TU Clausthal
Clausthal University of Technology

Strategic Commitment and Release in Logics for Multi-Agent Systems (Extended abstract)

Thomas Ågotnes[1], Valentin Goranko[2], and Wojciech Jamroga[3]

IfI Technical Report Series **IfI-08-01**

Department of Informatics
Clausthal University of Technology

Impressum

Publisher: Institut für Informatik, Technische Universität Clausthal
Julius-Albert Str. 4, 38678 Clausthal-Zellerfeld, Germany
Editor of the series: Jürgen Dix
Technical editor: Wojciech Jamroga
Contact: wjamroga@in.tu-clausthal.de
URL: http://www.in.tu-clausthal.de/forschung/technical-reports/
ISSN: 1860-8477

The IfI Review Board

Strategic Commitment and Release in Logics for Multi-Agent Systems (Extended abstract)

Thomas Ågotnes[1], Valentin Goranko[2], and Wojciech Jamroga[3]

[1] Department of Computer Engineering, Bergen University College, Norway
[2] School of Mathematics, University of the Witwatersrand, Johannesburg, South Africa
[3] Department of Informatics, Clausthal University of Technology, Germany
tag@hib.no, goranko@maths.wits.ac.za,
wjamroga@in.tu-clausthal.de

Abstract

We analyze the semantics of ATL* and related logics for strategic abilities. Our focus is on how these logics treat agents' commitments to strategies in the process of formula evaluation. We point out some questionable effects in that respect in the standard semantics, and discuss some alternatives leading to amendments of that semantics. We also propose and discuss various syntactic and semantics mechanisms for handling commitments to strategies and release from such commitments in the semantics of ATL*, leading to more expressive and semantically refined versions of that logic.

1 Introduction

Agents in multi-agent systems commit to strategies and relinquish their commitments in pursuit of their individual and collective goals in a dynamic, pragmatic, and often quite subtle way. While the semantics of some logics for multi-agent systems, such as ATL, is based on the notion of strategy, we claim that these logics do not yet capture adequately the strategic aspect of agents' behaviour. For instance, the meaning of an ATL formula $\langle\langle C\rangle\rangle\gamma$ is that the coalition C has a collective strategy to bring about the truth of γ. However, according to the formal semantics of ATL introduced in [3], the evaluation of γ in the possible future runs of the system *does not take into account that strategy anymore*. That is, the semantics of ATL *does not commit agents to strategies* they choose in order to bring about γ. The standard semantics of

ATL does not take into account the *strategy context*[1] when evaluating formulae. This apparent shortcoming of the original semantics of ATL has been independently addressed in different ways in several recent publications, including [1, 5, 8, 4, 10], where various proposals have been made in order to incorporate strategic commitment in the syntax and semantics of ATL.

In the present paper, which is a follow-up to [1], we analyze the problem further. We discuss some of the proposed solutions, and introduce some new mechanisms for managing strategic commitments and release in ATL. The main objective of this extended abstract is to raise and discuss the conceptual issues, rather than to present technical results, which will be included in a forthcoming full paper.

Before embarking on the actual discussion, we would like to make two preliminary remarks:

- We use the term *commitment* (to a strategy) in a specific technical sense. Thus, our notion of strategic commitment may differ somewhat from how it has been defined and used elsewhere. The issue is complex and subtle, allowing for different, possibly mutually exclusive interpretations, and we make no claim to have captured its full meaning. In fact, we believe that any formalization of that notion, including ours, is bound to be incomplete and imprecise.

- Likewise, we use the term *goal* (of coalition C) in a precise and technical sense, to indicate the subformula γ within formula $\langle\langle C\rangle\rangle\gamma$. In doing so, we ignore the issue of whether agents must have goals, how these goals arise etc. Again, this is a subject of a philosophical discussion which our paper is not intended to enter.

2 Preliminaries: alternating-time temporal logic

ATL [3] can be understood as a generalization of the branching time temporal logic CTL, in which path quantifiers are replaced with so called *cooperation modalities*. Formula $\langle\langle A\rangle\rangle\varphi$, where A is a coalition of agents, expresses that A have a collective strategy to enforce φ. ATL formulae include temporal operators: "$\mathcal{X}$" ("in the next state"), $\mathcal{G}$ ("always from now on"), $\mathcal{F}$ ("eventually"), and $\mathcal{U}$ ("until"). Here, we focus on the unrestricted version of the logic, usually called ATL*. Formulae of ATL* are defined by the following grammar:

$$\phi ::= p \mid \neg\phi \mid \phi \wedge \phi \mid \langle\langle A\rangle\rangle\gamma, \qquad \gamma ::= \phi \mid \neg\gamma \mid \gamma \wedge \gamma \mid \mathcal{X}\gamma \mid \gamma\mathcal{U}\gamma.$$

[1] We take this term from [4].

