strategies until their goal is fulfilled, e.g., only for one step if the* *goal is $X\phi$; until ϕ becomes true if it is $F\phi$; forever, if it is $G\phi$. For lack of space, these will not be discussed here, and we will mainly focus on* IATL.

4 Model Updates and Semantics with Irrevocable Strategies

We start by considering the memoryless case of irrevocable strategies. First, let us spell out again the intuition behind the semantics with irrevocable strategies: a formula such as $\langle\langle C\rangle\rangle\mathcal{X}\phi$ means that there is a joint strategy for C such that, if the actions of the agents from C are thereafter fixed by the choices prescribed by the strategy, ϕ will necessarily be true in the next state; likewise for the other temporal operators. This motivates the notion of a *model update*, similar to the one in [9, 7]: the update of a model by a joint memoryless strategy for C is a model obtained by fixing the choices of every agent in C in each state as prescribed by the strategy for that agent in that state.

Definition 3 (Model Update) *Let M be a CGS, C a coalition, and $f_C \in Str(M, C)$ a memoryless strategy. The update of M by f_C, denoted $M \dagger f_C$, is the same as M, except that the choices of each agent $i \in C$ are fixed by the strategy f_i: $d_i(q) = \{f_i(q)\}$ for each state q.*

The language of the logic IATL – ATL *with irrevocable strategies* – is the same as the language of ATL. Let q be a state in a CGS M. The semantics of the strategic operators in IATL is defined as follows:

$$M, q \models_{\text{IATL}} \langle\langle C\rangle\rangle\mathcal{X}\phi \Leftrightarrow \\ \exists f_C \in Str(M, C)\forall\lambda \in comp(M \dagger f_C, q, f_C) \\ (M \dagger f_C, \lambda[1] \models_{\text{IATL}} \phi)$$

$$M, q \models_{\text{IATL}} \langle\langle C\rangle\rangle\mathcal{G}\phi \Leftrightarrow \\ \exists f_C \in Str(M, C)\forall\lambda \in comp(M \dagger f_C, q, f_C) \\ \forall j \geq 0(M \dagger f_C, \lambda[j] \models_{\text{IATL}} \phi)$$

$$M, q \models_{\text{IATL}} \langle\langle C\rangle\rangle(\phi_1\mathcal{U}\phi_2) \Leftrightarrow \\ \exists f_C \in Str(M, C)\forall\lambda \in comp(M \dagger f_C, q, f_C) \\ \exists j \geq 0(M \dagger f_C, \lambda[j] \models_{\text{IATL}} \phi_2 \\ \text{and } \forall 0 \leq k < j(M \dagger f_C, \lambda[k] \models_{\text{IATL}} \phi_1))$$

Note, again, that the logic IATL is defined with memoryless strategies. We also define a version of the irrevocable strategies semantics for memory-based strategies, called MIATL (*memory-based* IATL). The language of the logic MIATL is the same as the language of IATL. Unlike in IATL, in MIATL we cannot update the model directly, but must first unfold the model into an (equivalent) tree-like structure, and then eliminate the branches which represent computations which do not conform to the strategy we update with. The tree-unfolding of a CGS M from a state q is denoted $T(M, q)$, see appendix B for a (standard) definition. Note that a memory-based strategy in M is equivalent to a memoryless strategy in $T(M, q)$. Thus, the MIATL semantics can be defined as follows:

$$M, q \models_{\text{MIATL}} \phi \Leftrightarrow T(M, q), q \models_{\text{IATL}} \phi$$

Intuitively, the MIATL meaning of the cooperation modalities involves pruning the model by a memory-based strategy.

The difference between the above interpretations and the original ATL interpretations is that in the former the subformula ϕ of a formula $\langle\langle C\rangle\rangle T\phi$ is evaluated in an updated model, where the actions of group C are fixed, while in the latter the subformula ϕ of $\langle\langle C\rangle\rangle T\phi$ is still evaluated in the original model. Consider model M_0 from Example 1 again. We have that $M_0, q_1 \models_{\text{ATL}} \langle\langle a\rangle\rangle\mathcal{G}\langle\langle a\rangle\rangle\mathcal{X}p$, but $M_0, q_1 \not\models_{\text{IATL}} \langle\langle a\rangle\rangle\mathcal{G}\langle\langle a\rangle\rangle\mathcal{X}p$, and $M_0, q_1 \not\models_{\text{MIATL}} \langle\langle a\rangle\rangle\mathcal{G}\langle\langle a\rangle\rangle\mathcal{X}p$. To that IATL and MIATL are different, it is enough to observe that $M_0, q_1 \models_{\text{MIATL}} \langle\langle a\rangle\rangle\mathcal{X}\langle\langle a\rangle\rangle\mathcal{X}p$, but $M_0, q_1 \not\models_{\text{IATL}} \langle\langle a\rangle\rangle\mathcal{X}\langle\langle a\rangle\rangle\mathcal{X}p$.

5 Non-Invariance of IATL/MIATL under Bisimulations

An important difference between the semantics of ATL and IATL/MIATL emerges when we consider bisimulations between pointed models. A definition of bismulation for CGSs, invariance results for ATL, and related definitions and results are presented in Appendix A[2].

