Fundamenta Informaticae 62 (2004) 1–35

IOS Press

Agents that Know How to Play

Wojciech Jamroga

Parlevink Group, University of Twente, Netherlands

Institute of Mathematics, University of Gdansk, Poland

jamroga@cs.utwente.nl

Wiebe van der Hoek

Department of Computer Science, University of Liverpool, UK

wiebe@csc.liv.ac.uk

Abstract. We look at ways to enrich Alternating-time Temporal Logic (ATL) – a logic for specification and verification of multi-agent systems – with a notion of knowledge. Starting point of our study is a recent proposal for a system called Alternating-time Temporal Epistemic Logic (ATEL). We show that, assuming that agents act under uncertainty in some states of the system, the notion of allowable strategy should be defined with some caution. Moreover, we demonstrate a subtle difference between an agent knowing that he has a suitable strategy and knowing the strategy itself. We also point out that the agents should be assumed similar epistemic capabilities in the semantics of both strategic and epistemic operators.

Trying to implement these ideas, we propose two different modifications of ATEL. The first one, dubbed Alternating-time Temporal Observational Logic (ATOL), is a logic for agents with bounded recall of the past. With the second, ATEL-R*, we present a framework to reason about both perfect and imperfect recall, in which we also incorporate operators for reasoning about the past. We identify some feasible subsystems of this expressive system.

Keywords: multiagent systems, temporal logic, epistemic logic, knowledge, transition systems, games with incomplete information.

Address for corespondence: Parlevink Group, University of Twente, PO Box 217, NL-7500 AE Enschede, Netherlands

1. Introduction: What Coalitions can Achieve

Two logical systems based on temporal logic have been proposed recently to tackle specification and verification of multi-agent systems properties.[1] First, Alur, Henzinger and Kupferman posed their Alternating-time Temporal Logic (ATL and ATL* [2, 3, 4]), which offered a possibility of expressing the capabilities of autonomous agents in a way similar to the branching time temporal logics CTL and CTL* [9, 21]. The second language, Alternating-time Temporal Epistemic Logic (ATEL/ATEL*) proposed by van der Hoek and Wooldridge [17], enriches the picture with an epistemic component.

ATL* is an extension of CTL* in which a class of *cooperation modalities* $\langle\!\langle A \rangle\!\rangle$ ($A \subseteq \Sigma$, where Σ is the set of all *agents* or *players*) replace the simple path quantifiers E (*there is a path*) and A (*for all paths*).[2] The common-sense reading of $\langle\!\langle A \rangle\!\rangle \Phi$ is: "the group of agents A have a collective strategy to enforce Φ regardless of what all the other agents do". The original CTL operators E and A can be expressed in ATL with $\langle\!\langle \Sigma \rangle\!\rangle$ and $\langle\!\langle \emptyset \rangle\!\rangle$ respectively, but between both extremes we can express much more about the abilities of particular agents and groups of agents. The dual operator $[\![A]\!]$ can be defined in the usual way as $[\![A]\!]\Phi \equiv \neg\langle\!\langle A \rangle\!\rangle\neg\Phi$, meaning that A cannot avoid Φ on their own. ATL* inherits all the temporal operators from CTL*: $\bigcirc$ (*nexttime*), $\Diamond$ (*sometime*), $\Box$ (*always*) and $\mathcal{U}$ (*until*).

ATL extends CTL in the same way as ATL* extends CTL*: in ATL, every temporal modality is preceded by exactly one temporal operator. So, typical ATL formulas are $\langle\!\langle A \rangle\!\rangle\Diamond\varphi$, $\langle\!\langle A \rangle\!\rangle\Box\varphi$ and $\langle\!\langle A \rangle\!\rangle\varphi\mathcal{U}\psi$, where φ and ψ are ATL formulas.[3] Since model-checking for ATL* requires 2EXPTIME, but it is linear for ATL, ATL is more interesting for practical applications [4].

Examples of interesting properties that can be expressed with ATL include:

1. $\langle\!\langle A \rangle\!\rangle\Diamond\varphi$
2. $\langle\!\langle A \rangle\!\rangle\Box\varphi$
3. $\neg\langle\!\langle A \rangle\!\rangle\bigcirc\varphi \wedge \neg\langle\!\langle B \rangle\!\rangle\bigcirc\varphi \wedge \langle\!\langle A \cup B \rangle\!\rangle\bigcirc\varphi$
4. $\langle\!\langle A \cup \{a\} \rangle\!\rangle\bigcirc\varphi \rightarrow \langle\!\langle \{a\} \rangle\!\rangle\bigcirc\varphi$

The first of these expresses a kind of *cooperative liveness* property: coalition A can assure that eventually some ATL-formula φ will hold. The second item then expresses a *cooperative safety* property: A can ensure that φ is an invariant of the system. The third item is an example of what coalitions can achieve by forming bigger ones; although coalition A and B both cannot achieve that in the next state φ will be true, if they *joined their forces*, they would have a strategy to enforce φ in the next state. Finally, the last property expresses that a does not need any partner from A to achieve that φ will hold in the next state: read as a scheme, it says that whatever A together with a can achieve next, can be achieved by a on his own.

ATEL* (ATEL) adds to ATL* (ATL, respectively) operators for representing knowledge in the world of incomplete information. $K_a\varphi$ reads as "agent a knows that φ". Additional operators $C_A\varphi$, $E_A\varphi$ and $D_A\varphi$ refer to the situations of common knowledge, "everybody knows" situation, and distributed knowledge among the agents from A. Thus, $E_A\varphi$ means that every agent in A knows that φ holds.

[1] for an introduction to temporal logics see for example [9] or [30]. A survey of logic-based approaches to multi-agent systems can be found in [20].

[2] "Paths" refer to alternative courses of events; typically, paths are interpreted as sequences of successive states of computations.

[3] Note that the "sometime" operator $\Diamond$ can be defined in the usual way as: $\langle\!\langle A \rangle\!\rangle\Diamond\varphi \equiv \langle\!\langle A \rangle\!\rangle true\,\mathcal{U}\varphi$.

$C_A\varphi$ implies much more: the agents from A not only know that φ, but they also *know that they know* this, know that they know that they know, and so on. Distributed knowledge $D_A\varphi$ denotes a situation in which, if the agents could combine their individual beliefs together, they would be able to infer that φ holds. The complexity of model checking for ATEL is still polynomial [17].

Intuitively, ATEL should enable expressing various epistemic properties of agents under uncertainty:

1. $\langle\langle a\rangle\rangle\Diamond\varphi \rightarrow K_a\psi$
2. $K_b(c = s) \rightarrow \langle\langle b\rangle\rangle(\langle\langle b\rangle\rangle\bigcirc o)\,\mathcal{U}\neg(c = s)$
3. $d \rightarrow \langle\langle a\rangle\rangle\Diamond(K_a d \wedge \bigwedge_{a\neq b}\neg K_b d)$

The first two items are examples of so-called *knowledge pre-conditions*. The first of them intuitively says that knowing ψ is a *necessary* requirement for having the ability to bring about φ. The second expresses that if Bob (b) knows that the combination of the safe is s, then he is able to open it (o), as long as the combination remains unchanged. In [7], *Knowledge Games* are investigated as a particular way of learning in multiagent systems. Epistemic updates are interpreted in a simple card game, where the aim of the player is to find out a particular deal d of cards. Having a winning strategy then easily translates into the third item displayed above: it says that if the actual card deal is d, agent a can establish that eventually he will know it, without any of the others knowing this deal.

One of the main challenges in ATEL, not really addressed in [17] but already hinted upon in [22], is the question how, given an explicit way to represent the agent's knowledge, this should interfere with the agents' available strategies. What does it mean that an agent has a way to enforce φ, if he should therefore make different choices in epistemically indistinguishable states, for instance? In Section 3, we argue that in order to add an epistemic component to ATL, one should give an account of the tension between *incomplete information* that is imposed on the agents on the one hand, and *perfect recall* that is assumed about them when they are to make their decisions, on the other hand. We also argue that, when reasoning about what an agent can *enforce*, it seems more appropriate to require the agent knows his winning strategy rather than he knows only that such a strategy exists.

Then, in Section 4 we will loosen the assumption of perfect recall to agents having no, or only limited memory. The epistemic component in Alternating-time Temporal Observational Logic (ATOL) is entirely based on the notion of observation: the agents can recall no history of the game except of the information "stored" in their local states. We give several examples of what agents can achieve if they are allowed to make specific observations. Then, in Section 5, full Alternating-time Temporal Epistemic Logic with Recall (ATEL-R*) is considered; here, agents are again allowed to memorize the whole game. We propose a semantics for ATEL-R*, and we use past-time operators to relate the several epistemic modalities; finally, expressivity and complexity of ATEL-R* is briefly investigated. But first, in Section 2, we would like to re-introduce the basic systems: ATL and ATEL.

2. The Alternating-time Language: ATL and ATEL

The full language of Alternating-time Temporal Logic ATL* consists of state formulas and path formulas. A state formula is one of the following:

- p, where p is an atomic proposition;

- $\neg\varphi$ or $\varphi \vee \psi$, where φ, ψ are ATL* state formulas;
- $\langle\langle A \rangle\rangle \Phi$, where A is a set of agents, and Φ is an ATL* path formula.

A path formula is one of the following:

- an ATL* state formula;
- $\neg\varphi$ or $\varphi \vee \psi$, where φ, ψ are ATL* path formulas;
- $\bigcirc\varphi$ or $\varphi\mathcal{U}\psi$, where φ, ψ are ATL* path formulas.[4]

In "vanilla" ATL (i.e. ATL without *) it is required that every occurrence of a temporal operator is preceded by exactly one occurrence of a cooperation modality. In consequence, only state formulas can be found in ATL: p, $\neg\varphi$, $\varphi \vee \psi$, $\langle\langle A \rangle\rangle \bigcirc \varphi$, $\langle\langle A \rangle\rangle \Box \varphi$, and $\langle\langle A \rangle\rangle \varphi \mathcal{U} \psi$, where p is an atomic proposition, φ, ψ are ATL formulas, and A is a set of agents.

ATEL (ATEL*) adds to ATL (ATL*) formulas for describing epistemic properties of agents and groups of agents:

- $K_a\varphi$, where a is an agent and φ is a formula of ATEL (ATEL* state formula, respectively);
- $C_A\varphi$, $E_A\varphi$ and $D_A\varphi$, where A is a set of agents and φ is a formula of ATEL (ATEL* state formula, respectively).

2.1. Models for ATL: Concurrent Game Structures

A model for ATL is defined as a *concurrent game structure* [4]:

$$S = \langle k, Q, \Pi, \pi, d, \delta \rangle$$

where k is a natural number defining the amount of players (so the players are identified with numbers $1, ..., k$ and the set of players Σ can be taken to be $\{1, ..., k\}$), Q is a finite set of (global) states of the system; Π is the set of atomic propositions, and $\pi : Q \to 2^{\Pi}$ is a mapping that specifies which propositions are true in which states. The decisions available to player a at state q are labeled with consecutive natural numbers, and function $d : \Sigma \times Q \to \mathbb{N}$ specifies how many options are available for a particular agent at a particular state. Thus, agent a at state q can choose his decision from set $\{1, ..., d_a(q)\}$. Finally, a complete tuple of decisions $\langle j_1, ..., j_k \rangle$ in state q implies a deterministic transition according to the transition function $\delta(q, j_1, ..., j_k)$.

Example 2.1. As an illustration, consider a system with a single binary variable `x`. There are two processes: the controller (or server) s can enforce that the variable retains its value in the next step, or let the client change the value. The client c can request the value of `x` to be 0 or 1. The players proceed with their choices simultaneously — they do not know the other player's decision until the transition is done. The states and possible transitions of the system as a whole are shown in Figure 1.[5] There

[4] "Sometime" and "always" can be defined as: $\Diamond\varphi \equiv true\mathcal{U}\varphi$, and $\Box\varphi \equiv \neg\Diamond\neg\varphi$.

[5] We should use natural numbers as labels for agents and their actions to make the example correct in the formal sense. For instance, s can be encoded as "agent 1", c as "agent 2", action *reject* as 1, *accept* as 2, *set0* as 1 and *set1* as 2. Obviously, such a translation can be easily made for any set of symbolic labels — therefore we will use symbolic names in our examples to make them easier to read and interpret.

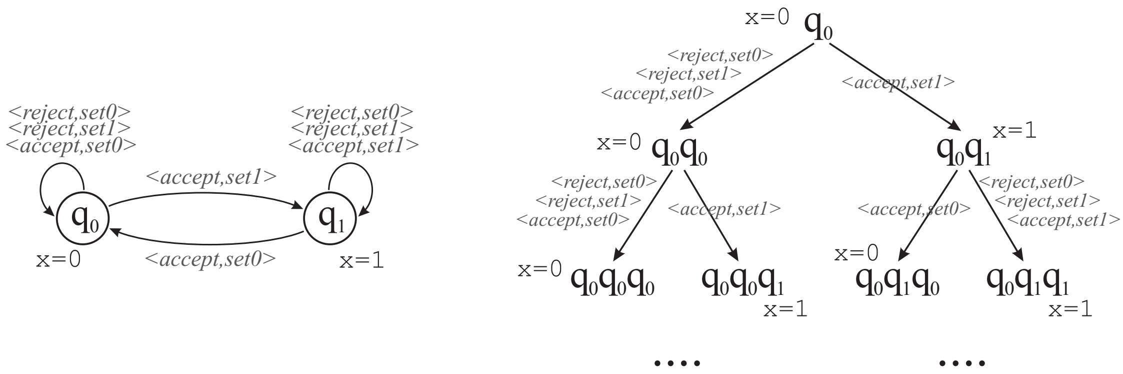


Figure 1. Transitions of the variable controller/client system, together with the tree of possible computations (assuming that q_0 is the initial state).

are two propositions available to observe the value of x: "x=0" and "x=1" (note: these are just atomic propositions, = is not the equality symbol here).