Additional temporal operators can be defined as: $\mathcal{F}\gamma \equiv \top\mathcal{U}\phi$, and $\mathcal{G}\phi \equiv \neg\mathcal{F}\neg\phi$.

The semantics of ATL* is defined over *concurrent game structures*, each including a set of agents $\mathbb{A}\mathrm{gt}$, states St, actions Act, and atomic propositions Π, plus a valuation $\pi : St \to \mathcal{P}(\Pi)$. Function $d : \mathbb{A}\mathrm{gt} \times St \to \mathcal{P}(Act)$ defines actions available to an agent in a state, and o is a transition function that assigns the outcome state $q' = o(q, \alpha_1, \ldots, \alpha_k)$ to state q and a tuple of actions $\langle \alpha_1, \ldots, \alpha_k \rangle$ that can be executed by $\mathbb{A}\mathrm{gt}$ in q. A (memory-based) *strategy* $s_a : St^+ \to Act$ is a conditional plan that specifies what $a \in \mathbb{A}\mathrm{gt}$ is going to do for every possible situation, as encoded by the current history of the transition process; strategies that only depend on the current state are called *memoryless* or *positional*. For the semantics of ATL positional strategies suffice, but for the full ATL* memory-based strategies are needed (in the sense that the interpretation of formulae is different if only positional strategies are used). A *collective strategy* S_A is a tuple of strategies, one per agent from $A \subseteq \mathbb{A}\mathrm{gt}$. A *path* λ in model M is an infinite sequence of states that can be effected by subsequent transitions. We use $\lambda[n]$ to denote the nth state in λ (note: we start counting from 0); $\lambda[i..j]$ denotes the subpath of λ between positions i and j. Function $out(q, S_A)$ returns the set of all paths that may result from agents A executing strategy S_A from state q onward. Then:

$M, q \models p$ iff $p \in \pi(q)$ (for $p \in \Pi$);

$M, q \models \neg\phi$ iff $M, q \not\models \phi$;

$M, q \models \phi_1 \wedge \phi_2$ iff $M, q \models \phi_1$ and $M, q \models \phi_2$;

$M, q \models \langle\langle A \rangle\rangle\gamma$ iff there is a collective (memory-based) strategy S_A such that, for every $\lambda \in out(q, S_A)$ we have $M, \lambda \models \gamma$;

$M, \lambda \models \phi$ iff $M, \lambda[0] \models \phi$;

$M, \lambda \models \neg\gamma$ iff $M, \lambda \not\models \gamma$;

$M, \lambda \models \gamma_1 \wedge \gamma_2$ iff $M, \lambda \models \gamma_1$ and $M, \lambda \models \gamma_2$;

$M, \lambda \models \mathcal{X}\gamma$ iff $M, \lambda[1..\infty] \models \gamma$;

$M, \lambda \models \gamma_1\mathcal{U}\gamma_2$ iff $M, \lambda[i..\infty] \models \gamma_2$ for some $i \geq 0$, and $\lambda[j..\infty] \models \gamma_1$ for all $0 \leq j \leq i$.

3 A critical analysis of the standard semantics of ATL*

An interesting feature of ATL is that strategies in the logic are not really *permanent*, in the sense that in the evaluation of the goal ϕ the agent i is no longer restricted by the strategy she has chosen [1]. That is, if ϕ includes a

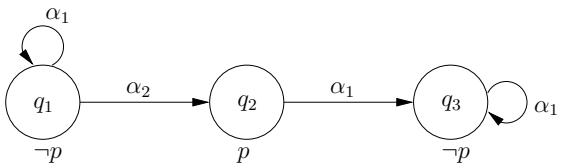

Figure 1: System M_0 with a single agent a. The transitions between states are labeled by the actions chosen by a.

nested cooperation modality for i, then i is again free to choose any strategy to demonstrate the truth of ϕ. This is 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" ($\mathbf{E}\mathcal{G}\mathbf{E}\phi$). However, it somehow contradicts the usual game-theoretical view of a strategy as a conditional plan that *completely* specifies the agent's future behavior.