Proposition 4 *Truth of formulae in* IATL *is not invariant under bisimilations.*

Proof: Let M, M' be the two CGSs shown in Figure 2. Note that (M, q_1) and (M', q_1') are bisimilar (take $q_1\beta q_1'$, $q_2\beta q_2'$, $q_3\beta q_3'$, $q_5\beta q_3'$ and $q_4\beta q_4'$), so ATL cannot discern between them.

However, observe that

$$M, q_1 \models_{\text{IATL}} \langle\langle 1\rangle\rangle\mathcal{X}((\langle\langle 2\rangle\rangle\mathcal{X}\mathbf{A}\mathcal{X}\neg p) \wedge \langle\langle 2\rangle\rangle\mathcal{X}\mathbf{A}\mathcal{X}p)$$

– the strategy for the first path quantifier (agent 1) is $q_3 \mapsto \alpha_1, q_5 \mapsto \alpha_2$; the strategy for the second path quantifier (agent 2) is $q_2 \mapsto \beta_1$; the strategy for the third path quantifier (agent 2) is $q_2 \mapsto \beta_2$. Similarly, we have that

$$M, q_1 \models_{\text{MIATL}} \langle\langle 1\rangle\rangle\mathcal{X}((\langle\langle 2\rangle\rangle\mathcal{X}\mathbf{A}\mathcal{X}\neg p) \wedge \langle\langle 2\rangle\rangle\mathcal{X}\mathbf{A}\mathcal{X}p)$$

(the memory-based strategies for the three quantifiers are $q_1q_2q_3 \mapsto \alpha_1, q_1q_2q_5 \mapsto \alpha_2$; $q_1q_2 \mapsto \beta_1$ and $q_1q_2 \mapsto \beta_2$, respectively).

[2] Although these are standard notions from modal logic adapted to CGSs and ATL, we are not aware that they have appeared in the literature before.

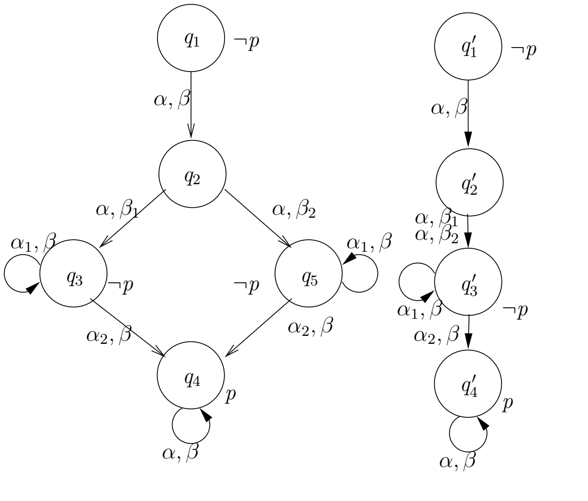

Figure 2: CGSs M (left) and M' (right), each with two agents.

In other words, the strategy witness for the first quantifier (agent 1) makes different choices in the two bisimilar states q_3 and q_5. This is "detectable" by IATL, because this strategy is still in "effect" when the second and third quantifier are evaluated.

However, we also have that

$$M', q_1' \not\models_{\text{IATL}} \langle\langle 1\rangle\rangle\mathcal{X}((\langle\langle 2\rangle\rangle\mathcal{X}\mathbf{A}\mathcal{X}\neg p) \wedge \langle\langle 2\rangle\rangle\mathcal{X}\mathbf{A}\mathcal{X}p)$$

and

$$M', q_1' \not\models_{\text{MIATL}} \langle\langle 1\rangle\rangle\mathcal{X}((\langle\langle 2\rangle\rangle\mathcal{X}\mathbf{A}\mathcal{X}\neg p) \wedge \langle\langle 2\rangle\rangle\mathcal{X}\mathbf{A}\mathcal{X}p).$$

□

Corollary 5 *Coalitional ability under irrevocable strategies is not expressible in* ATL.

6 Comparing ATL, IATL, and MIATL

In this section, ATL, IATL, and MIATL will also denote the sets of validities of the respective logics. Here we will present some basic facts relating these sets, let us but first point out that the semantics of IATL and MIATL coincide for a special class of models.

Definition 6 *A* CGS *is* tree-like *if there is a state (root) from which every state can be reached by a unique finite computation. A typical example of tree-like CGS is the* tree-unfolding $T(M, q)$ *of a given CGS M from a given state q in it (see the Appendix B for details).*

Since every state in a tree-like CGS has a unique history (the path from the root), memoryless and memory-based strategies coincide in tree-like CGSs, and therefore IATL and MIATL are equivalent in them. More precisely:

Proposition 7 *For every pointed tree-like CGS (M, q) and an* ATL*-formula ϕ:*

$$M, q \models_{\text{IATL}} \phi \textit{ iff } M, q \models_{\text{MIATL}} \phi.$$

Theorem 8

1. ATL $\not\subseteq$ IATL*, and* ATL $\not\subseteq$ MIATL*;*

2. IATL $\not\subseteq$ ATL*, and* MIATL $\not\subseteq$ ATL*;*

3. IATL $\subseteq$ MIATL*;*

4. MIATL $\not\subseteq$ IATL.

Proof:

1. It follows from the fact that IATL and MIATL are not closed under uniform substitution. Indeed, the ATL axiom $\neg\langle\langle\emptyset\rangle\rangle\mathcal{X}p \rightarrow \langle\langle\Sigma\rangle\rangle\mathcal{X}p$ is valid in each of IATL and MIATL, too, but the result of substitution of $\langle\langle\Sigma\rangle\rangle\mathcal{X}p \wedge \langle\langle\Sigma\rangle\rangle\mathcal{X}\neg p$ for p in it is no longer valid in either of them.