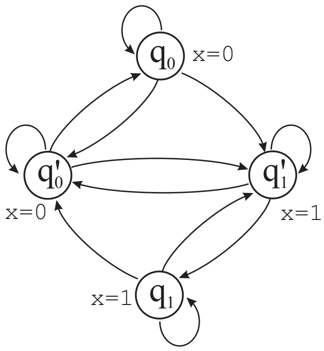
It is important to distinguish between the *computational structure*, defined explicitly in the model, and the *behavioral structure*, i.e. the model of how the system is supposed to behave in time [30]. In most temporal logics — ATL no exception — the computational structure is finite, while the implied behavioral structure is infinite. In ATL, the finite automaton lying at the core of every concurrent game structure can be seen as a way of imposing the tree of possible (infinite) computations that may occur in the system. The way the computational structure (concurrent game structure) unravels into a behavioral structure (computation tree) is shown in Figure 1, too.

It should be noted that at least three different versions of ATL have been proposed by Alur and colleagues over the course of the last 6 years, each with a slightly different definition of the semantic structure. The earliest version [2] includes only definitions for a synchronous turn-based structure and an asynchronous structure in which every transition is owned by a single agent. [3] offers more general structures (called alternating transitions systems) with no action labels and a sophisticated transition function. In [4], function d is introduced and δ simplified; moreover, an arbitrary finite set of agents Σ is replaced with set $\{1, ..., k\}$. All of this may lead to some confusion.

Since the models for ATEL from [17] are based on the second version of ATL [3] (and not the most recent one [4]), we will introduce the models from [3] as well. An *alternating transition system (ATS)* is a tuple

$$S = \langle \Sigma, Q, \Pi, \pi, \delta \rangle.$$

The type of the system transition function is the main difference here — $\delta : Q \times \Sigma \rightarrow 2^{2^Q}$ is meant to encode all the choices available to agents at each state. Now $\delta(q, a) = \{Q_1, ..., Q_n\}$ ($Q_1, ..., Q_n \subseteq Q$) defines the possible outcomes of a's decisions at state q, and the choices are identified with the outcomes. The resulting (global) transition of the whole system is assumed to be the intersection of choices from all the agents: $Q_{a_1} \cap ... \cap Q_{a_k}$, $Q_{a_i} \in \delta(q, a_i)$. Since the system is required to be deterministic (given the state and the agents' choices), $Q_{a_1} \cap ... \cap Q_{a_k}$ must always be a singleton.



$$\delta(q_0, s) = \delta(q_0', s) = \{\{q_0\}, \{q_0', q_1'\}\}$$
$$\delta(q_1, s) = \delta(q_1', s) = \{\{q_1\}, \{q_0', q_1'\}\}$$
$$\delta(q_0, c) = \delta(q_0', c) = \{\{q_0, q_0'\}, \{q_0, q_1'\}\}$$
$$\delta(q_1, c) = \delta(q_1', c) = \{\{q_1, q_0'\}, \{q_1, q_1'\}\}$$

Figure 2. An ATS for the controller/client problem

An alternating transition system for the variable controller/client problem is shown in Figure 2. Note that the ATS is somewhat more complex than the original concurrent game structure from Figure 1. In general, both kinds of semantics are equivalent [12, 13, 14], but the concurrent game structures are smaller and easier to read in most cases [22]. Therefore we will tend to use concurrent game structures rather than alternating transition systems as the basis for our analysis.

2.2. Agents' Strategies and Semantics of Cooperation Modalities

In a concurrent game structure, a strategy for agent a is a function $f_a : Q^+ \to \mathbb{N}$ such that $f_a(\lambda) \leq d_a(q)$ for q being the last state in sequence λ (i.e. such that $f_a(\lambda)$ is a valid decision in q). The function specifies a's decision for every possible (finite) history of system states. In other words, the strategy is a *conditional plan with perfect recall*, since each agent can base his decisions on a complete history $\lambda \in Q^+$.

A collective strategy is a tuple of strategies $F_A : A \to (Q^+ \to \mathbb{N})$, one strategy for each agent from $A \subseteq \Sigma$. The set of all possible (infinite) computations starting from state q, consistent with F_A is denoted with $out(q, F_A)$.[6] Now $S, q \models \langle\langle A\rangle\rangle\Phi$ iff there exists a collective strategy F_A such that Φ is satisfied for all computations from $out(q, F_A)$. In other words, no matter what the rest of the agents decides to do, the agents from A have a way of enforcing Φ. The semantics of temporal operators in ATL* is the same as in CTL*.

Semantics of ATL (without the star) must be defined in more detail – for every combination of a cooperation modality and a temporal operator. Let, for any history $\lambda \in Q^+$, and any integer $i \in \mathbb{N}$, the i-th position in λ be denoted by $\lambda[i]$. Let S be a concurrent game structure, and q a state in it. Then:

- $S, q \models \langle\langle A\rangle\rangle\bigcirc\varphi$ iff there exists a set of strategies F_A, one for each $a \in A$, such that for all $\lambda \in out(q, F_A)$, we have $S, \lambda[1] \models \varphi$;
- $S, q \models \langle\langle A\rangle\rangle\Box\varphi$ iff there exists a set of strategies F_A, one for each $a \in A$, such that for all $\lambda \in out(q, F_A)$, we have $S, \lambda[u] \models \varphi$ for all $u \in \mathbb{N}$;

[6]Computation $\Lambda = q_0 q_1 \ldots$ is consistent with a (collective) strategy F_A if, for every $i = 0, 1, \ldots$, there exists a tuple of agents' decisions $1 \leq j_a \leq d_a(q_i)$ such that $\delta(q_i, j_{a_1}, \ldots, j_{a_k}) = q_{i+1}$ and $j_a = F_A(a)(q_0 \ldots q_i)$ for each $a \in A$.

- $S, q \models \langle\langle A \rangle\rangle \varphi \mathcal{U} \psi$ iff there exists a set of strategies F_A, one for each $a \in A$, such that for all $\lambda \in out(q, F_A)$, there exists some $u \in \mathbb{N}$ such that $S, \lambda[u] \models \psi$, and for all $0 \leq v < u$, we have $S, \lambda[v] \models \varphi$;

The type of a choice is different in alternating transition systems, therefore a strategy for a has a different type too — $f_a : Q^+ \rightarrow 2^Q$. The strategy must be consistent with the choices available to a: $f_a(\lambda) \in \delta(q, a)$ for q being the last state in sequence λ. Again, $S, q \vDash \langle\langle A \rangle\rangle \Phi$ iff there exists a collective strategy F_A such that Φ is satisfied for all computations from $out(q, F_A)$.

Example 2.2. The following example ATL formulas are true in every state of the concurrent game structure from Figure 1:

- $(x = 0 \rightarrow \langle\langle s \rangle\rangle \bigcirc x = 0) \wedge (x = 1 \rightarrow \langle\langle s \rangle\rangle \bigcirc x = 1)$: the server can enforce the value of x to remain the same in the next step;
- $x = 0 \rightarrow \neg\langle\langle c \rangle\rangle \Diamond x = 1$: c cannot change the value from 0 to 1 on his own, even in multiple steps;
- $x = 0 \rightarrow \neg\langle\langle s \rangle\rangle \Diamond x = 1$: s cannot change the value on his own either;
- $x = 0 \rightarrow \langle\langle s, c \rangle\rangle \Diamond x = 1$: s and c can cooperate to change the value effectively.

2.3. AETS and Semantics of Epistemic Formulas

Models for ATEL — *alternating epistemic transition systems* (AETS) — add epistemic accessibility relations $\sim_1, ..., \sim_k \subseteq Q \times Q$ for expressing agents' beliefs [17]:

$$S = \langle \Sigma, Q, \Pi, \pi, \sim_1, ..., \sim_k, \delta \rangle.$$

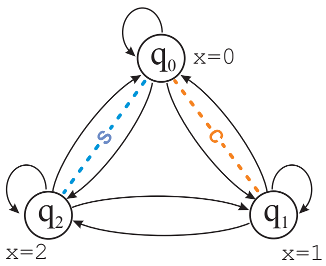
The accessibility relations are assumed to be reflexive, symmetric and transitive.

Agent a's epistemic relation is meant to encode a's inability to distinguish between the (global) system states: $q \sim_a q'$ means that, while the system is in state q, agent a cannot really determine whether it is in q or q'. The truth definition of knowledge is standard then. Let S be a concurrent game structure, and q and q' be states:

$$S, q \models K_a\varphi \text{ iff for all } q' \text{ such that } q \sim_a q' \text{ we have } S, q' \models \varphi$$

Relations $\sim_A^E$, $\sim_A^C$ and $\sim_A^D$, used to model "everybody knows" property, common knowledge and distributed knowledge, are derived from the epistemic accessibility relations of agents from A. First, $\sim_{\{a_1,...,a_n\}}^E$ is the union of all the $\sim_{a_i}$: $q \sim_{\{a_1,...,a_n\}}^E q'$ iff $q \sim_{a_i} q'$ for some a_i from group A. In other words, if everybody knows φ, then no agent may be unsure about the truth of it, and hence φ should be true in all the states that cannot be distinguished from the current state by even one member of the group. Next, $\sim_A^C$ can be defined as the reflexive and transitive closure of $\sim_A^E$ relation,[7] Finally, relation $\sim_{\{a_1,...,a_n\}}^D$ is the intersection of all the $\sim_{a_i}$: if any agent from group A can distinguish q from q', then

[7]Relation $\sim_A^C$ was originally defined as only the transitive closure of $\sim_A^E$ [17]. The reflexivity of the closure changes nothing here, since all $\sim_a$ are defined to be reflexive themselves — except for $A = \emptyset$. And that is exactly why we add it: now $\sim_\emptyset^C$ can be used to describe having complete information.



$$
\begin{array}{ll}
q_0 \sim_s q_0 & \\
q_0 \sim_s q_2 & \delta(q_0, s) = \{\{q_0\}, \{q_1, q_2\}\} \\
q_1 \sim_s q_1 & \delta(q_1, s) = \{\{q_1\}, \{q_0, q_2\}\} \\
q_2 \sim_s q_2 & \delta(q_2, s) = \{\{q_2\}, \{q_0, q_1\}\} \\
q_0 \sim_c q_0 & \delta(q_0, c) = \{\{q_0, q_1\}, \{q_0, q_2\}\} \\
q_0 \sim_c q_1 & \delta(q_1, c) = \{\{q_1, q_0\}, \{q_1, q_2\}\} \\
q_1 \sim_c q_1 & \delta(q_2, c) = \{\{q_2, q_0\}, \{q_2, q_1\}\} \\
q_2 \sim_c q_2 &
\end{array}
$$

Figure 3. An AETS for the modified controller/client problem. The dotted lines display the epistemic accessibility relations (modulo transitivity and reflexivity).

the whole group can distinguish the states in the sense of distributed knowledge. The semantics of group knowledge can be defined as below:

$$
\begin{array}{lll}
S, q \models E_A\varphi & \text{iff} & \text{for all } q' \text{ such that } q \sim_A^E q' \text{ we have } S, q' \models \varphi \\
S, q \models C_A\varphi & \text{iff} & \text{for all } q' \text{ such that } q \sim_A^C q' \text{ we have } S, q' \models \varphi \\
S, q \models D_A\varphi & \text{iff} & \text{for all } q' \text{ such that } q \sim_A^D q' \text{ we have } S, q' \models \varphi.
\end{array}
$$

Example 2.3. Let us consider another variation of the variable/controller example: the client can try to add 1 or 2 to the current value of x now (the addition is modulo 3 in this case). Thus the operations available to c are: "$x := x + 1 \mod 3$" and "$x := x + 2 \mod 3$". The server can still accept or reject the request from c (Figure 3). The dotted lines show that c cannot distinguish being in state q_0 from being in state q_1, while s is not able to discriminate q_0 from q_2.

Going from the model to the behavioral structure behind it, there are at least two ways of unraveling the alternating epistemic transition system into a computation tree with epistemic relations. If agents have no recall of the past, except for the information encapsulated in the current state (modulo relation $\sim$), then only the last state in a sequence matters for the epistemic accessibility links; if the agents can remember the history of previous states, then the whole sequence matters: the agents cannot discriminate two situations if they cannot distinguish any corresponding parts from the alternative histories (Figure 4 A and B). These two approaches reflect in fact two different "common-sense" interpretations of the computational structure with an epistemic component. In (A), a state (together with relation $\sim_a$) is meant to constitute the whole description of an agents' position, while in (B) states (and $\sim_a$) are more about what agents can perceive or observe at that point. More precisely, since agent c cannot distinguish q_0 from q_0q_0 in (A), he is not aware of any transition being happened that stays in q_0. In (B) however, indistinguishable situations occur always on the same level of tree, denoting that here the agents at least know how many transitions have been made.