3.1 Revocability of strategies in the semantics of ATL*

To illustrate that the meaning of ATL formulae can sometimes run contrary to intuition, let us consider an example from [1]:

Example 1 *We are given a system with a shared resource, and we 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. Intuitively, this is expressed in ATL by the 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.*

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 counter-intuitive 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.*

Control design and *algorithm design* are contexts where the original semantics of ATL clearly does not work well. Consider a system with agents/processes

$\mathbb{A}\mathrm{gt}$; the most natural model includes all the actions that can be in principle executed by the processes, plus appropriate state transformations as transitions. Now, e.g., $\langle\langle a\rangle\rangle\mathcal{G}\langle\langle b\rangle\rangle\mathcal{F}p$ has a natural reading as: "process a can (by choosing the right strategy) control the system so that property p is always achievable for b". Such a statement refers thus to a's control over things that can be executed by other processes. Of course, such control can be imposed only when the strategy chosen by a influences the evaluation of $\langle\langle b\rangle\rangle\mathcal{F}p$ (like in our "Irrevocable ATL" [1] and unlike in the original ATL).

Example 2 *Suppose that we deal with a computational system; then $\langle\langle a\rangle\rangle\ldots\langle\langle b\rangle\rangle\ldots\varphi$ can be as well understood as "there is an algorithm for process a so that b can/cannot bring about φ". For example, we may want the operating system to be designed in such a way that no process can crash the system, neither individually nor collectively. This can be specified as: $\langle\langle os\rangle\rangle\mathcal{G}\neg\langle\langle\mathbb{A}\mathrm{gt}\rangle\rangle\mathcal{F}\mathsf{crash}$. In the original semantics of ATL, the evaluation of $\neg\langle\langle\mathbb{A}\mathrm{gt}\rangle\rangle\mathcal{F}\mathsf{crash}$ is independent from the strategy that has been chosen for $\langle\langle os\rangle\rangle$, which suggests that the operating system has no influence whatsoever on states that the other processes can bring about.*

3.2 Strategic updates in the process of formula evaluation

In the standard semantics of ATL*, agents are completely free to change their strategies in the evaluation of subformulas, in order to achieve the respective new goals appearing there. Perhaps the best argument in favour of such choice is the compositionality of the semantics of ATL*. However, this compositionality comes at a questionable price. To illustrate the possible side effects of that semantics, let us consider one more example:

Example 3 *Consider the ATL-formula*

$$A = \langle\langle 1,2\rangle\rangle\mathcal{F}\langle\langle 2,3\rangle\rangle\mathcal{G}p,$$

where $1,2,3$ are agents and p is an atomic proposition.

Let M be any concurrent game model and q a state in M. Then, $M,q \models \langle\langle 1,2\rangle\rangle\mathcal{F}\langle\langle 2,3\rangle\rangle Gp$ means that agents $1,2$ have a strategy profile $\langle s_1,s_2\rangle$ so that on every computation λ in M compatible with that strategy profile there is a state $\lambda[n]$ such that $M,\lambda[n] \models \langle\langle 2,3\rangle\rangle\mathcal{G}p$. Note first, that the strategy profile $\langle s_1,s_2\rangle$ does not feature at all in the evaluation of $\langle\langle 2,3\rangle\rangle\mathcal{G}p$ at the state $\lambda[n]$ in M. Therefore, in order to justify the truth of $\mathcal{G}p$, the standard semantics of ATL* allows any strategy profile $\langle t_2,t_3\rangle$ for the agents $2,3$ to be selected, that would guarantee that truth on all computations compatible with it, even though the strategy t_2 may deviate from the earlier chosen strategy s_2 for agent 2; moreover, the strategy of agent 1 is no longer taken into account, either.

This phenomenon seems objectionable, not only conceptually, but and from technical viewpoint: it leads to rather questionable logical equivalences in ATL*, such as $\langle\langle A\rangle\rangle\langle\langle B\rangle\rangle p \equiv \langle\langle B\rangle\rangle p$ for any coalitions (intersecting or not) of agents A and B.

A partial remedy to the problem has been suggested already in [3], where so called Game Logic (GL) was introduced. In GL, the (existential) quantification over strategies and the (universal) quantification over the outcome computations are decoupled into separate strategy quantifiers and path quantifiers. Moreover, three types of formulae are introduced: state, path, and tree formulae. Then, a strategy context is introduced for the evaluation of a tree formula prefixed with a strategy quantifier:

> $M, q \models \exists A.\theta$ iff there is a strategy profile S_A for the coalition A, such that $M, comp(M, q, S_A) \models \theta$,

where $comp(M, q, S_A)$ is the tree of all computations compatible with S_A. However, because the strategy context is not preserved when passing to evaluation of state subformulae, a similar effect occurs in GL, too, e.g., $\exists A.\exists B.p \equiv \exists B.p$ for any coalitions A and B.

So, what are the reasonable alternatives to the standard semantics of ATL* that address the issue discussed above? Let us analyze them using again the formula $\langle\langle 1, 2\rangle\rangle\mathcal{F}\langle\langle 2, 3\rangle\rangle\mathcal{G}p$.