2. $\langle\langle C\rangle\rangle\mathcal{X}\langle\langle C\rangle\rangle\mathcal{X}\phi \leftrightarrow \langle\langle C\rangle\rangle\mathcal{X}\langle\langle\emptyset\rangle\rangle\mathcal{X}\phi$, for $C \neq \emptyset$, is a validity of both IATL and MIATL, but not of ATL.

3. If an ATL formula ϕ is IATL-valid, then it is IATL-valid in every tree-like CGS. Therefore, by Proposition 7, it is MIATL-valid in every every tree-like CGS. Hence, by Corollary 24, it is MIATL-valid in every CGS.

4. Let $\phi \equiv \neg p \wedge \langle\langle\emptyset\rangle\rangle\mathcal{G}(\neg p \rightarrow \langle\langle\emptyset\rangle\rangle\mathcal{X}p) \wedge \langle\langle\emptyset\rangle\rangle\mathcal{G}(p \rightarrow \langle\langle\Sigma\rangle\rangle\mathcal{X}p \wedge \langle\langle\Sigma\rangle\rangle\mathcal{X}\neg p)$. Let us divide the set of states of a given model into those that satisfy p, and those that do not. Formula ϕ says (roughly) that, for the first set, the system is free stay within it or to proceed to the other set; from the latter set, however, the system is bound to come back to the first set in one step. Moreover, the current state must belong to the second set. Now, formula $\Psi \equiv \phi \rightarrow \langle\langle\Sigma\rangle\rangle\mathcal{X}(\langle\langle\emptyset\rangle\rangle\mathcal{X}p \wedge \langle\langle\emptyset\rangle\rangle\mathcal{X}\langle\langle\emptyset\rangle\rangle\mathcal{X}\neg p)$ is MIATL-valid, with the following strategy that demonstrates it. First, for the history consisting only of the current state q, the agents execute any combination of actions, which leads to a state $q' \models p$. Then, for qq', they execute a combination of actions that leads to $q'' \models p$. Finally, for $qq'q''$, they execute actions that lead to $q''' \not\models p$.

   We now demonstrate that Ψ is not IATL-valid. Let M be the CGS depicted in Figure 3. We have that $M, q_2 \models_{\text{IATL}} \phi$, but there is no memoryless strategy that satisfies the right-hand side of Ψ in q_2.

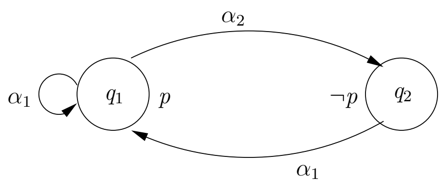

Figure 3: CGS with a single agent. Transitions are labeled with the possible choices of the agent.

□

These observations indicate that there is no straightforward way of reducing the decidability of satisfiability and complete axiomatization of the validities in IATL to those in ATL established in [4].

7 Normalizing Formulae of IATL

$\langle\langle\emptyset\rangle\rangle\mathcal{X}\phi$ expresses the fact that ϕ is inevitably true in the next state. We can say that a coalition C has *no choice* wrt. a formula ϕ if C can achieve ϕ in the next state precisely when ϕ is inevitable:

$$NoChoice(C, \phi) = \langle\langle C\rangle\rangle\mathcal{X}\phi \leftrightarrow \langle\langle\emptyset\rangle\rangle\mathcal{X}\phi.$$

Now, for any formula ϕ and coalitions C, C' we define:

$$Irrevocable(C, C', \phi) = [\![C]\!]\mathcal{G}NoChoice(C', \phi).$$

Proposition 9 *After a coalition C has committed to a strategy, all its sub-coalitions have no choice any more. Formally, for any formula ϕ and coalitions C, C' such that $C' \subseteq C$:*

$$\models_{\text{IATL}} Irrevocable(C, C', \phi).$$

The above can be generalized to:

$$\models_{\text{IATL}} [\![C]\!]\mathcal{G}(\langle\langle C'\rangle\rangle\mathcal{X}\phi \leftrightarrow \langle\langle C' \setminus C\rangle\rangle\mathcal{X}\phi)$$

for any coalitions C, C'.

We denote the last formula scheme by Normal. Given any formula ϕ, by repeated use of Normal, one can subsequently remove from the internal coalition C' in every subformula $\langle\langle C\rangle\rangle \ldots \langle\langle C'\rangle\rangle \ldots$ of ϕ all agents in $C \cap C'$, and thus eventually show the following.

Definition 10 *An* ATL*-formula ϕ is* normal *if it contains no nested occurrences of strategic operators with intersecting coalitions. Formally, ϕ is normal if it contains no subformula of the type $\langle\langle A_1\rangle\rangle \ldots \langle\langle A_2\rangle\rangle \ldots$ where $A_1 \cap A_2 \neq \emptyset$.*

Proposition 11 *Every formula is* IATL*-equivalent to an effectively computable normal formula.*

Consequently, testing IATL-satisfiability of any formula can be effectively reduced to testing IATL-satisfiability of a normal formula, which in turn can be established by suitably modifying the construction of alternating tree automata associated with such formulae, presented in [4].