Some properties that hold for this AETS are shown below:

- $q_1 \models K_s x = 1$ (or alternatively: $q \models x = 1 \rightarrow K_s x = 1$ for every q),
- $q_0 \models \langle\langle s \rangle\rangle \bigcirc x = 0$ (or alternatively: $x = 0 \rightarrow \langle\langle s \rangle\rangle \bigcirc x = 0$),

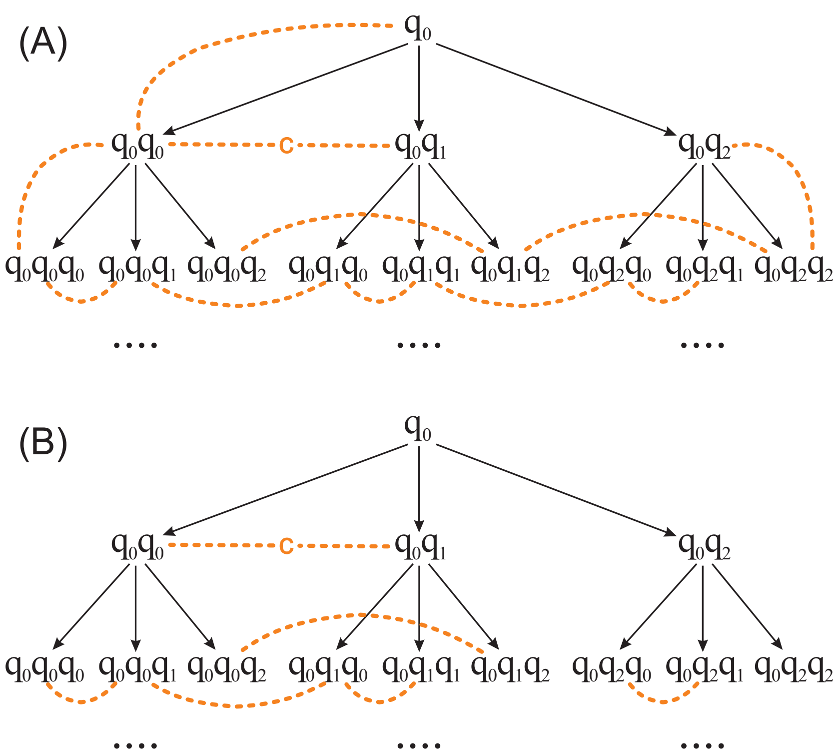


Figure 4. Unraveling: the computation trees with an epistemic relation for the client process. (A) indistinguishability relation based completely on $\sim_c$ — the agent does not remember the history of the game; (B) the client has perfect recall. The resulting indistinguishability relation should be read as the reflexive and transitive closure of the dotted arcs.

- $q_2 \models \langle\langle s, c\rangle\rangle \bigcirc x = 0$.

Note that ATEL agents are assumed to have perfect recall within the semantics of cooperation modalities: the knowledge available to agent a when he is choosing his action is determined by the type of strategy function f_a (which allows a to remember the whole history of previous states). Thus the epistemic abilities of agents with respect to their decision making should be the ones shown in Figure 4B. On the other hand, the knowledge modality K_a refers to indistinguishability of *states* — therefore its characteristics is rather displayed in Figure 4A.

Unfortunately, it can be also proved that $\neg x = 2 \rightarrow \langle\langle c\rangle\rangle \bigcirc \neg x = 2$ holds in the system from Figure 3 (because $q_0 \models \langle\langle c\rangle\rangle \bigcirc \neg x = 2$ and $q_1 \models \langle\langle c\rangle\rangle \bigcirc \neg x = 2$), which is counterintuitive: c cannot really choose a good strategy that the next transition will not lead to q_2 since he can never be sure whether the system is in q_0 or q_1. Asking about c's knowledge does not make things better: it can be proved that $K_c(\neg x = 2 \rightarrow \langle\langle c\rangle\rangle \bigcirc \neg x = 2)$, too! As it turns out, not every function of type $f_a : Q^+ \rightarrow 2^Q$ represents a feasible strategy under incomplete information. We will study the problem in more detail throughout the next section.

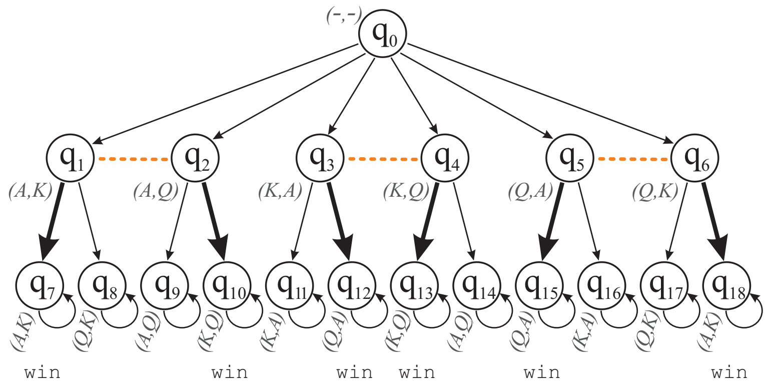
3. Knowledge and Action under Incomplete Information

ATEL and ATEL* are interesting languages to describe and verify properties of autonomous processes in situations of incomplete information. However, their semantics — the way it is defined in [17] — is not entirely consistent with the assumption that agents have incomplete information about the current state. Something seems to be lacking in the definition of a valid strategy for an agent in AETS. When defining a strategy, the agent can make his choices for every state independently. This is not feasible in a situation of incomplete information if the strategy is supposed to be deterministic: if a cannot recognize whether he is in situation s_1 or s_2, he cannot plan to proceed with one action in s_1, and another in s_2. Going back to Example 2.3, since the client cannot epistemically distinguish q_0 from q_1, and in both he should apply a different action to ensure that x will not have the value of 2 in the next state, it is not realistic to say that the client has a strategy to enforce $\bigcirc\neg(x = 2)$ in q_0. It is very much like with the information sets from von Neumann and Morgenstern [28]: for every state in an information set the same action must be chosen within a strategy. Such strategies are sometimes called *uniform* in the field of logic and games [5, 6].

Example 3.1. The following example can be considered: agent a plays a very simple card game against the environment env. The deck consisting of Ace, King and Queen (A, K, Q); it is assumed that A beats K, K beats Q, but Q beats A. First env gives a card to a, and assigns one card to itself. Then a can trade his card for the one remaining in the deck, or he can keep the current one. The player with the better card wins the game. A turn-based synchronous AETS for the game is shown in Figure 5. Right after the cards are given, a does not know what is the hand of the other player; for the rest of the game he has complete information about the state. Atomic proposition $\mathtt{win}$ enables to recognize the states in which a is the winner. States $q_7, ..., q_{18}$ are the final states for the this game; however, the transition function *must* specify at least one outgoing transition for each state. A reflexive arrow at every final state shows that – once the game is over – the system remains in that state forever.

Note that $q_0 \models \langle\langle a\rangle\rangle\Diamond\mathtt{win}$, although it should definitely be false for this game! Of course, a may *happen* to win, but he does not have the *power* to bring about winning because he has no way of recognizing the right decision until it is too late. Even if we ask about whether the player can *know* that he has a winning strategy, it does not help: $K_a\langle\langle a\rangle\rangle\Diamond\mathtt{win}$ is satisfied in q_0, too, because for all $q \in Q$ such that $q_0 \sim_a q$ we have $q \models \langle\langle a\rangle\rangle\Diamond\mathtt{win}$.

This calls for a constraint like the one from [28]: if two situations s_1 and s_2 are indistinguishable, then a strategy f_a must specify the same action for both s_1 and s_2. In order to accomplish this, some relation of "subjective unrecognizability" over the agents' choices can be useful — to tell which decisions will be considered the same in which states. Probably the easiest way to accomplish this is to provide the decisions with explicit labels — the way it has been done in concurrent game structures — and assume that the choices with the same label represent the same action from the agent's subjective point of view. This kind of solution fits also well in the tradition of game theory. Note that it is harder to specify this requirement if we identify agents' actions with their outcomes completely, because the same action started in two different states seldom generates the same result. If a trades his Ace in q_1, the system moves to q_8 and a loses the game; if he trades the card in q_2, the system moves to q_{10} and he wins. Still a cannot discriminate trading the Ace in both situations.



$\Sigma = \{a, env\}$
$\delta(q_0, a) = \{\{q_1, ..., q_6\}\}$
$\delta(q_1, a) = \{\{q_7\}, \{q_8\}\}$ etc.
$\delta(q_0, env) = \{\{q_1\}, ..., \{q_6\}\}$
$\delta(q_1, env) = \{\{q_7, q_8\}\}$ etc.
$\delta(q_7, a) = \delta(q_7, env) = \{\{q_7\}\}$ etc.

$q_0 \sim_a q_0$
$q_1 \sim_a q_1,\ q_1 \sim_a q_2,\ q_2 \sim_a q_2$
$q_3 \sim_a q_3,\ q_3 \sim_a q_4,\ q_4 \sim_a q_4$ etc.
$q_7 \sim_a q_7,\ q_8 \sim_a q_8,\ q_9 \sim_a q_9$ etc.
$q_0 \sim_{env} q_0,\ q_1 \sim_{env} q_1,\ q_2 \sim_{env} q_2$ etc.

Figure 5. Epistemic transition system for the card game. For every state, the players' hands are described. The dotted lines show a's epistemic accessibility relation $\sim_a$. The thick arrows indicate a's winning strategy.

3.1. Towards a Solution

The first attempt to solve the problem sketched above has been presented in [22]. The idea was to define ATEL models as tuples of the following shape:

$$S = \langle k, Q, \Pi, \pi, \sim_1, ..., \sim_k, d, \delta \rangle$$

where agents had the same choices available in indistinguishable states, i.e. for every q, q' such that $q \sim_a q'$ it was required that $d_a(q) = d_a(q')$ (otherwise a could distinguish q from q' by the decisions he could make).[8] An *incomplete information strategy* (we will follow [5, 6] and call it a *uniform* strategy within this paper) is a function $f_a : Q^+ \to \mathbb{N}$ for which the following constraints held:

- $f_a(\lambda) \le d_a(q)$, where q is the last state in sequence λ;
- if two histories are indistinguishable $\lambda \approx_a \lambda'$ then $f_a(\lambda) = f_a(\lambda')$.

Two histories are indistinguishable for a if he cannot distinguish their corresponding states. Recall that the ith position of λ is denoted by $\lambda[i]$. Then $\lambda \approx_a \lambda'$ iff $\lambda[i] \sim_a \lambda'[i]$ for every i. Alternatively, decisions can be specified for sequences of *local* states instead of global ones – $f_a : Q_a^+ \to \mathbb{N}$, where local states are defined as the equivalence classes of relation $\sim_a$, i.e. $Q_a = \{[q]_{\sim_a} \mid q \in Q\}$. This kind of presentation has been employed in [31], for example.

[8] The authors of ATEL suggested a similar requirement in [19]. They also considered whether some further constraint on the possible runs of the system should be added, but they dismissed the idea.

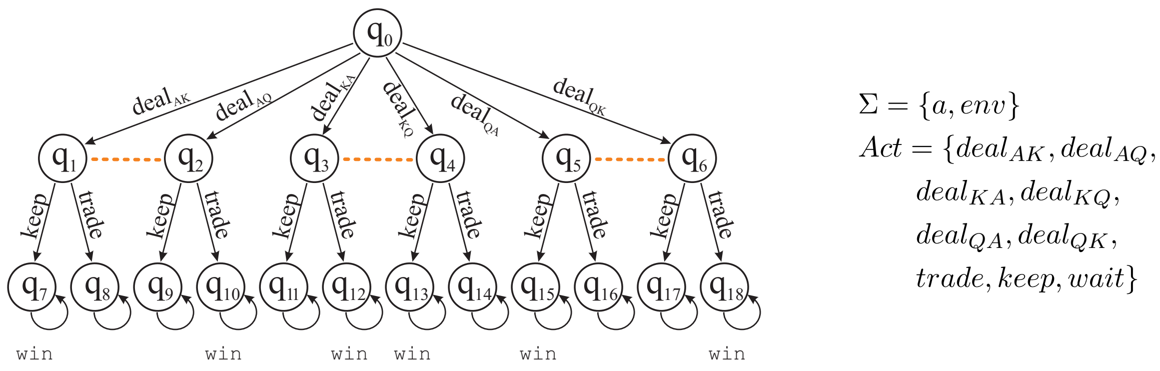


Figure 6. New AETS for the game. The transitions are labeled with decisions from the player who takes turn.

Example 3.2. A new transition system for the card game is shown in Figure 6. Now, using only uniform strategies, a is unable to bring about winning on his own: $q_0 \models \neg\langle\langle a\rangle\rangle\Diamond win$. Like in the real game, he can win only with some "help" from the environment: $q_0 \models \langle\langle a, env\rangle\rangle\Diamond win$.

Unfortunately, the new constraint proves insufficient for ruling out strategies that are not feasible under incomplete information. Consider the last game structure and state q_1. It is easy to show that $q_1 \models \langle\langle a\rangle\rangle\Diamond win$. Moreover, $q_0 \models \langle\langle\rangle\rangle\bigcirc\langle\langle a\rangle\rangle\Diamond win$, although still $q_0 \not\models \langle\langle a\rangle\rangle\Diamond win$. In other words, no conditional plan is possible for a at q_0, and at the same time he is bound to have one in the next step! The paradoxical results lead in fact to one fundamental question: *what does it mean for an agent to have a plan*?