1. At the point of evaluation of $\langle\langle 1, 2\rangle\rangle\mathcal{F}\langle\langle 2, 3\rangle\rangle\mathcal{G}p$ the strategies of agents 1 and 2 are selected and fixed, so when evaluating the subformula $\langle\langle 2, 3\rangle\rangle\mathcal{G}p$ only the strategy of agent 3 can vary.

   This is the alternative adopted in [1], where commitment to strategies is considered *irrevocable*. This choice leads to different semantics, depending on whether memory-based or positional strategies are considered; both versions have been discussed in that paper. We write $M, q \models_{\text{IATL}} \phi$ and $M, q \models_{\text{MIATL}} \phi$ for satisfaction under positional (memoryless) and memory-based irrevocable strategies, respectively. The difference between the irrevocable and the standard semantics is illustrated in the model of Example 1, where we have: $M_0, q_1 \models_{\text{ATL}} \langle\langle a\rangle\rangle\mathcal{G}\langle\langle a\rangle\rangle\mathcal{X}p$, 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 $M_0, q_1 \not\models_{\text{MIATL}} \langle\langle a\rangle\rangle\mathcal{G}\langle\langle a\rangle\rangle\mathcal{X}p$.

   Shortcomings of this choice:

   - First, it imposes unnecessarily strong commitments when the goals are local (next-state goals) or eventualities – there is no rational reason why an agent should remain committed to a strategy after the goal has been achieved. As an example, consider the model in Figure 2. We have that:

$$\begin{array}{lll} M_1, q_1 & \models_{\text{IATL}} & \langle\langle 1\rangle\rangle\mathcal{X}((\langle\langle 2\rangle\rangle\mathcal{X}\mathbf{A}\mathcal{X}\neg p) \wedge \langle\langle 2\rangle\rangle\mathcal{X}\mathbf{A}\mathcal{X}p), \text{ and} \\ M_1, q_1 & \models_{\text{MIATL}} & \langle\langle 1\rangle\rangle\mathcal{X}((\langle\langle 2\rangle\rangle\mathcal{X}\mathbf{A}\mathcal{X}\neg p) \wedge \langle\langle 2\rangle\rangle\mathcal{X}\mathbf{A}\mathcal{X}p). \end{array}$$

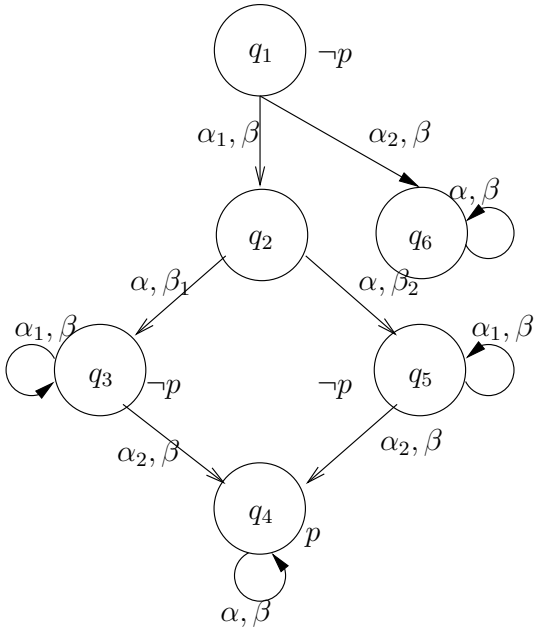

Figure 2: System M_1 with two agents a and b. The transitions between states are labeled by the actions chosen by a (left) and b (right).

In the first case the strategy for the first path quantifier (agent 1) is $q_1 \mapsto \alpha_1, q_3 \mapsto \alpha_1, q_5 \mapsto \alpha_2$; 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 $q_2 \mapsto \beta_2$. The formula says that in state q_1, agent 1 has the power to ensure that in the next state, agent 2 can alone choose whether in the next state after that p will be true in all following states, or false. But looking at the model, this property might seem counterintuitive. The irrevocable semantics requires agent 1 to choose a strategy *for any future circumstance*, even if the formula after $\langle\langle 1 \rangle\rangle$ refers only to what happens at the *next* moment. Consequently, that strategy will be in effect forever. So, in evaluating $\langle\langle 1 \rangle\rangle \mathcal{X}((\langle\langle 2 \rangle\rangle \mathcal{X} \mathbf{A} \mathcal{X} \neg p) \wedge \langle\langle 2 \rangle\rangle \mathcal{X} \mathbf{A} \mathcal{X} p)$ in q_1, we must come up with a witness strategy for agent 1 which not only chooses α_1 in q_1 (ensuring that the next state is q_2 and not q_6) but which also must make a choice in q_3 and q_5. In the irrevocable semantics that strategy is still in effect when evaluating $(\langle\langle 2 \rangle\rangle \mathcal{X} \mathbf{A} \mathcal{X} \neg p) \wedge \langle\langle 2 \rangle\rangle \mathcal{X} \mathbf{A} \mathcal{X} p$ in q_2. In the present paper we address this issue by considering mechanisms for *release of commitments*.