The importance of normal formulae derives from the fact that their IATL-semantics is essentially compositional and they behave in many respects as in ATL. However, one additional effect of the irrevocable strategies is that once all agents commit to their strategies, only one computation remains possible, i.e. the system becomes *deterministic*, which admits an additional ATL-valid scheme. This yields a new scheme of IATL-valid formulae which are only ATL-valid in deterministic systems:

$$[\![\Sigma]\!]\mathcal{G}([\![\emptyset]\!]\mathcal{X}p \rightarrow \langle\langle\emptyset\rangle\rangle\mathcal{X}p).$$

Thus, within the scope of $\langle\langle\Sigma\rangle\rangle$ all cooperation modalities are completely trivialized and can be simply omitted; the result is an LTL formula, evaluated on the unique computation determined by the committed collective strategy of all agents.

8 Model Checking Irrevocable Strategies

In this section we consider the complexity of verification of IATL formulae through model checking. The (local) *model checking* problem asks whether a given formula ϕ holds in a given model M and state q. We prove that model checking is NP-hard and $\Delta_2^{\mathbf{P}}$-easy in the size of models, and length of formulae. As the gap between these two classes is not large (they both belong to the first level of the polynomial hierarchy), we conjecture that the problem is probably $\Delta_2^{\mathbf{P}}$-complete, and leave the definite answer for future work.

We begin with sketching algorithm $mcheck(M, q, \phi)$ that returns "yes" if $M, q \models_{\text{IATL}} \phi$ and "no" otherwise, running in nondeterministic polynomial time. Let $mctl(\phi, M)$ be a CTL model checker that returns the set of all states that satisfy ϕ in M.

- Cases $\phi \equiv p, \phi \equiv \neg\psi, \phi \equiv \psi_1 \wedge \psi_2$: straightforward (proceed as usually).
- Case $\phi \equiv \langle\langle A\rangle\rangle\mathcal{G}\psi$:
  1. Run $mcheck(\psi, M, q)$ for every $q \in Q$, and label the states in which the answer was "yes" with an additional proposition yes (not used elsewhere).
  2. Guess the best strategy of A, and "trim" model M by removing all the transitions

inconsistent with the strategy (yielding a sparser model M').

3. Return "yes" iff $Q \subseteq \mathit{mctl}(\mathsf{A}\mathcal{G}\mathsf{yes}, M')$.

- Cases $\phi \equiv \langle\langle A\rangle\rangle \mathcal{X}\psi$ and $\phi \equiv \langle\langle A\rangle\rangle \psi_1 \mathcal{U} \psi_2$: analogous.

Note that the size of a strategy is polynomial in the number of transitions, and $\mathit{mctl}(\phi, M)$ runs in polynomial time wrt. the size of M and ϕ. Thus, the algorithm runs in nondeterministic polynomial time for the simple formulae that include *only one* cooperation modality. For more complex formulae, it requires a polynomial number of calls to an oracle of range NP: namely, the oracle is *mcheck* itself, and the number of calls is the number of subformulae in ϕ. This gives us the following.

Proposition 12 *Model checking* IATL *is in* $\Delta_2^{\mathbf{P}}$ *wrt the size of the model and the formula.*

Now we can briefly sketch the NP-hardness proof, which follows by a reduction of the model checking problem for a subset of Schobbens's ATL$_{ir}$ [6], which is a variant of ATL for agents with imperfect information and imperfect recall. Models of ATL$_{ir}$ extend concurrent game structures with *indistinguishability relations* $\sim_a$ (equivalences), one per agent $a \in \Sigma$. Then, these relations are used to define agents' strategies that specify the same choice in indistinguishable states: s_A is *uniform* iff $q \sim_a q'$ implies $s_a(q) = s_a(q')$ for all $q, q' \in Q$, $a \in A$. Now:

$M, q \models \langle\langle A\rangle\rangle_{ir} \mathcal{X}\phi$ iff there is a uniform strategy s_A such that, for each $a \in A$, q' with $q \sim_a q'$, and $\lambda \in \mathit{out}(q', S_A)$, we have $M, \lambda[1] \models \phi$

and similarly for $\langle\langle A\rangle\rangle_{ir} \mathcal{G}\phi$, $\langle\langle A\rangle\rangle_{ir} \phi \mathcal{U} \psi$.