3.2. Having a Strategy: *de re* vs. *de dicto*

The consecutive attempts to ATEL semantics seem to refer to various levels of "strategic" nondeterminism:

1. the first semantics proposed in [17] allows for *subjectively non-deterministic strategies* in a sense: the agent is allowed to guess which choice is the right one, and if there is any way for him to guess correctly, we are satisfied with this. Therefore the notion of a strategy from [17] makes formula $\langle\langle A\rangle\rangle\Phi$ describe what coalition A may *happen* to bring about against the most efficient enemies (i.e. when the enemies know the current state and even A's collective strategy beforehand), whereas the original idea from ATL was rather about A being able to *enforce* Φ;

2. in the updated semantics from [22], presented in the previous section, every strategy is subjectively deterministic (i.e. uniform), but the agent can choose non-deterministically between them (guess which one is right). This is because $\langle\langle a\rangle\rangle\Phi$ (in the updated version) is true if there is a consistent way of enforcing Φ, but the agent may be unaware of it, and unable to obtain it in consequence;

3. having restricted the set of strategies to the uniform ones, we can strengthen the notion of a successful strategy by requiring that a can enforce Φ only if $K_a\langle\langle a\rangle\rangle\Phi$; still, this is not enough as the examples showed. $K_a\langle\langle a\rangle\rangle\Phi$ implies that, for every q indistinguishable from the current state, a

must have a uniform strategy to achieve Φ from q – but these can be different strategies for different q's. Thus, $K_a\langle\langle a\rangle\rangle\Phi$ (in the updated version) is true if a knows there is a consistent way of enforcing Φ – unfortunately he is not required to know the way itself;

4. for planning purposes, the agent should be rather interested in having a strategy and *knowing it* (i.e. not only knowing that he has *some* strategy)!

The above hierarchy reminds the distinction between beliefs *de re* and beliefs *de dicto*. The issue is well known in the philosophy of language [29], as well as research on the interaction between knowledge and action [26, 27, 32]. Suppose we have dynamic logic-like modalities, parameterized with strategies: $[F_A]\Phi$ meaning "A can use strategy F_A to bring about Φ" (or: "every execution of F_A guarantees Φ"). Suppose also that strategies are required to be uniform. Cases (2), (3) and (4) above can be then described as follows:

- $\exists_{F_a}[F_a]\Phi$ is (possibly unaware) having a uniform strategy to achieve Φ – see point (2) above;
- $K_a\exists_{F_a}[F_a]\Phi$ is having a strategy *de dicto* (3);
- $\exists_{F_a}K_a[F_a]\Phi$ is having a strategy *de re* (4).

This would be a flexible way to express such subtleties. However – having extended ATEL this way – we would enable explicit quantification over strategies in the object language, and the resulting logic would be propositional no more. Instead, we can change the range of computations that are taken into account by the player when analyzing a strategy — out^* must include all the (infinite) paths that are possible from the agent's subjective perspective. Since strategies in ATEL are perfect recall strategies, the player must be able to use the information from the past during his analysis of possible future courses of action. Thus, the past history is relevant for determining the set of potential outcome paths for a strategy, and it plays an important role in the definition of out^*. Section 3.3 offers a more detailed discussion of this issue.

We need some terminology. Let λ be a variable over finite sequences of states, and let Λ denote an infinite sequence. Moreover, for any sequence $\xi = q_0q_1\dots$ (be it either finite or infinite):

- $\xi[i] = q_i$ is the ith position in ξ,
- $\xi_{|i} = q_0q_1\dots q_i$ denotes the first $i+1$ positions of ξ,
- $\xi^i = q_iq_{i+1}\dots$ is the ith suffix of ξ.

If i is greater than the length of $\xi + 1$, these notions are undefined. The length $\ell(\lambda)$ of a finite sequence λ is defined in a straightforward way.

Definition 3.1. Let λ be a finite non-empty sequence of states, and f_a a strategy for agent a. We say that Λ is a *feasible* computation run given finite history λ and agent a's strategy f_a, if the following holds:

- Λ starts with a sequence indistinguishable from λ, i.e. $\Lambda_{|n} \approx_a \lambda$, where $n = \ell(\lambda) - 1$,
- Λ is consistent with f_a. In fact, only the future part of Λ must be consistent with f_a since the past-oriented part of the strategy is irrelevant: no agent can plan the past.

Then, we define $out^*(\lambda, f_a) = \{\Lambda |\ \Lambda$ is feasible, given λ and $f_a\}$

If cooperation modalities are to reflect the property of having a strategy *de re*, then $out^*(\lambda, f_a)$ should replace the original set of objectively possible computations in the semantics of $\langle\langle a\rangle\rangle$, so that $\langle\langle a\rangle\rangle\Phi$ holds for a history λ iff there is an incomplete information strategy f_a such that Φ is true for every computation $\Lambda \in out^*(\lambda, f_a)$. Then the new semantics of the cooperation modality can be given as:

$\lambda \models \langle\langle a\rangle\rangle_{K(a)}\Phi$ iff a has a uniform strategy f_a such that for every $\Lambda \in out^*(\lambda, f_a)$ we have that Φ holds in Λ.

We use notation $\langle\langle a\rangle\rangle_{K(a)}$ to emphasize that these cooperation modalities differ from the original ones [4, 17]: agent a must have a uniform strategy and be able to identify it himself.

Example 3.3. Let us consider the card game example from Figure 6 again. Suppose q_0 has been the initial state and the system has moved to q_1 now, so the history is $\lambda = q_0q_1$. For every strategy f_a:

$$\begin{aligned} out^*(q_0q_1, f_a) &= \{\Lambda \mid \Lambda \text{ starts with } \lambda' \approx_a q_0q_1 \text{ and } \Lambda \text{ is consistent with } f_a\} \\ &= \{\Lambda \mid \Lambda \text{ starts with } q_0q, q \sim_a q_1 \text{ and } \Lambda \text{ is consistent with } f_a\} \\ &= \{\Lambda \mid \Lambda \text{ starts with } q_0q_1 \text{ or } q_0q_2 \text{ and } \Lambda \text{ is consistent with } f_a\}. \end{aligned}$$

Note that f_a must be a uniform strategy - in particular, $f_a(q_0q_1) = f_a(q_0q_2)$. There are two possible combinations of decisions for these histories:

(1) $f_1(q_0q_1) = f_1(q_0q_2) = keep$,
(2) $f_2(q_0q_1) = f_2(q_0q_2) = trade$.

Suppose there exists f such that for every $\lambda \in out^*(q_0q1, f)$ we have $\Diamond win$. We can check both cases:

case (1): $out^*(q_0q_1, f_1) = \{q_0q_1q_7q_7..., q_0q_2q_9q_9...\}$,
case (2): $out^*(q_0q_1, f_2) = \{q_0q_1q_8q_8..., q_0q_2q_1q_1...\}$.

Now, $\Diamond win$ does not hold for $q_0q_2q_9q_9...$ nor $q_0q_1q_8q_8...$, so $q_0q_1 \not\models \langle\langle a\rangle\rangle_{K(a)}\Diamond win$.

Note also that function out^* has a different type than the old function out, and that we interpreted formula $\langle\langle a\rangle\rangle_{K(a)}\Diamond win$ over a (finite) path and not a state in the above example. This shows another very important issue: epistemic properties of alternating-time systems with perfect recall are properties of *sequences of states* rather than single states.

3.3. Knowledge as Past-Related Phenomenon

Throughout the preceding sections the term "situation" was used in many places instead of "state". This was deliberate. The implicit assumption that states characterize epistemic properties of agents (expressed via the semantics of knowledge operators K_a, C_A etc. in the original version of ATEL) is probably one of the confusion sources about the logic. In concurrent game structures a *state* is not a complete description of a *situation* when the agent can remember the whole history of the game (as the type of agents' strategies suggest). Note that in the classical game theory models [28] situations do correspond to states — but these are computation trees that are used there, so every state in the tree uniquely identifies a path in it as well. At the same time a concurrent game structure or an alternating transition system is based on a finite automaton that indirectly imposes a tree of possible computations. A node in the tree corresponds to a *sequence of states* in the automaton (a history).

Within the original ATEL, agents are assumed different epistemic capabilities when making decisions, from what is actually being expressed with the epistemic operator K_a. The interpretation of knowledge operators refers to the agents' capability to distinguish one *state* from another; the semantics of $\langle\langle A \rangle\rangle$ allows the agents to base their decisions upon *sequences* of states. This dichotomy reflects the way a concurrent epistemic game structure can be unraveled (Figure 4 in section 2.3). We believe that the dilemma whether to assign agents with the ability to remember the whole history should be made explicit in the meta-language. Therefore we will assume that relation $\sim_a$ expresses what agent a can "see" (or observe) directly from his current state (i.e. having no recall of the past except for the information that is actually "stored" in the agent's local state), and we will call it an *observational accessibility relation* to avoid confusion. The *(perfect) recall accessibility relation* for agents that do not forget can be derived from $\sim_a$ in the form of relation $\approx_a$ over histories.

As the past is important when it comes to epistemic state of agents with perfect recall, knowledge operators should be given semantics in which the past is included. Thus, formulas like $K_a\varphi$ should be interpreted over paths rather than states of the system. The new semantics we propose for ATEL* in section 5 (meant as a logic for agents with finite set of states and perfect recall) draws much inspiration from branching-time logics that incorporate past in their scope [25]. The simpler case — agents with bounded memory — is also interesting. We will discuss it in section 4, proposing a logic aimed at observational properties of agents.

3.4. Feasible Strategies for Groups of Agents

The dichotomy between having a strategy *de re* and *de dicto* was discussed in section 3.2. The first notion is arguably more important if we want to express what agents with incomplete information can really *enforce*. In order to restrict the semantics of the cooperation modalities to feasible plans only, we suggest to rule out strategies with choices that cannot be deterministically executed by the players (via redefinition of the set of strategies available to agents) and to require that a player is able to identify a winning strategy (via redefinition of function out: all the computations must be considered that are possible from the agent's perspective — and not only the objectively possible ones).

This looks relatively straightforward for a single agent: $\langle\langle a \rangle\rangle_{K(a)}\Phi$ should mean: "a has a uniform strategy to enforce Φ and he knows that if he executes the strategy then he will bring about Φ" (cf. Definition 3.1 and Example 3.3). In such a case, there is nothing that can prevent a from executing it. The situation is not so simple for a coalition of agents. The coalition should be able to identify a winning strategy — but in what way? Suppose we require that this is common knowledge among the agents that F_A is a winning strategy — would that be enough? Unfortunately, the answer is no.

Example 3.4. Consider the following variant of the *matching pennies* game. There are two agents – both with a coin – and each can choose to show heads or tails. If they choose the same, they win, otherwise they loose. There are two obvious collective strategies that result in victory for the coalition, even when we consider common knowledge *de re*; hence $\exists_{F_{\{1,2\}}} C_{\{1,2\}}[F_{\{1,2\}}]\mathtt{win}$. However, both agents have to choose *the same* winning strategy, so it is still hard for them to win this game! In fact, they cannot play it successfully with no additional communication between them.

Thus, even common knowledge among A of a winning strategy F_A for them does not imply that the agents from A can automatically apply F_A as long as there are other winning strategies commonly identified by A. It means that the coalition must have a strategy selection criterion upon which all agents from A agree. How have they come to this agreement? Through some additional communication "outside the model"? But why should not distributed knowledge be used instead then – if the agents are allowed to communicate outside the model at all, perhaps they can share their private knowledge too? Other settings make sense as well: there can be a leader within the team that can assign the rest of the team with their strategies (then it is sufficient that the strategy is identified by the leader). Or, the leader may even stay out of the group (then he is not a member of the coalition that executes the plan). In order to capture the above intuitions in a general way, we propose to extend the simple cooperation modality $\langle\langle A\rangle\rangle$ to a family of operators: $\langle\langle A\rangle\rangle_{\mathcal{K}(\Gamma)}\Phi$ with the intended meaning that coalition A has a (uniform) strategy to enforce Φ, and the strategy can be identified by agents $\Gamma \subseteq \Sigma$ in epistemic mode $\mathcal{K}$ (where $\mathcal{K}$ can be any of the epistemic operators K, C, E, D):

$\lambda \models \langle\langle A\rangle\rangle_{\mathcal{K}(\Gamma)}\Phi$ iff A have a collective uniform strategy F_A such that for every $\Lambda \in out^*_{\mathcal{K}(\Gamma)}(\lambda, F_A)$ we have that Φ holds in Λ.

These operators generalize Jonker's cooperation modalities with indices: $\langle\langle A\rangle\rangle_C$, $\langle\langle A\rangle\rangle_E$ and $\langle\langle A\rangle\rangle_{K_i}$, introduced in [23].

We will use the generic notation $\approx^{\mathcal{K}}_{\Gamma}$ to denote the (path) indistinguishability relation for agents Γ in epistemic mode $\mathcal{K}$:

$$\lambda \approx^{\mathcal{K}}_{\Gamma} \lambda' \quad \text{iff} \quad \lambda[i] \sim^{\mathcal{K}}_{\Gamma} \lambda'[i] \text{ for every } i.$$

Function $out^*_{\mathcal{K}(\Gamma)}(\lambda, F_A)$ returns the computations that are possible from the viewpoint of group Γ (with respect to knowledge operator $\mathcal{K}$) after history λ took place:

$$out^*_{\mathcal{K}(\Gamma)}(\lambda, F_A) = \{\Lambda |\ \Lambda \text{ starts with } \lambda' \text{ such that } \lambda' \approx^{\mathcal{K}}_{\Gamma} \lambda, \text{ and the rest of } \Lambda \text{ is consistent with } F_A\}$$

Examples include:

- $\langle\langle A\rangle\rangle_{C(A)}\Phi$: the agents from A have a collective strategy to enforce Φ and the strategy is common knowledge in A. This requires the least amount of additional communication. It is in fact sufficient that the agents from A agree upon some total order over their group strategies at the beginning of the game (the lexicographical order, for instance) and that they will always choose the maximal winning strategy with respect to this order;

- $\langle\langle A\rangle\rangle_{E(A)}\Phi$: coalition A has a collective strategy to enforce Φ and everybody in A knows that the strategy is winning;

- $\langle\langle A\rangle\rangle_{D(A)}\Phi$: the agents from A have a strategy to enforce Φ and if they share their knowledge at the current state, they can identify the strategy as winning;

- $\langle\langle A\rangle\rangle_{K(a)}\Phi$: the agents from A have a strategy to enforce Φ, and a can identify the strategy and give them orders how to achieve the goal;

- $\langle\langle A\rangle\rangle_{D(\Gamma)}\Phi$: group Γ acts as a kind of "headquarters committee": they can fully cooperate within Γ (at the current state) to find a strategy to achieve Φ. The strategy is aimed for A, so it must be uniform for agents from A.