- Second, it leads to a non-compositional semantics – the truth of

a formula is now dependent not only on the model and state, but also on the strategy context, i.e. the currently committed strategies. We argue that the sacrifice of compositionality is inevitable here.[2]

2. At the point of evaluation of $\langle\langle 1,2\rangle\rangle\mathcal{F}\langle\langle 2,3\rangle\rangle\mathcal{G}p$ the strategies of agents 1 and 2 are selected and fixed, but when evaluating the subformula $\langle\langle 2,3\rangle\rangle\mathcal{G}p$ the agent 2 is granted the freedom to *change her strategy in order to achieve the current goal*, viz. $\mathcal{G}p$. Thus, both agents 2 and 3 can choose new strategies, and moreover they can do that under the assumption that agent 1 remains committed to his strategy selected at the point of evaluation of $\langle\langle 1,2\rangle\rangle\mathcal{F}\langle\langle 2,3\rangle\rangle\mathcal{G}p$.

   This is the alternative chosen in [4]. As demonstrated there, it leads to a quite expressive language (apparently more expressive than the previous alternative), especially when combined with the mechanism for "forgetting" strategies that have been selected.

   On the other hand, the resulting semantics is still non-compositional. Moreover, it raises the conceptual objection of not being faithful to the very concept of "strategy" which is usually understood as a *complete* specification of an agent's behaviour.

It would be instructive to compare the situation with the mechanism for quantification and variable binding in first-order logic. When evaluating a first order formula $\exists x A(x)$ in a given structure, an appropriate value for x is chosen and *fixed throughout the evaluation of the subformula* $A(x)$. On the other hand, when evaluating e.g., $\exists x(A(x) \wedge \exists x B(x))$, the value of x assigned at the point of evaluation of the whole formula has no bearing anymore when the subformula $\exists x B(x)$ is evaluated, and to justify its truth *any other value of* x *can be selected*. Thus, the semantics of first-order logic is in line with the second alternative discussed above. However, there is a subtle but important difference here: in first-order logic quantifiers may happen to use the same label (bound variable), but that does not mean that they refer to the same element of the domain. On the other hand, different occurrences of $\langle\langle a\rangle\rangle$ seem to refer to *the same* object: namely, the strategy that is effectively going to be executed by the agent.

Example 4 *Consider the design of the operating system from Example 2. Obviously, in such a context, the* os*'s strategy behind* $\langle\langle \mathbb{A}\mathrm{gt}\rangle\rangle$ *should be bound by the one already assigned to* $\langle\langle os\rangle\rangle$*, and not the other way around. The designer of the operating system looks for an algorithm (strategy) that prevents the system from crashing;*

[2]This does not mean that there is no compositional semantics. But we suspect that it would be rather technical and unintuitive, as e.g. in the case of Hodges' semantics for the Independence Friendly logic.

if the operating system itself changes its algorithm at runtime, the new algorithm may not guarantee this property any more.

3.3 An amended semantics for ATL* and the Game Logic with strategy contexts

In both alternatives to the original semantics of ATL*, considered above, a strategy context is being accumulated and taken into account throughout the process of formula evaluation; the difference is in the order of committing and overruling strategies: *from outside inwards* in alternative 1, and *from inside outwards* in alternative 2. That difference, however, affects the semantics in both alternatives considerably. Regardless of the specific choice of that order, we insist that it is reasonable to take strategy contexts into account in the evaluation of *every formula*.

Such strategy context can be effected technically by applying a (temporary) *model update*, as in [1], that fixes the actions available to the agents with strategies listed in the context, in accordance with these strategies. Thus, every state formula of ATL* is evaluated in the amended semantics at a triple (model, state, strategy context); respectively, every path formula of ATL* is evaluated at a triple (model, computation, strategy context), where the computation must be compatible with the current strategy context. Like [4], we will use collective strategies to represent current commitments of agents. For a collective strategy S_A, we define $S_A|B$ as the part of strategy S_A that refers to the agents from B as well. Then, strategy S_A is consistent with strategy R_B iff their common part agree, i.e. $S_A|B = R_B|A$. Like [4], we also define the update of commitment, but unlike [4] we assume that commitments persist: $S_A \circ R_B$ is a strategy T for agents $A \cup B$, such that $T|A = S_A$, and $T|(B \setminus A) = R|(B \setminus A)$. Now, we can re-write the semantics of "Irrevocable ATL" from [1] as follows:

$M, q, S \models p$ iff $p \in \pi(q)$;