Regarding model checking complexity, the NP-hardness proof from [6] can be easily adapted to show the following:

Proposition 13 *Let* $\mathcal{M}_1$ *be the class of ATL$_{ir}$ models that include only one agent (i.e.,* $\Sigma = \{1\}$*), and let ATL$_{ir}^-$ be the sublanguage of ATL$_{ir}$ that includes only formulae* $\langle\langle 1\rangle\rangle_{ir} \mathcal{F} p$ *(where p is a proposition). Model checking ATL$_{ir}^-$ over* $\mathcal{M}_1$ *is NP-complete.*

Let the following be given: an ATL$_{ir}$ model $M \in \mathcal{M}_1$, a state q in M, and an ATL$_{ir}^-$ formula $\phi \equiv \langle\langle 1\rangle\rangle_{ir} \mathcal{F} p$. First, we construct a model M' in which the last action of the agent is "remembered" in the subsequent state, as in [3, Proposition 16]. Then, for each $\alpha \in \mathit{Act}$, we add a new proposition $\overline{\alpha}$ such that $\overline{\alpha}$ holds exactly in the states after α has been executed (i.e., states (q, α)). Finally, we add a new state q_0 with a sole outgoing transition to q, and no incoming transitions. Note that the number of states in M' is linear in the transitions of M, and the number of transitions in M' is at most quadratic in the transitions of M. Let us consider the following formula of IATL *extended with epistemic operators* K_a:[3]

$$\begin{aligned} \Psi &\equiv \langle\langle 1\rangle\rangle \mathcal{X} \langle\langle \emptyset\rangle\rangle (\mathit{uniform}\, \mathcal{U} p), \text{ where} \\ \mathit{uniform} &\equiv \bigvee_{\alpha \in \mathit{Act}} K_1 \langle\langle \emptyset\rangle\rangle \mathcal{X} \overline{\alpha}. \end{aligned}$$

The formula *uniform* characterizes uniformity of the presently executed strategy with respect to the current state (and states indistinguishable from it). Thus, formula Φ says that there is a strategy such that, if agent 1 commits to executing it (through $\langle\langle 1\rangle\rangle \mathcal{X}$), a sequence of uniform moves will follow that end up in a state satisfying p.[4] To get rid of the epistemic operators and relations, we use the satisfaction-preserving construction from [3, Section 4.4], that yields a CGS M'' and IATL formula Ψ'. Now, $M, q \models \phi$ iff $M'', q_0 \models \Psi''$, which concludes the reduction.

Proposition 14 *Model checking* IATL *is in NP-hard wrt the size of the model and the formula.*

9 Related Work

Counterfactual ATL *(*CATL*)* [7] extends ATL with "counterfactual commitment" operators $C_i(\sigma, \phi)$ where i is an agent, σ is a term symbol standing for a strategy, and ϕ is a formula. The models of ATL (CGSs) are extended to provide an interpretation of the new operators; in particular, a function $||\cdot||$ interpreting term symbols such that $||\sigma||$ is a strategy. The informal reading of $C_i(\sigma, \phi)$ is supposed to be "if it were the case that agent i commited to strategy σ, then ϕ would hold". Formally:

$$M, q \models_{\text{CATL}} C_i(\sigma, \phi) \Leftrightarrow M \dagger ||\sigma||_i, q \models_{\text{CATL}} \phi$$

where updates are defined similarly to in this paper.

It is clear that IATL does not subsume CATL, since formulae of the latter logic can refer directly to a strategy, and to the same strategy in difference places in a single formula (indeed, this expressive power is the main motivation behind CATL). On the other hand, CATL does not seem to subsume IATL either. Take the IATL formula $\langle\langle i\rangle\rangle \mathcal{X}\phi$. One attempt to write this property

[3] We recall the standard semantics: $M, q \models K_a \phi$ iff $M, q' \models \phi$ for every q' such that $q \sim_a q'$.

[4] Note that the epistemic operator K_a refers to incomplete information, but $\langle\langle 1\rangle\rangle$ refers now to arbitrary (i.e., not necessarily uniform) strategies.

in CATL could be $\bigvee_{\sigma_i \in \Upsilon_i} C_i(\sigma_i, \mathbf{A}\mathcal{X}\phi)$, where Υ_i is the set of term symbols that can be used to refer to strategies for agent i. These two formulae are equivalent if there is a one-to-one correspondence between the set $\{||\sigma_i|| : \sigma_i \in \Upsilon_i\}$ and the set of possible strategies for agent i. This is not necessarily the case however; there is no such restrictions on the models of CATL. The number of strategies, even memoryless, available to an agent at a given state depends on the CGS and there is no upper bound for that number, so the formula above can only work in a fixed CGS, where for every such strategy there is a term in the language. While CATL can quantify over strategies in another way, namely using the ATL connectives, there is no way to connect the witness of the existential strategy quantifier $\langle\langle i \rangle\rangle$ to a strategy term σ in an expression $C_i(\sigma, \phi)$. Although CATL has a mechanism of irrevocable commitment, it does not seem to model *strategic ability* under irrevocable commitment, like IATL does.

10 Conclusions

This paper is a preliminary report on a study of multi-agent logics with irrevocable strategies. Our main objectives here have been to raise and discuss the issue and to demonstrate that the range of semantics based on irrevocable strategies present a natural and meaningful alternative to the standard compositional semantics of ATL – an alternative that is both conceptually interesting and technically challenging. We have shown that the non-compositionality comes at a price, as many comfortable features of the compositional Tarski-style semantics cease to hold, and the intuition built on that semantics can often be deluded when dealing with irrevocable strategies. However, we claim that this price is worth paying, as it brings the semantics of multi-agent logics closer to the spirit of the traditional game-theoretical concept of strategy.

Acknowledgements

T. Ågotnes acknowledges the financial support of the Research Council of Norway through research grant 166525/V30. V. Goranko acknowledges the financial support of the National Research Foundation of South Africa through research grant GUN 2072436.