Note also that $\langle\langle A\rangle\rangle_{C(\emptyset)}\Phi$ means that A have a uniform strategy to achieve Φ (but they may be unaware of it, and of the strategy itself), because $\sim^C_\emptyset$ is the accessibility relation when complete information is available. In consequence, $\mathcal{K}_A\langle\langle A\rangle\rangle_{C(\emptyset)}\Phi$ captures the notion of having a strategy *de dicto* from section 3.2. Since the original ATL meaning of $\langle\langle A\rangle\rangle\Phi$ (there is a *complete information* strategy to accomplish Φ) does not seem to be expressible with the new modalities, we suggest to leave the operator in the language as well.

Example 3.5. Let us consider the modified variable client/server system from Figure 3 once more to show how the new modalities work:

- $x = 1 \rightarrow \langle\langle s\rangle\rangle_{K(s)}\bigcirc x = 1$, because every time s is in q_1, he can choose to reject the client's request (and he knows it, because he can distinguish q_1 from the other states);
- $\neg x = 2 \rightarrow \neg\langle\langle s, c\rangle\rangle_{K(c)}\bigcirc x = 2$, because – for every history $(qq'q''...q_1)$ – c cannot distinguish it from $(qq'q''...q_0)$ and vice versa, so he cannot effectively identify a uniform strategy;
- $x = 2 \rightarrow \neg\langle\langle s\rangle\rangle_{K(s)}\bigcirc x = 2 \wedge \neg\langle\langle c\rangle\rangle_{K(c)}\bigcirc x = 2$, because c has no action to request no change, and s is unable to identify the current state;
- however, $x = 2 \rightarrow \langle\langle s\rangle\rangle_{K(c)}\bigcirc x = 2$! The client can "indicate" the right strategy to the server;
- $x = 0 \rightarrow \neg\langle\langle s\rangle\rangle_{K(s)}\bigcirc x = 0 \wedge \neg\langle\langle s\rangle\rangle_{K(c)}\bigcirc x = 0 \wedge \langle\langle s\rangle\rangle_{D(\{s,c\})}\bigcirc x = 0$: only if s and c join their pieces of knowledge, they can identify a feasible strategy for s in q_0;
- $x = 1 \rightarrow \langle\langle s, c\rangle\rangle_{E(\{c,s\})}\bigcirc\neg x = 0 \wedge \neg\langle\langle c, s\rangle\rangle_{C(\{s,c\})}\bigcirc\neg x = 0$: both processes can identify a suitable collective strategy, but they are not sure if the other party can identify it too.

The next two sections follow with a formalization of the intuitions described so far.

4. ATOL: a Logic of Observations

Assigning an agent the ability to remember everything that has happened in the past seems unrealistic in many cases. Both humans and software agents have obviously limited memory capabilities. On the other hand, we usually cannot know precisely what the agents in question will actually remember from their history – in such situations perfect recall can be attractive as the upper bound approximation of the agents' potential. Some agents may also enhance their capacity (install new memory chips when more storage space is needed, for instance). In this case the memory of the agents is finite, but not bounded, and they cannot be appropriately modeled with bounded recall apparatus.

We believe that both settings are interesting and worth further investigation. In this section, we start with introducing the simpler case of imperfect recall in the form of Alternating-time Temporal Observational Logic (ATOL). As the original ATL and ATEL operators $\langle\langle A\rangle\rangle$ were defined to describe agents with perfect recall, it seems best to leave them with this meaning. Instead, we will use a new modality $\langle\langle A\rangle\rangle^\bullet$ to express that the agents from A can enforce a property while their ability to remember is bounded. When uniform strategies are to be considered, the operator will be used with an appropriate subscript in the way proposed in Sections 3.2 and 3.4.

If agents are assumed to remember no more than p most recent positions in a finite automaton, a new automaton can be proposed in which the last p positions are included in the states and the epistemic links define what the agents actually remember in every situation. Thus, for every model in which the agents can remember a limited number of past events, an equivalent model can be constructed in which they can recall no past at all (cf. Example 4.2). ATOL is a logic for agents with no recall – it refers to the features that agents can *observe* on the spot. Note, however, that these are observations in the broadest sense, including perceptions of the external world, and the internal (local) state of the agent.

4.1. Syntax

An ATOL formula is one of the following:

- p, where p is an atomic proposition;
- $\neg\varphi$ or $\varphi \vee \psi$, where φ, ψ are ATOL formulas;
- $Obs_a\varphi$, where a is an agent and φ is a formula of ATOL;
- $CO_A\varphi$, $EO_A\varphi$ and $DO_A\varphi$, where A is a set of agents and φ is a formula of ATOL;
- $\langle\langle A\rangle\rangle^{\bullet}_{Obs(\gamma)}\bigcirc\varphi$, $\langle\langle A\rangle\rangle^{\bullet}_{Obs(\gamma)}\Box\varphi$, or $\langle\langle A\rangle\rangle^{\bullet}_{Obs(\gamma)}\varphi\mathcal{U}\psi$, where φ, ψ are ATOL formulas and A is a set of agents, and γ an agent (not necessarily a member of A).
- $\langle\langle A\rangle\rangle^{\bullet}_{\Theta(\Gamma)}\bigcirc\varphi$, $\langle\langle A\rangle\rangle^{\bullet}_{\Theta(\Gamma)}\Box\varphi$, $\langle\langle A\rangle\rangle^{\bullet}_{\Theta(\Gamma)}\varphi\mathcal{U}\psi$, where φ, ψ are ATOL formulas and A and Γ are sets of agents and $\Theta(\Gamma) \in \{CO(\Gamma), DO(\Gamma), EO(\Gamma)\}$.

Formula $Obs_a\varphi$ reads: "agent a observes that φ". Operators CO_A, EO_A and DO_A refer to "common observation", "everybody sees" and "distributed observation" modalities. The informal meaning of $\langle\langle A\rangle\rangle^{\bullet}_{Obs(\gamma)}\Phi$ is: "group A has a strategy to enforce Φ, and agent γ can see the strategy". The common sense reading of $\langle\langle A\rangle\rangle^{\bullet}_{CO(\Gamma)}\Phi$ is that coalition A has a collective strategy to enforce Φ, and the strategy itself is a common observation for group Γ. The meaning of $\langle\langle A\rangle\rangle^{\bullet}_{EO(\Gamma)}\Phi$ and $\langle\langle A\rangle\rangle^{\bullet}_{DO(\Gamma)}\Phi$ is analogous. Since the agents are assumed to have no recall in ATOL, the choices they make within their strategies must be based on the current state only. As we want them to specify deterministic plans under incomplete information, the plans should be uniform strategies as well.

Note that ATOL contains only formulas for which the past is irrelevant and no specific future branch is referred to, so it is sufficient to evaluate them over single states of the system.

4.2. Semantics

Formulas of Alternating-time Temporal Observational Logic are interpreted in *concurrent observational game structures*:

$$S = \langle k, Q, \Pi, \pi, \sim_1, ..., \sim_k, d, \delta\rangle$$

in which agents have the same choices in indistinguishable states: for every q, q' such that $q \sim_a q'$ it is required that $d_a(q) = d_a(q')$. To specify plans, they can use uniform strategies with no recall.

Definition 4.1. An *uniform strategy with no recall* is a function $v_a : Q \to \mathbb{N}$ for which:

- $v_a(q) \leq d_a(q)$ (the strategy specifies valid decisions),
- if two states are indistinguishable $q \sim_a q'$ then $v_a(q) = v_a(q')$.

As usually, a collective strategy V_A assigns every agent $a \in A$ with one strategy v_a. The group observational accessibility relations can also be defined in the standard way:

$$\begin{array}{lcl} \sim_A^{DO} & = & \bigcap_{a \in A} \sim_a; \\ \sim_A^{EO} & = & \bigcup_{a \in A} \sim_a; \\ \sim_A^{CO} & \text{is} & \text{the reflexive and transitive closure of } \sim_A^{EO}. \end{array}$$

The set of computations that are possible from agent γ's point of view, consistent with strategy V_A and starting from state q, can be defined as:

$$out_{Obs(\gamma)}(q, V_A) = \{\Lambda |\ \Lambda \text{ is consistent with } V_A \text{ and } \Lambda[0] \sim_\gamma q\}.$$

Definition 4.2. More generally, for $\Gamma \subseteq \Sigma$, and Θ being any of the collective observation modes CO, EO, DO:

$$out_{\Theta(\Gamma)}(q, V_A) = \{\Lambda |\ \Lambda \text{ is consistent with } V_A \text{ and } \Lambda[0] \sim_\Gamma^\Theta q\},$$

Definition 4.3. We define the semantics of ATOL with the following rules:

$q \models p$	iff	$p \in \pi(q)$
$q \models \neg\varphi$	iff	$q \not\models \varphi$
$q \models \varphi \vee \psi$	iff	$q \models \varphi$ or $q \models \psi$
$q \models Obs_a\, \varphi$	iff	for every $q' \sim_a q$ we have $q' \models \varphi$
$q \models \langle\langle A \rangle\rangle^{\bullet}_{Obs(\gamma)} \bigcirc \varphi$	iff	there is a strategy V_A such that for every $\Lambda \in out_{Obs(\gamma)}(q, V_A)$ we have $\Lambda[1] \models \varphi$
$q \models \langle\langle A \rangle\rangle^{\bullet}_{Obs(\gamma)} \Box \varphi$	iff	there is a strategy V_A such that for every $\Lambda \in out_{Obs(\gamma)}(q, V_A)$ we have $\Lambda[i] \models \varphi$ for all $i = 0, 1, \ldots$
$q \models \langle\langle A \rangle\rangle^{\bullet}_{Obs(\gamma)} \varphi \mathcal{U} \psi$	iff	there is a strategy V_A such that for every $\Lambda \in out_{Obs(\gamma)}(q, V_A)$ there is a $k \geq 0$ such that $\Lambda[k] \models \psi$ and $\Lambda[i] \models \varphi$ for all $0 \leq i \leq k$
$q \models \Theta_A\, \varphi$	iff	for every $q' \sim_A^\Theta q$ we have $q' \models \varphi$
$q \models \langle\langle A \rangle\rangle^{\bullet}_{\Theta(\Gamma)} \bigcirc \varphi$	iff	there is a strategy V_A such that for every $\Lambda \in out_{\Theta(\Gamma)}(q, V_A)$ we have $\Lambda[1] \models \varphi$
$q \models \langle\langle A \rangle\rangle^{\bullet}_{\Theta(\Gamma)} \Box \varphi$	iff	there is a strategy V_A such that for every $\Lambda \in out_{\Theta(\Gamma)}(q, V_A)$ we have $\Lambda[i] \models \varphi$ for all $i = 0, 1, \ldots$
$q \models \langle\langle A \rangle\rangle^{\bullet}_{\Theta(\Gamma)} \varphi \mathcal{U} \psi$	iff	there is a strategy V_A such that for every $\Lambda \in out_{\Theta(\Gamma)}(q, V_A)$ there is a $k \geq 0$ such that $\Lambda[k] \models \psi$ and $\Lambda[i] \models \varphi$ for all $0 \leq i \leq k$

Remark 4.1. Note that operators Obs_a and $\langle\langle A \rangle\rangle^{\bullet}_{Obs(\gamma)}$ are in fact redundant:

- $Obs_a\varphi \equiv CO_{\{a\}}\varphi$;
- $\langle\langle A\rangle\rangle^{\bullet}_{Obs(\gamma)}\bigcirc\varphi \equiv \langle\langle A\rangle\rangle^{\bullet}_{CO(\{\gamma\})}\bigcirc\varphi$, $\langle\langle A\rangle\rangle^{\bullet}_{Obs(\gamma)}\Box\varphi \equiv \langle\langle A\rangle\rangle^{\bullet}_{CO(\{\gamma\})}\Box\varphi$, and $\langle\langle A\rangle\rangle^{\bullet}_{Obs(\gamma)}\varphi\mathcal{U}\psi \equiv \langle\langle A\rangle\rangle^{\bullet}_{CO(\{\gamma\})}\varphi\mathcal{U}\psi$.

Remark 4.2. ATOL generalizes ATL_{ir} from [31]. The other paper introduces four basic variants of the alternating-time logic, that differ in their treatment of information available to agents, and the type of agents' recall. The actual kind of agents being treated by a particular variant is reflected in the indices added to the name of the logic as well as to the cooperation modalities: I stands for perfect information, and i for imperfect information; similarly, R denotes perfect, and r – imperfect recall. Thus, ATL_{IR} is about agents with perfect information and perfect recall (hence being equivalent to the original ATL), ATL_{iR} treats agents with imperfect information and perfect recall, and so on.

ATL_{ir} encodes the same view to agents as ATOL: the agents have imperfect recall (they use strategies based on states rather than histories) and incomplete information (modeled via epistemic accessibility relations over states). ATOL, however, is more expressive: not only it includes the epistemic operators, but the basic modalities of ATL_{ir} can be seen as special cases of the ATOL modalities:

$$\langle\langle A\rangle\rangle_{ir}\Phi \equiv \langle\langle A\rangle\rangle^{\bullet}_{EO(A)}\Phi.$$

Thus, it seems that ATOL allows to distinguish more subtle cases of having a (collective) strategy, and does it in a more explicit way.