$M, q, S \models \neg\phi$ iff $M, q, S \not\models \phi$;

$M, q, S \models \phi_1 \wedge \phi_2$ iff $M, q, S \models \phi_1$ and $M, q, S \models \phi_2$;

$M, q, S \models \langle\langle A \rangle\rangle \gamma$ iff there is a collective memoryless strategy F_A consistent with S, such that for every $\lambda \in out(q, S \circ F_A)$ we have $M, \lambda, S \circ F_A \models \gamma$;

$M, \lambda, S \models \phi$ iff $M, \lambda[0], S \models \phi$;

$M, \lambda, S \models \neg\gamma$ iff $M, \lambda, S \not\models \gamma$;

$M, \lambda, S \models \gamma_1 \wedge \gamma_2$ iff $M, \lambda, S \models \gamma_1$ and $M, \lambda, S \models \gamma_2$;

$M, \lambda, S \models \mathcal{X}\gamma$ iff $M, \lambda[1..\infty], S \models \gamma$;

$M, \lambda, S \models \gamma_1 \mathcal{U} \gamma_2$ iff $M, \lambda[i..\infty], S \models \gamma_2$ for some $i \geq 0$, and $\lambda[j..\infty], S \models \gamma_1$ for all $0 \leq j \leq i$.

One advantage of having strategy contexts is that it is easy now to define commitments that can be released at some moment in the future, since the "hard" structure of the model does not change during the evaluation of a formula (see Section 4.2 for some proposals). Another advantage is that defining the "full memory" variant of IATL is now straightforward: it is enough to replace "memoryless" with "full memory" in the semantic clause for $\langle\langle A\rangle\rangle\gamma$, and we get semantics for what we call MIATL* in [1].

Note that a similar update of the semantics can be considered for Game Logic. For instance, the amended GL semantics of $\exists C$ becomes:

> $M, q, S \models \exists A.\theta$ iff there is a collective strategy F_A, consistent with S, such that $M, q, S \circ F_A \models \theta$.

4 Mechanisms for strategy commitment and release in ATL*

Here, following our choice in [1], we adopt alternative 1 from Section 3.2. However, as mentioned earlier, we concede that the semantics of irrevocable strategies introduced in [1] is too rigid and unnecessarily restrictive in handling the strategic choices of the agents. We therefore consider here mechanisms for agents' dynamic commitments to strategies and releases from such strategies in the process of evaluation of formulae.

We distinguish two types of such mechanisms: *semantic and syntactic*. The former mechanisms are built into the semantics and there are no syntactic means to interfere with them explicitly, while the syntactic mechanisms leave the control on the update within the formula itself.

4.1 Semantic mechanisms for strategy update

There are various options, with different levels of sophistication here. Again, the two extremities are:

- the standard ATL*: complete freedom of update at every evaluation of a (sub)formula $\langle\langle C\rangle\rangle$;
- ATL* with irrevocable strategies, as in [1]: once a strategy for an agent is selected in the process of evaluation of a formula, it remains fixed throughout that evaluation.

Here is a sample running story:

> Three researchers, Alex, Bob, and Chris, work in the same scientific area. Alex and Bob decide to write a paper together. We will refer to it as Paper. They commit themselves to not start working on any other paper,

> alone, together, or with anyone else until they complete and submit Paper. Thus, the commitment is only *temporary*: until the goal is achieved. Once that is done, each one of them is free to release their commitment and start new research endeavors.

The story suggests a more refined option: to modify the semantics with irrevocable strategies in a *goal-driven* style, by letting agents commit to strategies only for as long as necessary to achieve their goals; e.g., for $\langle\langle C\rangle\rangle\varphi\mathcal{U}\psi$ the goal-driven semantics commits the agents in C to a strategy *only until the goal* ψ *is achieved*, and then automatically releases them from the commitment.

Back to the running story:

> For various reasons, the goal of Alex and Bob may never be achieved. That may become obvious gradually or suddenly, say, if Alex or Bob gets a very lucrative offer from the non-research world and decides to quit research, or develops incapacitating mental illness, or simply dies. Now, in that situation, it would make no sense for the other researcher to still faithfully adhere to his/her commitment, as it is certain that it will not bring about realization of the original goal.

Thus, it makes good sense to make provision for "escape clauses" and to refine the "goal-driven semantics" to a *pragmatic* one, where *the commitment of the agent only last until either the goal is achieved, or it becomes unachievable ever in the future*. In that case an automatic release from a commitment would be justified, and probably necessary in order to achieve other goals. This issue closely resembles the discussion on intention adoption and revision within the BDI community, which started with [6] and continues to this day [9].