References

[1] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time Temporal Logic. *Journal of the ACM*, 49:672–713, 2002.

[2] E. A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, pages 995–1072. Elsevier Science Publishers, 1990.

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

[4] V. Goranko and G. van Drimmelen. Complete axiomatization and decidability of the Alternating-time Temporal Logic. *Theoretical Computer Science*, 353:93–117, 2006.

[5] M. Pauly. A modal logic for coalitional power in games. *Journal of Logic and Computation*, 12(1):149–166, 2002.

[6] P. Y. Schobbens. Alternating-time logic with imperfect recall. *Electronic Notes in Theoretical Computer Science*, 85(2), 2004.

[7] W. van der Hoek, W. Jamroga, and M. Wooldridge. A logic for strategic reasoning. In *Proceedings of AAMAS'05*, pages 157–164, 2005.

[8] W. van der Hoek and M. Pauly. Modal logic for games and information. In Johan van Benthem, Patrick Blackburn, and Frank Wolter, editors, *Handbook of Modal Logic*, pages 1077–1148. Elsevier, 2006.

[9] W. van der Hoek, M. Roberts, and M. Wooldridge. Social laws in alternating time: Effectiveness, feasibility and synthesis. *Synthese*, 2005.

A Bisimulations between CGSs

Here, we present some standard semantic notions and results from modal logic, adapted to ATL and CGSs. Consider a fixed language ATL(Π, Σ).

Definition 15

1. Given two CGSs $M_1 = (\Sigma, \Pi, Q_1, \pi_1, Act_1, d_1, \delta_1)$ and $M_2 = (\Sigma, \Pi, Q_2, \pi_2, Act_2, d_2, \delta_2)$, and a set of agents $C \subseteq \Sigma$, a relation $\beta \subseteq Q_1 \times Q_2$ is a (global) C-bisimulation between M_1 and M_2, *denoted $M_1 \rightleftarrows^C_\beta M_2$, iff for any $q_1 \in Q_1$ and $q_2 \in Q_2$, $q_1 \beta q_2$ implies that*

Local harmony $\pi_1(q_1) = \pi_2(q_2)$,

Forth *For any $\alpha_1(C) \in D_1(q_1, C)$, there exists $\alpha_2(C) \in D_2(q_2, C)$ such that for every $s_2 \in out(M_2, q_2, \alpha_2(C))$, there exists $s_1 \in out(M_1, q_1, \alpha_1(C))$ such that $s_1 \beta s_2$.*

Back *Likewise, for 1 and 2 swapped.*

2. If $M_1 \rightleftarrows^C_\beta M_2$ and $q_1 \beta q_2$, then we also say that β is a local C-bisimulation between (M_1, q_1) and (M_2, q_2), *denoted $(M_1, q_1) \rightleftarrows^C_\beta (M_2, q_2)$.*

3. If β is a C-bisimulation between M_1 and M_2 for every $C \subseteq \Sigma$, we call it a (full global) bisimulation between M_1 and M_2, *denoted $M_1 \rightleftarrows_\beta M_2$. Likewise, we define a full local bisimulation between (M_1, q_1) and (M_2, q_2), denoted $(M_1, q_1) \rightleftarrows_\beta (M_2, q_2)$.*

Given a pointed CGS (M, q), $C \subseteq \Sigma$, and a joint strategy f_C, we define the set $\mathsf{reach}(M, q, f_C)$ of all states in M which are last states of finite computations in $\mathit{fincomp}(M, q, f_C)$, i.e., $\mathsf{reach}(M, q, f_C) = \{l(\sigma) \mid \sigma \in \mathit{fincomp}(M, q, f_C)\}$. In particular, we write $\mathsf{reach}(M, q)$ for $\mathsf{reach}(M, q, \emptyset)$. Likewise, given a CGS M a finite computation σ in it, $C \subseteq \Sigma$, and a joint strategy f_C, we put $\mathsf{reach}(M, \sigma, f_C) := \mathsf{reach}(M, l(\sigma), f_C)$. Note that a CGS M can be naturally restricted to a CGS over $\mathsf{reach}(M, q)$, denoted by $\mathsf{Reach}(M, q)$.

Proposition 16 *For any pointed CGS (M, q), the identity relation in M restricted to $\mathsf{reach}(M, q)$ is a bisimulation between (M, q) and $(\mathsf{Reach}(M, q), q)$.*

Proof: Straightforward. □

Proposition 17 *Given pointed CGSs (M_1, q_1) and (M_2, q_2) and a relation $\beta \subseteq \mathsf{reach}(M_1, q_1) \times \mathsf{reach}(M_2, q_2)$:*

$$(M_1, q_1) \rightleftarrows_\beta (M_2, q_2) \text{ iff } (\mathsf{Reach}(M_1, q_1), q_1) \rightleftarrows_\beta (\mathsf{Reach}(M_2, q_2), q_2).$$

Proof: Follows from prop. 16 and the fact that a composition of bisimulations is a bisimulation. □

For a fixed $C \subseteq \Sigma$ we denote by $\text{ATL}(\Pi[C])$ the fragment of $\text{ATL}(\Pi, \Sigma)$ consisting of only those formulae generated by

$$\phi ::= \top \mid p \mid \neg\phi \mid \phi \wedge \phi \mid \langle\langle C \rangle\rangle \mathcal{X}\phi \mid \langle\langle C \rangle\rangle \mathcal{G}\phi \mid \langle\langle C \rangle\rangle \phi \mathcal{U} \phi.$$

Theorem 18

1. If $M_1 \rightleftarrows^C_\beta M_2$ and $q_1 \beta q_2$, then

$$M_1, q_1 \models_{\text{ATL}} \phi \text{ iff } M_2, q_2 \models_{\text{ATL}} \phi$$

for every formula $\phi \in \text{ATL}(\Pi[C])$.