Proposition 4.1. Model checking ATOL is NP-hard and $\Delta_2 P$-easy (i.e., ATOL model checking falls between nondeterministic polynomial time complexity, and complexity of polynomial calls to a nondeterministic polynomial oracle).

Proof:
The proof is analogous to the respective proofs for ATL_{ir} from [31]. □

Remark 4.3. ATOL syntactically subsumes most of CTL. Although none of $\langle\langle\Sigma\rangle\rangle^{\bullet}_{\Theta(\Gamma)}$ is equivalent to the CTL's E, yet still the universal path quantifier A can be expressed with $\langle\langle\emptyset\rangle\rangle^{\bullet}_{CO(\emptyset)}$. Thus also most of "there is a path" formulas can also be redefined: $\mathsf{E}\bigcirc\varphi \equiv \neg\langle\langle\emptyset\rangle\rangle^{\bullet}_{CO(\emptyset)}\bigcirc\neg\varphi$, $\mathsf{E}\Box\varphi \equiv \neg\langle\langle\emptyset\rangle\rangle^{\bullet}_{CO(\emptyset)}\Diamond\neg\varphi$, and $\mathsf{E}\Diamond\varphi \equiv \neg\langle\langle\emptyset\rangle\rangle^{\bullet}_{CO(\emptyset)}\Box\neg\varphi$.

Remark 4.4. The language of ATOL does *not* cover the expressive power of full CTL. Unlike in ATL (and even ATL_{Ir}), $\mathsf{E}\varphi\mathcal{U}\psi$ cannot be translated to $\langle\langle\Sigma\rangle\rangle^{\bullet}_{CO(\emptyset)}\varphi\mathcal{U}\psi$. Moreover, $\mathsf{E}\varphi\mathcal{U}\psi$ cannot be expressed as a combination of $\mathsf{A}\varphi\mathcal{U}\psi$, $\mathsf{E}\Diamond\varphi$, $\mathsf{E}\Box\varphi$, $\mathsf{A}\Box\varphi$, $\mathsf{E}\bigcirc\varphi$, and $\mathsf{A}\bigcirc\varphi$ (cf. [24]).

In consequence, the modalities for complete information (i.e. possibly non-uniform) strategies with no recall: $\langle\langle A\rangle\rangle^{\bullet}\bigcirc\varphi$, $\langle\langle A\rangle\rangle^{\bullet}\Box\varphi$ and $\langle\langle A\rangle\rangle^{\bullet}\varphi\mathcal{U}\psi$ (corresponding to the ATL_{Ir} logic [31]) can be a valuable extension of the basic ATOL.

Remark 4.5. Note that ATL_{Ir} is equivalent to ATL [31], so ATOL begins to cover the expressive power of CTL as soon as we add the perfect information modalities to ATOL.

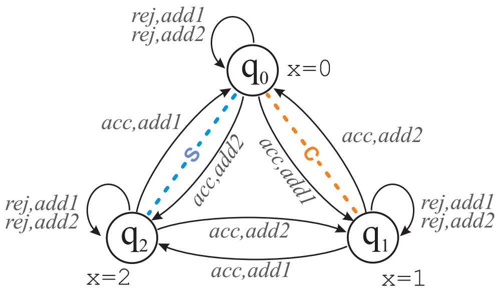


Figure 7. The controller/client problem again

Remark 4.6. ATOL semantically subsumes CTL: it is enough to restrict the alternating observational game structures to the case of one agent ($\Sigma = \{a\}$), with the identity relation as the only observational accessibility relation ($q \sim_a q'$ iff $q = q'$). For this class of models: $\mathsf{E}\bigcirc\varphi \equiv \langle\langle a\rangle\rangle^{\bullet}_{Obs(a)}\bigcirc\varphi$, $\mathsf{A}\bigcirc\varphi \equiv \langle\langle\emptyset\rangle\rangle^{\bullet}_{Obs(a)}\bigcirc\varphi$, $\mathsf{E}\Box\varphi \equiv \langle\langle a\rangle\rangle^{\bullet}_{Obs(a)}\Box\varphi$ and so on.

4.3. Examples

Let us consider a few examples to see how properties of agents and their coalitions can be expressed with ATOL. We believe that especially Example 4.3 demonstrates the potential of ATEL in reasoning about limitations of agents, and the ways they can be overcome.

Example 4.1. First, we can have a look at the variable client/server system from Example 2.3 again, this time in the form of a concurrent observational game structure (see Figure 7). Note how the observational relation is defined: if we think of x in binary representation x_1x_2, we have that c can observe x_1, whereas s observers x_2. The following formulas are valid in the system:

- $Obs_s x = 1 \vee Obs_s \neg x = 1$: the server can recognize whether the value of x is 1 or not;
- $\langle\langle s, c\rangle\rangle^{\bullet}_{CO(s,c)}\bigcirc\neg x = 2$: the agents have a strategy *de re* to avoid $x = 2$ in the next step. For instance, the client can always execute ***add1***, and the server rejects the request in q_1 and accepts otherwise;
- $x = 2 \rightarrow \neg\langle\langle s\rangle\rangle^{\bullet}_{Obs(s)}\bigcirc(x = 2) \wedge \langle\langle s\rangle\rangle^{\bullet}_{Obs(c)}\bigcirc(x = 2)$: The server s must be hinted a strategy by c if he wants the variable to retain the value of 2. To see why this is true, suppose that $x = 2$. We have to find a strategy v_s such that for every $\Lambda \in out_{Obs(c)}(q_2, v_s)$, we have $\Lambda[1] \models \bigcirc(x = 2)$. Let v_s be the strategy that picks ***rej*** in all states. Then, obviously, v_s is an incomplete information strategy. All the computation paths consistent with this strategy are $q_0q_0\dots, q_1q_1\dots$ and $q_2q_2\dots$. The runs from those that are in $out_{Obs(c)}(q_2, v_s)$ are those that start in q_2, so the only element we retain is $\Lambda = q_2q_2\dots$. Obviously, for this Λ, we have $\Lambda[1] \models (x = 2)$. To further see that in q_2 we have $\neg\langle\langle s\rangle\rangle^{\bullet}_{Obs(s)}\bigcirc(x = 2)$, assume that there is some strategy v_s such that for every $\Lambda \in out_{Obs(c)}(q_2, v_s)$ we have $\Lambda[1] \models (x = 2)$. The only strategy v_s that works here chooses ***rej*** in q_2. Since v_s has to be an incomplete information strategy, v_s prescribes ***rej*** in q_2 as well. But the runs generated by this v_s in $out_{Obs(c)}(q_2, v_s)$ are $\Lambda = q_2q_2\dots$ and $\Lambda' = q_0q_0\dots$. Obviously, we do not have $\Lambda'[1] \models x = 2$;

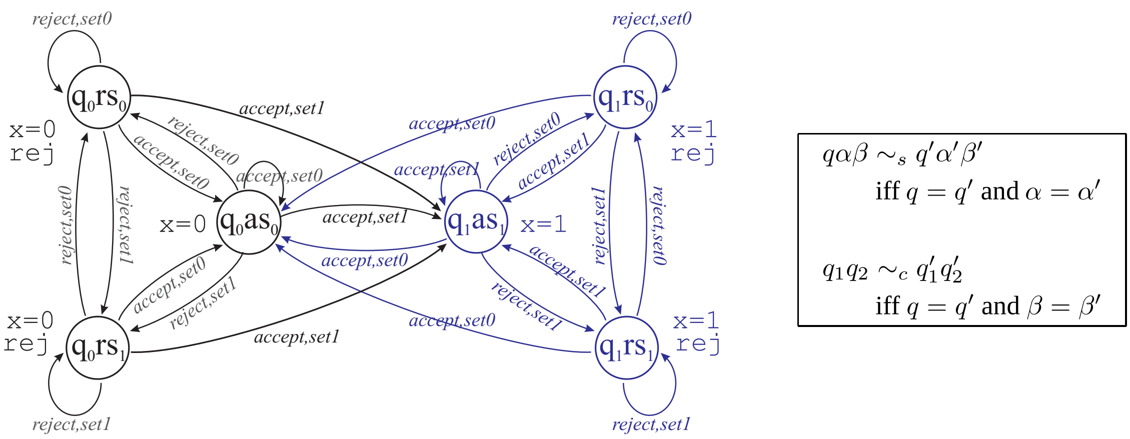


Figure 8. Agents with some memory of the past. Proposition `rej` holds in the states immediately after a request has been rejected.

- $\langle\langle s, c\rangle\rangle^{\bullet}_{CO(s,c)} \Diamond(Obs_s x = 0 \vee Obs_s x = 1 \vee Obs_s x = 2) \wedge \langle\langle s, c\rangle\rangle^{\bullet}_{CO(s,c)} \Diamond(Obs_c x = 0 \vee Obs_c x = 1 \vee Obs_c x = 2) \wedge \neg\langle\langle s, c\rangle\rangle^{\bullet}_{CO(s,c)} \Diamond(EO_{\{s,c\}} x = 0 \vee EO_{\{s,c\}} x = 1 \vee EO_{\{s,c\}} x = 2)$: the agents have a way to make the value of x observable for any of them, but they have no strategy to make it observable to both of them at the same moment.

Example 4.2. Let us go back to the first variable/controller system with only two states (Example 2.1). The system can be modified to include bounded memory of the players: for instance, it seems reasonable to assume that each agent remembers at least the last decision he made. Resulting concurrent observational game structure is shown in Figure 8. For this structure, we may for instance demonstrate that:

- s can always reject the claim: $\mathsf{A}\Box\langle\langle s\rangle\rangle^{\bullet}_{Obs(s)} \bigcirc rej$ (where $\mathsf{A} \equiv \langle\langle\emptyset\rangle\rangle^{\bullet}_{CO(\emptyset)}$ – cf. Remark 4.3);
- if s rejects the claims then the value of x will not change – and s can see it: $Obs_s[(x = 0 \rightarrow \mathsf{A}\bigcirc(rej \rightarrow x = 0)) \wedge (x = 1 \rightarrow \mathsf{A}\bigcirc(rej \rightarrow x = 1)) \wedge (x = 2 \rightarrow \mathsf{A}\bigcirc(rej \rightarrow x = 2))]$. Note that this kind of formulas can be used in ATOL to specify results of particular strategies in the object language (in this case: the "always reject" strategy).

Example 4.3. Let us consider a train controller example similar to the one from [1, 17]. There are two trains tr_1, tr_2, and a controller c that can let them into the tunnel. The algorithm of train tr_i is sketched in Figure 9. Each train can opt to stay out of the tunnel (action s) for some time – its local state is "away" (a_i) then. When the train wants to enter the tunnel (e), it must wait (state w_i) until the controller lights a green light for it (action let_i from the controller). In the tunnel (t_i), the train can again decide to stay for some time (s) or to exit (e). There is enough vocabulary to talk about the position of each train (propositions `a1`,`w1`,`t1`,`a2`, `w2` and `t2`).

The set of possible situations (global states) is

$$Q = \{a_1a_2, a_1w_2, a_1t_2, w_1a_2, w_1w_2, w_1t_2, t_1a_2, t_1w_2, t_1t_2\}.$$

The transition function for the whole system, and the accessibility relations are depicted in Figure 10. Every train can observe only its own position. The controller is not very capable observationally: it can

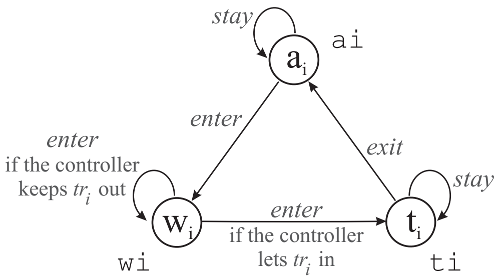


Figure 9. Train template: tr_i for the train controller problem

	s,s,let_1	s,s,let_2	s,e,let_1	s,e,let_2	e,s,let_1	e,s,let_2	e,e,let_1	e,e,let_2
a_1a_2	a_1a_2	a_1a_2	a_1w_2	a_1w_2	w_1a_2	w_1a_2	w_1w_2	w_1w_2
a_1w_2	–	–	–	a_1t_2	–	–	–	w_1t_2
w_1a_2	–	–	–	–	t_1a_2	–	t_1w_2	–
w_1w_2	–	–	–	–	–	–	t_1w_2	w_1t_2
a_1t_2	–	a_1t_2	–	a_1a_2	–	w_1t_2	–	w_1a_2
t_1a_2	t_1a_2	–	t_1w_2	–	a_1a_2	–	a_1w_2	–
w_1t_2	–	–	–	–	t_1t_2	w_1t_2	t_1a_2	w_1a_2
t_1w_2	–	–	t_1w_2	t_1t_2	–	–	a_1w_2	a_1t_2
t_1t_2	t_1t_2	t_1t_2	t_1t_2	t_1t_2	t_1t_2	t_1t_2	t_1t_2	t_1t_2

$$q_1q_2 \sim_{tr_1} q_1'q_2' \quad \text{iff} \quad q_1 = q_1'$$
$$q_1q_2 \sim_{tr_2} q_1'q_2' \quad \text{iff} \quad q_2 = q_2'$$

$\sim_c$	a_1a_2	a_1w_2	w_1a_2	w_1w_2	a_1t_2	t_1a_2	w_1t_2	t_1w_2	t_1t_2
a_1a_2	+								
a_1w_2		+			+				
w_1a_2			+			+			
w_1w_2				+			+	+	+
a_1t_2		+			+				
t_1a_2			+			+			
w_1t_2				+			+	+	+
t_1w_2				+			+	+	+
t_1t_2				+			+	+	+

Figure 10. Transitions and observational accessibility for the system with two trains and a controller