However, we observe that defining such a semantics is not as straightforward as it seems. Suppose that we want to assume that, in the evaluation of $\langle\langle C\rangle\rangle\mathcal{F}\phi$, C are committed to their strategy only until ϕ is achieved, and then they are automatically released from their commitment. The point is, ϕ is the *whole* of what comes after $\langle\langle C\rangle\rangle\mathcal{F}$, so there is no further subformula which will be evaluated without the commitment. Thus, the evaluation of $\langle\langle C\rangle\rangle\mathcal{F}\phi$ yields the same result with and without such automatic release of commitments! Note that, in most cases, the shape of the actual specification would be $\langle\langle C\rangle\rangle\mathcal{F}(\phi\wedge\langle\langle C\rangle\rangle\psi)$, where ϕ specifies the "current" goal, and ψ a possible future goal. In this case, one may expect that the agents in C are committed to their first strategy as long as they pursue $\mathcal{F}\phi$ (or, until it becomes impossible to obtain), and then they are free again to choose their best policy towards ψ. It seems, however, that there are too many different cases to allow for a uniform semantic solution. We believe that syntactic means (discussed further in Section 4.2) prove more flexible: one must specify explicitly what the "current goal" is, and impose release as soon as it is achieved.

One viable semantic approach is to only consider *least commitment strategies*. Recall that a problem with the irrevocable semantics, which we illustrated in Section 3.2 (using Figure 2), is that an agent is required to commit

to actions in all future circumstances, also those which are not relevant for achieving the goal states. The reason for this is that strategies in ATL are *total* functions. A perhaps natural variant of the irrevocable semantics is to, instead, allow also *partial* functions as strategies ("partial strategies"). Partial strategies are also used in [2] where agents with bounded memory are considered (albeit not in the context of irrevocable strategies). The simplest case is the next-state operator. Changing the semantics in this case to only consider least commitment strategies correspond to only considering partial functions prescribing an action in the current state. Thus, in the model in Figure 2, we would have that

$$M_1, q_1 \not\models \langle\langle 1 \rangle\rangle \mathcal{X}((\langle\langle 2 \rangle\rangle \mathcal{X} \mathbf{A} \mathcal{X} \neg p) \wedge \langle\langle 2 \rangle\rangle \mathcal{X} \mathbf{A} \mathcal{X} p)$$

(for both memoryless and memory-bound variants). But would this not be throwing the baby out with the bath water; are we not back to the standard ATL* semantics if we take this idea seriously? We argue that we are not. Consider again the model in Figure 1. Recall that we have $M_0, q_1 \models_{ATL^*} \langle\langle a \rangle\rangle \mathcal{G} \langle\langle a \rangle\rangle \mathcal{X} p$, but $M_0, q_1 \not\models_{IATL} \langle\langle a \rangle\rangle \mathcal{G} \langle\langle a \rangle\rangle \mathcal{X} p$. If we only consider least commitment strategies (together with the irrevocable semantic interpretation), we would *still* have $M_0, q_1 \not\models \langle\langle a \rangle\rangle \mathcal{G} \langle\langle a \rangle\rangle \mathcal{X} p$.

As a further example of the subtleties of commitments, let us consider our running story dynamically.

> During the period of commitment, i.e. while Alex and Bob are working on the Paper, each of them may undertake other commitments, for as long as they are consistent with the one above.[3] Suppose, for instance, that at some stage Chris approaches Bob with the proposal to start working on a paper together. Bob cannot accept the proposal at that moment, but can say : 'OK, I have promised Alex not to start working on anything else until we submit our paper with her. But, *I can promise to you now to start working with you as soon as we are done with Alex, and not to undertake any similar promise with anyone else*.
> This is another commitment, and one that refers to the future.

4.2 Syntactic mechanisms for strategy update

Many of the semantic subtleties and complications suggested above can be controlled within the object language, in the formulae themselves, and this is the underlying idea of the syntactic approach. Again, there is a variety of meaningful proposals for enrichment of the language of ATL* with means for explicit control over the strategic commitments and releases. Here we briefly discuss two such proposals:

[3] It is a challenge, however, to define precisely the notion of "consistency" of commitments.

1. Add strategic operators $\langle\langle C/\theta\rangle\rangle\psi$, meaning:

   Coalition C has a strategy such that, if committed to it until θ becomes true, and then relinquished, will guarantee that ψ is true.

   Example: $\langle\langle A/\theta\rangle\rangle G\neg\langle\langle B\rangle\rangle\mathcal{F}\psi$ means: "*Coalition A has a strategy such that coalition B will never have a strategy to achieve ψ, even after θ becomes true and A's commitment is released*".

   Remark: we acknowledge a problem here: if the goal θ involves future operators, its achievement is subject to the strategy context, and the release of a coalition from its strategic commitment may hinder the success of the goal.