2. If, furthermore, $Dom(\beta) = Q_1$ and $Rng(\beta) = Q_2$, then

$$M_1 \models_{\text{ATL}} \phi \text{ iff } M_2 \models_{\text{ATL}} \phi$$

for every formula $\phi \in \text{ATL}(\Pi[C])$.

Proof: Claim 1. The proof goes by a induction on formulae. The key step is to show that every strategy on $\mathsf{reach}(M_1, q_1)$ can be 'simulated' by a strategy on $\mathsf{reach}(M_2, q_2)$ and vice versa. We will prove this for memoryless strategies, but the proof can be easily adapted to the memory-based semantics.

For instance, consider the case of $\langle\langle C \rangle\rangle \mathcal{G}\phi$, assuming that the claim holds for ϕ. Suppose $M_1, q_1 \models_{\text{ATL}} \langle\langle C \rangle\rangle \mathcal{G}\phi$ and let f^1_C be a joint strategy for C in M_1 that ensures $M_1, s_1 \models_{\text{ATL}} \phi$ for every $s_1 \in \mathsf{reach}(M_1, q_1, f^1_C)$.

Note that, assuming the set of actions in M_1 is at most countable, the set $\mathsf{reach}(M_1, q_1, f^1_C)$ will be at most countable, too; then we fix an enumeration of it[5].

We will define a joint strategy f^2_C for C in M_2 that ensures $M_2, s \models_{\text{ATL}} \phi$ for every $s \in \mathsf{reach}(M_2, q_2, f^2_C)$, as follows.

First, we define, simultaneously by induction on n the following:

(i) a chain of sets

$$\mathsf{reach}_0(M_2, q_2) \subseteq \ldots \subseteq \mathsf{reach}_n(M_2, q_2) \subseteq \ldots \subseteq \mathsf{reach}(M_2, q_2),$$

(where $\mathsf{reach}_n(M_2, q_2)$ will be the set of states in M_2 reachable in at most n steps from q_2 if the agents in C follow the joint strategy f^2_C);

(ii) a chain of mappings

$$\zeta_0 \subseteq \ldots \subseteq \zeta_n \subseteq \ldots,$$

such that $\zeta_n : \mathsf{reach}_n(M_2, q_2) \to \mathsf{reach}(M_1, q_1, f^1_C)$, and

$\zeta_n(s) \beta s$ for every $s \in \mathsf{reach}_n(M_2, q_2)$;

(iii) and a chain of partial joint strategies for C in M_2:

$$f_0(C) \subseteq \ldots \subseteq f_n(C) \subseteq \ldots$$

where the domain of $f_n(C)$ is $\mathsf{reach}_{n-1}(M_2, q_2)$.

(We put $\mathsf{reach}_{-1}(M_2, q_2) := \emptyset$, to get the inductive construction going).

First, we put $\mathsf{reach}_0(M_2, q_2) := \{q_2\}$, $\zeta_0(q_2) = q_1$, and $f_0(C) := \emptyset$.

Now, assuming that $\mathsf{reach}_n(M_2, q_2), \zeta_n$, and $f_n(C)$ are defined accordingly, we define $\mathsf{reach}_{n+1}(M_2, q_2), \zeta_{n+1}$, and $f_{n+1}(C)$ as follows.

Let $S_n := \mathsf{reach}_n(M_2, q_2) \setminus \mathsf{reach}_{n-1}(M_2, q_2)$. For every $s \in S_n$, let $s' = \zeta_n(s)$. Since $s' \beta s$, wee can

[5] In general, the construction in the proof would require application of Axiom of Choice, though.

choose a joint action $\alpha(s, C)$ for C from s in M_2, corresponding to the joint action for C from s' in M_1 determined by f^1_C, so as to ensure that for every $t \in out(M_2, s, \alpha(s, C))$ there is a β-bisimilar state $\zeta(t) \in out(M_1, s', f^1_C)$. In case $t \in \mathsf{reach}_n(M_2, q_2)$ we put $\zeta(t) = \zeta_n(t)$, otherwise – the first suitable state in the enumeration of $\mathsf{reach}(M_1, q_1, f^1_C)$ (such state exists since $s'\beta s$). Note, that $\zeta(t)$ does not depend on s.

Now, we define

$$\begin{array}{l}\mathsf{reach}_{n+1}(M_2, q_2) := \\ \quad \mathsf{reach}_n(M_2, q_2) \cup \bigcup_{s \in S_n} out(M_2, s, \alpha(s, C))\end{array} ,$$

$$\zeta_{n+1} := \zeta_n \cup \zeta,$$

and

$$f_{n+1}(C) := f_n(C) \cup \bigcup_{s \in S_n} \alpha(s, C).$$

It is immediate from the construction that $\mathsf{reach}_{n+1}(M_2, q_2), \zeta_{n+1}$, and $f_{n+1}(C)$ satisfy the requirements set above. This completes the inductive definition.