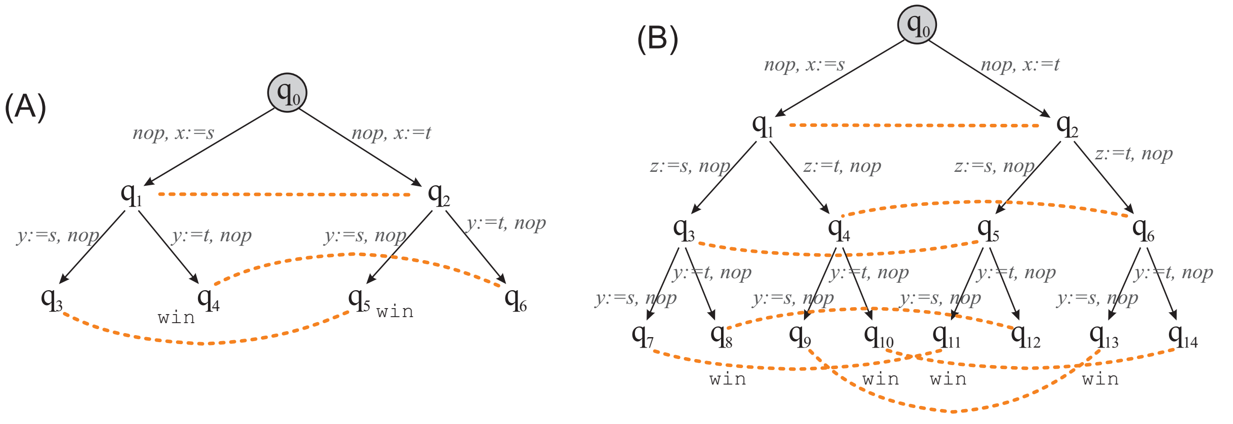


Figure 11. Models for two *IF* games: (A) $S[\forall x \exists y/x \; x \neq y]$; (B) $S[\forall x \exists z \exists y/x \; x \neq y]$. To help the reader, the nodes in which Falsifier makes a move are marked with grey circles; it is Verifier's turn at all the other nodes.

see which train is away – but nothing more. When one of the trains is away and the other is not, c has to light the green light for the latter.[9] The trains crash if they are in the tunnel at the same moment ($crash \equiv t1 \wedge t2$), so the controller should not let a train into the tunnel if the other train is inside. Unfortunately:

- c is not able to do so: $\neg\langle\langle c\rangle\rangle^{\bullet}_{Obs(c)}\Box\neg crash$, because it has to choose the same option in w_1t_2 and w_2t_1. Note that the controller would be able to keep the trains from crashing if it had perfect information: $\neg\langle\langle c\rangle\rangle^{\bullet}\Box\neg crash$, which shows exactly that insufficient epistemic capability of c is the source of this failure;

- on the other hand, a train (say, tr_1) can hint the right strategy (pass a signal) to the controller every time it is in the tunnel, so that there is no crash in the next moment: $\mathsf{A}\Box(t1 \rightarrow \langle\langle c\rangle\rangle^{\bullet}_{Obs(tr_1)}\bigcirc\neg crash)$;

- when tr_1 is out of the tunnel, then c can choose the strategy of letting tr_2 in if tr_2 is not away (and choosing let_1 else) to succeed in the next step: $\mathsf{A}\Box(\neg t1 \rightarrow \langle\langle c\rangle\rangle^{\bullet}_{Obs(c)}\bigcirc\neg crash)$;

- two last properties imply also that $\mathsf{A}\Box\langle\langle c\rangle\rangle^{\bullet}_{DO(c,tr_1)}\bigcirc\neg crash$: the controller can avoid the crash when he has enough communication from tr_1;

- however, $\neg\langle\langle c\rangle\rangle^{\bullet}_{DO(c,tr_1)}\Box\neg crash$, so a one-time communication is not enough;

- finally, c is not a very good controller for one more reason – it cannot detect a crash even if it occurs: $crash \rightarrow \neg Obs_c\, crash$.

[9]This is meant to impose fair access of the trains to the tunnel: note that when tr_i wants to enter the tunnel, it must be eventually allowed if only the other train does not stay in the tunnel for ever. Adding explicit fairness conditions, like in Fair ATL [4], would probably be a more elegant solution, but it goes far beyond the scope of the example and the paper.

Example 4.4. The last example refers to *IF* games, introduced by Hintikka and Sandu in [16], and investigated further in [6] from a game-theoretic perspective. The metaphor of mathematical proof as a game between Verifier V (who wants to show that the formula in question is true) and Falsifier F (who wants to demonstrate the opposite) is the starting point here. One agent takes turn at each quantifier: at $\exists x$, Verifier is free to assign x with any domain object he likes, while at $\forall x$ the value is chosen by Falsifier. *IF* games generalize the idea with their "slash notation": $\exists x/y$ means that V can choose a value for x, but at the same time he must forget everything he knew about the value of y (for ever). [6] suggests that such logic games can be given a proper game-theoretical treatment too, and uses dynamic-epistemic logic to reason about the players' knowledge, their powers etc. Obviously, ATOL can be used for the same purpose.

Let us consider two *IF* games from [6]: one for $\forall x \exists y/x \; x \neq y$, the other for $\forall x \exists z \exists y/x \; x \neq y$. The game trees for both games are shown in Figure 11. The arcs are labeled with j_V, j_F where j_V is the action of Verifier and j_F is the Falsifier's decision; nop stands for "no-operation" or "do-nothing" action. Dotted lines display V's observational accessibility links. F has perfect information in both games. It is assumed that the domain contains two objects: s and t. Atom win indicates the states in which the Verifier wins, i.e. the states in which he has been able to prove the formula in question.

We will use the trees as concurrent observational game structures to demonstrate interesting properties of the players with ATOL formulas.

- $S[\forall x \exists y/x \; x \neq y], q_0 \models \neg\langle\langle V \rangle\rangle^{\bullet}_{Obs(V)} \Diamond win$: Verifier has no uniform strategy to win this game;
- note that $S[\forall x \exists y/x \; x \neq y], q_0 \models \neg\langle\langle F \rangle\rangle^{\bullet}_{Obs(F)} \Box \neg win$: Falsifier has no power to prevent V from winning as well in the first game – in other words, the game is non-determined. Thus, the reason for V's failure lies in his insufficient epistemic abilities – in the second move, to be more specific: $S[\forall x \exists y/x \; x \neq y], q_0 \models \langle\langle V \rangle\rangle^{\bullet}_{Obs(V)} \bigcirc \langle\langle V \rangle\rangle^{\bullet}_{CO(\emptyset)} \Diamond win$;
- the vacuous quantifier in (B) does matter a lot: V can use it to store the actual value of x, so $S[\forall x \exists z \exists y/x \; x \neq y], q_0 \models \langle\langle V \rangle\rangle^{\bullet}_{Obs(V)} \Diamond win$!
- Verifier has a strategy that guarantees win (see above), but he will never be able to observe that he has actually won: $S[\forall x \exists z \exists y/x \; x \neq y], q_0 \models \neg\langle\langle V \rangle\rangle^{\bullet}_{Obs(V)} \Diamond Obs_V win$.

Giving a complete axiomatization for ATOL is beyond the scope of this paper. We only mention a few tautologies below.

Proposition 4.2. The following are valid ATOL properties:

- $\langle\langle A \rangle\rangle^{\bullet}_{Obs(\gamma)} \Phi \rightarrow Obs_\gamma \langle\langle A \rangle\rangle^{\bullet}_{Obs(\gamma)} \Phi$: if γ is able to identify A's strategy to bring about Φ, then he can see that A have such a strategy, too;
- more generally: $\langle\langle A \rangle\rangle^{\bullet}_{\Theta(\Gamma)} \Phi \rightarrow \Theta_\Gamma \langle\langle A \rangle\rangle^{\bullet}_{\Theta(\Gamma)} \Phi$;
- $\langle\langle A \rangle\rangle^{\bullet}_{\Theta(\Gamma)} \Phi \rightarrow \Theta_\Gamma \langle\langle A \rangle\rangle^{\bullet}_{CO(\emptyset)} \Phi$: if A have a strategy *de re* in any sense, then they have also a strategy *de dicto* in the same sense.
- having a strategy *de dicto* implies having a complete information strategy: $\langle\langle A \rangle\rangle^{\bullet}_{CO(\emptyset)} \Phi \rightarrow \langle\langle A \rangle\rangle^{\bullet} \Phi$.

5. ATEL-R*: Knowledge, Cooperation and Time with no Restraint

Real agents have finite memory and unless they can extend their capacity when necessary (hence making the memory finite, but unbounded), models with no recall can be used for them. However, even if we know that an agent has limited memory capabilities, we seldom know which observations he will actually decide to remember. Models with no recall exist for many problems, but they are often extremely large and must be constructed on the fly for every particular setting. Assigning agents with perfect recall can be a neat way to get rid of these inconveniences, although at the expense of making the agents remember (and accomplish) too much. Our language to talk about agents with recall – Alternating-time Temporal Epistemic Logic with Recall (ATEL-R*) – includes the following formulas:

- p, where p is an atomic proposition;
- $\neg\varphi$ or $\varphi \vee \psi$, where φ, ψ are ATEL-R* formulas;
- $\bigcirc\varphi$ or $\varphi\mathcal{U}\psi$, where φ, ψ are ATEL-R* formulas.
- $\mathcal{K}_A\ \varphi$, where $\mathcal{K}$ is any of the collective knowledge operators: C, E, D, A is a set of agents, and φ is an ATEL-R* formula;
- $\langle\langle A\rangle\rangle_{\mathcal{K}(\Gamma)}\ \varphi$, where A, Γ are sets of agents, $\mathcal{K} = C, E, D$, and φ is an ATEL-R* formula.

We would like to embed the observational logic ATOL, and modalities for strategies with complete information into ATEL-R* in a general way. Past time operators can be also useful in the context of perfect recall, so the following formulas are meant to be a part of ATEL-R* as well ($\Theta = CO, EO, DO$ and $\mathcal{K} = C, E, D$):

- $\Theta_A\ \varphi$;
- $\langle\langle A\rangle\rangle^{\bullet}_{\Theta(\Gamma)}\varphi$, $\langle\langle A\rangle\rangle^{\bullet}_{\mathcal{K}(\Gamma)}\varphi$, $\langle\langle A\rangle\rangle_{\Theta(\Gamma)}\varphi$;
- $\langle\langle A\rangle\rangle^{\bullet}\varphi$, $\langle\langle A\rangle\rangle\varphi$;
- $\bigcirc^{-1}\varphi$ ("previously φ") and $\varphi\,\mathcal{S}\,\psi$ ("φ since ψ") .

Several derived operators can be defined:

- $\varphi \wedge \psi \equiv \neg(\neg\varphi \vee \neg\psi)$ etc.;
- $K_a\varphi \equiv C_{\{a\}}\varphi$ and $\langle\langle A\rangle\rangle_{K(\gamma)}\Phi \equiv \langle\langle A\rangle\rangle_{C(\{\gamma\})}\varphi$;
- $Obs_a\varphi \equiv CO_{\{a\}}\varphi$ and $\langle\langle A\rangle\rangle^{\bullet}_{Obs(\gamma)}\Phi \equiv \langle\langle A\rangle\rangle^{\bullet}_{CO(\{\gamma\})}\Phi$;
- $\Diamond\varphi \equiv true\,\mathcal{U}\varphi$ and $\Box\varphi \equiv \neg\Diamond\neg\varphi$;
- $\Diamond^{-1}\varphi \equiv true\,\mathcal{S}\,\varphi$ and $\Box^{-1}\varphi \equiv \neg\Diamond^{-1}\neg\varphi$;
- $\mathsf{A}\varphi \equiv \langle\langle\emptyset\rangle\rangle_{C(\emptyset)}\varphi$ and $\mathsf{E}\varphi \equiv \neg\langle\langle\emptyset\rangle\rangle_{C(\emptyset)}\neg\varphi$.

5.1. Semantics for ATEL-R*

A few semantics have been proposed for CTL* with past time [15, 25]. The semantics we use for ATEL-R* is based on [25], where cumulative linear past is assumed: the history of the current situation increases with time and is never forgotten. In a similar way, we do not make the usual (unnecessary) distinction between state and path formulas here.

The knowledge accessibility relation for agent a is defined as before: $\lambda \approx^{\mathcal{K}}_{A} \lambda'$ iff $\lambda[i] \sim^{\mathcal{K}}_{A} \lambda'[i]$ for all i. Again, $\xi[i]$, $\xi_{|i}$, and ξ^i denote the ith position, first $i+1$ positions, and the ith suffix of ξ respectively. The semantics for ATEL-R*, proposed below, exploits also function $out^{*}_{\mathcal{K}(\Gamma)}(\lambda, F_A)$ which returns the set of computations that are possible from the viewpoint of group Γ (with respect to knowledge operator $\mathcal{K}$) in situation λ (i.e. after history λ took place):

$$out^{*}_{\mathcal{K}(\Gamma)}(\lambda, F_A) = \{\Lambda \mid \Lambda_{|n} \approx^{\mathcal{K}}_{\Gamma} \lambda \text{ and } \Lambda^n \text{ is consistent with } F_A \text{, where } n \text{ is the length of } \lambda\}.$$

Definition 5.1. The semantics of ATEL-R* is defined with the following rules:

$\Lambda, n \models p$	iff	$p \in \pi(\Lambda[n])$
$\Lambda, n \models \neg\varphi$	iff	$\Lambda, n \not\models \varphi$
$\Lambda, n \models \varphi \vee \psi$	iff	$\Lambda, n \models \varphi$ or $\Lambda, n \models \psi$
$\Lambda, n \models \bigcirc\varphi$	iff	$\Lambda, n+1 \models \varphi$
$\Lambda, n \models \varphi\mathcal{U}\psi$	iff	there is a $k \geq n$ such that $\Lambda, k \models \psi$ and $\Lambda, i \models \varphi$ for all $n \leq i < k$
$\Lambda, n \models \mathcal{K}_A\, \varphi$	iff	for every Λ' such that $\Lambda'_{\vert n} \approx^{\mathcal{K}}_{A} \Lambda_{\vert n}$ we have $\Lambda', n \models \varphi$ (where $\mathcal{K}$ can be any of the collective knowledge operators: C, E, D)
$\Lambda, n \models \langle\langle A \rangle\rangle_{\mathcal{K}(\Gamma)}\, \varphi$	iff	there exists a collective uniform strategy F_A such that for every $\Lambda' \in out^{*}_{\mathcal{K}(\Gamma)}(\Lambda_{\vert n}, F_A)$ we have $\Lambda', n \models \varphi$.