2. Another solution is to add to the language of ATL* explicit operators:

   commit (the agents in C to their strategies): $\langle\langle\downarrow C\rangle\rangle$, and

   release (the agents in C from their strategy commitment): $\langle\langle\uparrow C\rangle\rangle$

   Example: $\langle\langle\downarrow A\rangle\rangle G((\phi\vee\langle\langle\emptyset\rangle\rangle G\neg\phi)\rightarrow\langle\langle\uparrow A\rangle\rangle G\langle\langle B\rangle\rangle\mathcal{F}\psi)$ means: '*the coalition A has a strategy such that, if committing to it until the goal ϕ is achieved or becomes untenable and then relinquishing that strategy, the coalition B will always have a strategy to achieve the goal ψ*'.

   We note that the release operator $\langle\langle\uparrow C\rangle\rangle$ is very similar to the 'forget' operator, independently proposed in [4].

5 Expressiveness, satisfiability and model checking of ATL* with strategy commitment and release

The introduction of either semantic or syntactic mechanisms for strategy commitment and release generally increases the expressiveness of ATL and ATL*, as demonstrated in [1] and [4]. The precise analysis and comparison of the expressiveness for the variations introduced here is still under investigation. However, at least in the case of ATL with irrevocable strategies introduced in [1], as well as its refinement with semantic mechanism for strategic update discussed above, satisfiability remains decidable (in EXPTIME), by suitable adaptation of the incremental tableau-based decision procedure for ATL recently developed in [7]. However, at present it is still unknown if the syntactic mechanisms for strategic update proposed here do not lead to undecidable satisfiability.

As for model checking, the results from [4] suggest that the problem is PSPACE-complete for ATL with irrevocable memoryless strategies (with and

without release of commitments). The complexity of model checking irrevocable memory-based strategies is still an open question.

6 Concluding remarks

The issues discussed here, viz., how agents commit to strategies and relinquish such commitments in a dynamic multi-agent environment, are conceptually deep and subtle. We do not purport to provide a philosophical solution to them, but rather discuss mechanisms for their formal handling and the formal consequences for the logical semantics ensuing from the adoption of such mechanisms. This is just a beginning of a systematic treatment of that topic; here we have only scratched the surface of the dynamics of strategic commitments and releases of agents – and it is not known yet what exactly lies beneath that surface, and even less how to best formalize it.

References

[1] T. Ågotnes, V. Goranko, and W. Jamroga. Alternating-time temporal logics with irrevocable strategies. In D. Samet, editor, *Proceedings of the 11th International Conference on Theoretical Aspects of Rationality and Knowledge (TARK XI)*, pages 15–24, Univ. Saint-Louis, Brussels, 2007. Presses Universitaires de Louvain.

[2] Thomas Ågotnes and Dirk Walther. Towards a logic of strategic ability under bounded memory. In Thomas Ågotnes, Natasha Alechina, and Brian Logan, editors, *Proceedings of the MALLOW'007 Workshop on Logics for Resource-bounded Agents*, pages 9–24, Durham, UK, September 2007.

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

[4] Thomas Brihaye, Arnaud Da Costa, François Laroussinie, and Nicolas Markey. ATL with strategy contexts. in preparation, 2008.

[5] Krishnendu Chatterjee, Thomas A. Henzinger, and Nir Piterman. Strategy logic. In Luís Caires and Vasco Thudichum Vasconcelos, editors, *CONCUR*, volume 4703 of *Lecture Notes in Computer Science*, pages 59–73. Springer, 2007.

[6] P.R. Cohen and H.J. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42:213–261, 1990.

[7] Valentin Goranko and Dmitry Shkatov. Tableau-based decision procedures
for logics of strategic ability in multi-agent systems. submitted, 2008.

[8] Sophie Pinchinat. A generic constructive solution for concurrent games with expressive constraints on strategies. In Kedar S. Namjoshi, Tomohiro Yoneda, Teruo Higashino, and Yoshio Okamura, editors, *5th International Symposium on Automated Technology for Verification and Analysis*, volume 4762 of *Lecture Notes in Computer Science*, pages 253–267. Springer-Verlag, 2007.

[9] W. van der Hoek, W. Jamroga, and M. Wooldridge. Towards a theory of intention revision. *Synthese*, 155(2):265–290, 2007.

[10] D. Walther, W. van der Hoek, and M. Wooldridge. Alternating-time temporal logic with explicit strategies. In D. Samet, editor, *Proceedings of the 11th International Conference on Theoretical Aspects of Rationality and Knowledge (TARK XI)*, pages 269–278, Univ. Saint-Louis, Brussels, 2007. Presses Universitaires de Louvain.