Finally, we define

$$f^2(C) := \bigcup_{n \in \omega} f_n(C),$$

and extend it arbitrarily (but constructively) to all states in M_2 where it is not defined, to obtain the desired f^2_C.

It follows from the definition, and from the main inductive hypothesis of the proof (applied to ϕ), that

$$\mathsf{reach}(M_2, q_2, f^2_C) = \bigcup_{n \in \omega} \mathsf{reach}_n(M_2, q_2),$$

and that $M_2, s \models_{\text{ATL}} \phi$ for every $s \in \mathsf{reach}(M_2, q_2, f^2_C)$. Thus, $M_2, q_2 \models_{\text{ATL}} \langle\!\langle C \rangle\!\rangle \mathcal{G}\phi$.

The converse direction is completely symmetric.

The proof for the case $\langle\!\langle C \rangle\!\rangle \phi_1 \mathcal{U} \phi_2$ is similar and the details are left to the reader.

Claim 2 is an immediate consequence from Claim 1.

□

Corollary 19 *If $M_1 \rightleftarrows_\beta M_2$ and $s_1 \beta s_2$, then*

$$M_1, s_1 \models_{\text{ATL}} \phi \textit{ iff } M_2, s_2 \models_{\text{ATL}} \phi$$

for every formula $\phi \in \text{ATL}(\Pi, \Sigma)$.

If, furthermore, $Dom(\beta) = Q_1$ (the set of states of M_1) and $Rng(\beta) = Q_2$ (the set of states of M_2), then

$$M_1 \models_{\text{ATL}} \phi \textit{ iff } M_2 \models_{\text{ATL}} \phi$$

for every formula $\phi \in \text{ATL}(\Pi, \Sigma)$.

Corollary 20 *For any pointed CGS (M, q) and formula $\phi \in \text{ATL}(\Pi, \Sigma)$,*

$$M, q \models_{\text{ATL}} \phi \textit{ iff } \mathsf{Reach}(M, q), q \models_{\text{ATL}} \phi.$$

B Tree-unfoldings of CGS

Definition 21 (Tree-unfolding of a CGS) *Given a CGS*

$$M = (\Sigma, \Pi, Q, \pi, Act, d, \delta)$$

and $q \in Q$, the tree-unfolding $T(M, q)$ of M from q *is defined as follows:*

$$T(M, q) = (\Sigma, \Pi, fincomp(M, q), \pi^*, Act, d^*, \delta^*),$$

where:

$\pi^*(\sigma) = \pi(l(\sigma))$, $d^*_i(\sigma) = d_i(l(\sigma))$, and $\delta^*(\sigma, \alpha) = \sigma\delta(l(\sigma), \alpha)$, i.e. the outcome state in $T(M, q)$ from σ under the joint action α is the finite computation obtained from σ by appending the outcome state in M from the last state of σ under α.

It is immediate from the definition that every tree unfolding $T(M, q)$ is a tree-like CGS.

Since every state in a tree has a unique history (path to the root), we have the following.

Proposition 22 *For any pointed CGS (M, q), $(T(M, q), q) \rightleftarrows_L (M, q)$, where $L = \{(\sigma, l(\sigma)) \mid \sigma \in fincomp(M, q)\}$.*

Proof: First, $l(q) = q$, hence qLq. The 'Local harmony' condition $\pi^*(\sigma) = \pi(l(\sigma))$, is satisfied by definition. For the 'Forth' condition: let $\alpha(C) \in D(T(M, q), \sigma, C)$. Then $\alpha(C) \in D(M, l(\sigma), C)$, and for every every $s \in out(M, l(\sigma), \alpha(C))$, we have that $(\sigma \circ s)Ls$ and $\sigma \circ s \in out(T(M, q), \sigma, \alpha(C))$, where $\sigma \circ s$ is the computation obtained from σ by appending s to it. Finally, the 'Back' condition: let $\alpha(C) \in D(M, l(\sigma), C)$. Then $\alpha(C) \in D(T(M, q), \sigma, C)$ and for every $\sigma' \in out(T(M, q), \sigma, \alpha(C))$ we have that $\sigma' = \sigma\delta(l(\sigma), \alpha')$ for some $\alpha' \in D(T(M, q), \sigma)$ such that $\alpha'(C) = \alpha(C)$. Then $\delta(l(\sigma), \alpha') \in out(M, l(\sigma), \alpha(C)$ and $\sigma' L \delta(l(\sigma), \alpha')$. □

Proposition 23 *Let (M, q) be a pointed CGS and ϕ any* ATL*-formula. Then the following are equivalent.*

1. *$M, q \models_{\text{MIATL}} \phi$.*
2. *$T(M, q), q \models_{\text{MIATL}} \phi$.*
3. *$T(M, q), q \models_{\text{IATL}} \phi$.*

Corollary 24 *If a formula is* MIATL*-satisfiable, then it is* MIATL*-satisfiable in a tree-like CGS.*