We believe that adding past time operators to ATEL-R* does not change its expressive power – the same way as CTL*+Past has been proven equivalent to CTL* [15, 25]. However, explicit past tense constructs in the language enable expressing history-oriented properties in a natural and easy way.

Definition 5.2. Semantics of past tense operators can be defined as follows:

$\Lambda, n \models \bigcirc^{-1}\varphi$	iff	$n > 0$ and $\Lambda, n-1 \models \varphi$
$\Lambda, n \models \varphi\,\mathcal{S}\,\psi$	iff	there is a $k \leq n$ such that $\Lambda, k \models \psi$ and $\Lambda, i \models \varphi$ for all $k < i \leq n$.

Example 5.1. Consider the trains and controller from Example 4.3. The trains can never enter the tunnel at the same moment, so $\mathsf{A}\Box(crash \rightarrow \bigcirc^{-1}(t_1 \vee t_2))$, i.e. if there is a crash, then a train must have already been in the tunnel in the previous moment. The formula is equivalent to $\mathsf{A}\Box\neg(\neg(t_1 \vee t_2) \wedge \bigcirc crash)$ when we consider both formulas from the perspective of the initial point of the computation: it cannot happen that no train is in the tunnel and in the next state the trains crash.[10]

Example 5.2. Another useful past time formula is $\neg\bigcirc^{-1}true$, that specifies the starting point in computation. For instance, we may want to require that no train is in the tunnel at the beginning: $\neg\bigcirc^{-1}true \rightarrow \neg t_1 \wedge \neg t_2$, which is initially equivalent to $\neg t_1 \wedge \neg t_2$, but states the fact explicitly and holds for all points in all computations. Also, tautology $\mathsf{A}\Box\Diamond^{-1}\neg\bigcirc^{-1}true$ makes it clear that we deal with finite past in ATEL-R*.

[10] For a precise definition and more detailed discussion of *initial equivalence*, consult for instance [25].

5.2. Knowledge vs. Observations

It can be interesting to reason about observations in ATEL-R*, too. We can embed ATOL in ATEL-R* in the following way:

Definition 5.3. For all $\Theta = CO, EO$ or DO:

$\Lambda, n \vDash \Theta_A\, \varphi$ iff for every Λ', n' such that $\Lambda'[n'] \sim_A^{\Theta} \Lambda[n]$ we have $\Lambda', n' \vDash \varphi$

$\Lambda, n \vDash \langle\langle A \rangle\rangle^{\bullet}_{\Theta(\Gamma)} \varphi$ iff there is a uniform strategy with no recall V_A such that for every Λ', n', for which $\Lambda'[n'] \sim_{\Gamma}^{\Theta} \Lambda[n]$ and Λ' is consistent with V_A, we have $\Lambda', n' \vDash \varphi$.

Operators for memoryless strategies, identified by agents with recall ($\langle\langle A \rangle\rangle^{\bullet}_{\mathcal{K}(\Gamma)}$, $\mathcal{K} = C, E, D$) and vice versa ($\langle\langle A \rangle\rangle_{\Theta(\Gamma)}$, $\Theta = CO, EO, DO$) can also be added in a straightforward way.

The explicit distinction between knowledge and observations can help to clarify a few things. The first one is more philosophical: an agent knows what he can see *plus* what he can remember to have seen. Or – more precisely – knowledge is what we can deduce from our present and past observations, provided we are given sufficient observational abilities (in the ontological sense, i.e. we can name what we see).

Proposition 5.1. Suppose our language is rich enough to identify separate states, i.e. the set of propositions Π includes a proposition α_q for every state $q \in Q$, such that α_q is true only in q (since the set of states is always finite, we can always add such propositions to Π for each particular model). Then for every formula φ there exist formulas $\varphi'_1, \varphi''_1, \ldots, \varphi'_n, \varphi''_n$, such that $\varphi'_1, \ldots, \varphi'_n$ contain no epistemic operators with recall (K, C, E, D), and $\varphi'_i \wedge \varphi''_i \rightarrow \varphi$ for every i, and:

$$\bigvee_{i=1..n} K_a\varphi \equiv (Obs_a\varphi'_i \wedge \bigcirc^{-1} K_a \bigcirc \varphi''_i) \vee (\neg\bigcirc^{-1} true \wedge Obs_a\varphi).$$

This implies that, in every situation, $K_a\varphi$ can be rewritten to some formula $Obs_a\varphi'_i \wedge \bigcirc^{-1}K_a\bigcirc\varphi''_i$ unless we are at the beginning of a run – then it should be rewritten to $Obs_a\varphi$.

Proof:
Consider formulas $\varphi'_i \equiv \neg Obs_a\neg\alpha_{q_i}$ and $\varphi''_i \equiv \neg Obs_a\neg\alpha_{q_i} \rightarrow \varphi$, one pair for each state $q_i \in Q$. Let Λ, n be a computation and a position in it, and $\Lambda[n] = q_j$ current state of the computation. Suppose that $\Lambda, n \models K_a\varphi$; then for every Λ' such that $\Lambda'_{|n} \approx_a \Lambda_{|n}$, we have that $\Lambda', n \models \varphi$. Note that $\neg Obs_a\neg\alpha_{q_j}$ is true exactly in the states belief-accessible for a from q_j, so $\Lambda, n \models Obs_a(\neg Obs_a\neg\alpha_{q_j})$. Now, $\Lambda'_{|n-1} \approx_a \Lambda_{|n-1}$ and $\Lambda', n \models \neg Obs_a\neg\alpha_{q_j}$ only if $\Lambda'_{|n} \approx_a \Lambda_{|n}$, so $\Lambda', n-1 \models \bigcirc\varphi$ for all Λ' such that $\Lambda'_{|n-1} \approx_a \Lambda_{|n-1}$. In consequence, $\Lambda', n-1 \models K_a\bigcirc(\neg Obs_a\neg\alpha_{q_j} \rightarrow \varphi)$ and hence $\Lambda, n \models \bigcirc^{-1}K_a\bigcirc(\neg Obs_a\neg\alpha_{q_j} \rightarrow \varphi)$. Finally, $\neg Obs_a\neg\alpha_{q_i}$ and $\neg Obs_a\neg\alpha_{q_i} \rightarrow \varphi$ obviously imply φ.

The proof that $\Lambda, n \models \bigcirc^{-1}K_a\bigcirc(\neg Obs_a\neg\alpha_{q_j} \rightarrow \varphi)$ and $\Lambda, n \models Obs_a(\neg Obs_a\neg\alpha_{q_j})$ imply $\Lambda, n \models K_a\varphi$ is analogous. □

Example 5.3. The above proposition can be illustrated with the systems in Figure 12. Consider path q_2q_6 for example. The agent must have known in q_2 that he was in q_1 or q_2 and therefore in the next step

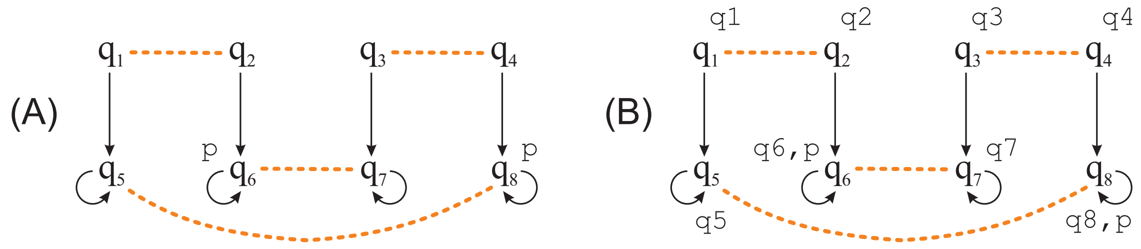


Figure 12. Knowledge vs. observations: with and without the vocabulary

he can be in either q_5 or q_6. Now, in q_6 he can observe that the current state is q_6 or q_7, so it must be q_6 in which p holds. Note that the agent's ontology is too poor in system (A): he cannot express with the available language the differences he can actually see. Sufficient vocabulary is provided in Figure 12(B): for instance, when q_6 is the current state, $K_a\, p$ can be always rewritten as

$$Obs_a \neg Obs_a \neg q6 \wedge \bigcirc^{-1} K_a \bigcirc (\neg Obs_a \neg q6 \rightarrow p)$$

and of course $\neg Obs_a \neg q6 \wedge (\neg Obs_a \neg q6 \rightarrow p) \rightarrow p$.

5.3. Complete Information vs. Uniform Strategies

Let $\langle\langle A \rangle\rangle \Phi$ denote that A have a complete information strategy to enforce Φ - like in ATL and original ATEL*. Relationship analogous to Proposition 5.1 can be shown between the incomplete and complete information cooperation modalities. This one is not past-, but future-oriented, however.

Proposition 5.2. $\langle\langle A \rangle\rangle \Phi \equiv \langle\langle A \rangle\rangle_{C(\emptyset)} \bigcirc \langle\langle A \rangle\rangle \bigcirc^{-1} \Phi$. In other words, having a complete information strategy is equivalent to having a uniform strategy that can be hinted at every step by an omniscient observer.

A similar property can be shown for agents with no recall:

Proposition 5.3. $\langle\langle A \rangle\rangle^{\bullet} \Phi \equiv \langle\langle A \rangle\rangle^{\bullet}_{CO(\emptyset)} \bigcirc \langle\langle A \rangle\rangle^{\bullet} \bigcirc^{-1} \Phi$.

5.4. More Examples

Several further examples for ATEL-R* are presented below.

Example 5.4. For the variable client/server system from Examples 2.3 and 4.1, recall of the past adds nothing to the agents' powers:

- $\langle\langle s \rangle\rangle_{K(s)} \varphi \rightarrow \langle\langle s \rangle\rangle^{\bullet}_{Obs(s)} \varphi$,
- $\langle\langle c \rangle\rangle_{K(c)} \varphi \rightarrow \langle\langle c \rangle\rangle^{\bullet}_{Obs(c)} \varphi$,
- $K_s \varphi \rightarrow Obs_s \varphi$ etc.

This is because each state can be reached from all the other ones in a single step. Thus, knowledge of the previous positions in the game does not allow for any elimination of possible alternatives. Obviously, in a realistic setting, the agents would remember not only their local states from the past, but also the decisions they made – and that would improve the client's epistemic capacity.

Example 5.5. Consider a model for the variable client and server, extended in a similar way as in Example 4.2 (in which every player remembers his last decision). For this system, the client starts to have complete knowledge of the situation as soon as x is assigned the value of 2:

- $\mathsf{A}\Box(x = 2 \rightarrow \mathsf{A}\Box(K_c(x = 0) \vee K_c(x = 1) \vee K_c(x = 2)))$;
- note that still $\mathsf{A}\Box(\neg Obs_c(x = 0) \wedge \neg Obs_c(x = 1))$.

On the other hand, the server gains only some knowledge of the past. If he has been rejecting the claims all the time, for instance, he knows that at the beginning the value of x must have been the same as now:

- $\Box^{-1} rej \rightarrow K_s(x = 0 \rightarrow \Box^{-1}(\neg\bigcirc^{-1} true \rightarrow x = 0))$ etc.

Example 5.6. Some properties of the train controller from Example 4.3 can be analyzed through formulas of ATEL-R*:

- $t_i \rightarrow K_c \bigcirc^{-1} \neg a_i$: every time a train is in the tunnel, c knows at least that in the previous moment it was not away;
- the controller is still unable to accomplish its mission: $\neg\langle\langle c\rangle\rangle_{K(c)} \Box\neg crash$, but...
- $a_1 \wedge a_2 \rightarrow \langle\langle c\rangle\rangle_{K(c)} (\Box\neg(a_1 \wedge a_2 \wedge \bigcirc(w_1 \wedge w_2)) \rightarrow \Box\neg crash)$. Suppose the trains never enter "the waiting zone" simultaneously and both are initially away – then c can finally keep them from crashing. The strategy is to immediately grant the green light to the first train that enters the zone, and keep it until the train is away again – then switch it to the other one if it has already entered the zone, and so on;
- also, if c is allowed to remember his last decision (i.e. the model is modified in the same way as in previous examples), then c knows who is in the tunnel: $\mathsf{A}\Box(K_c t_i \vee K_c \neg t_i)$ in the new model. In consequence, c can keep the other train waiting and avoid crash as well.

Example 5.7. Consider *IF* games again (see Example 4.4). An interesting variation on the theme can be to allow that a game is played repeatedly a (possibly infinite) number of times. For instance, we can have formula Υ_1 defined as a fixed point: $\Upsilon_1 \equiv \forall x \exists y/x\ (x \neq y \vee \Upsilon_1)$, which means that the game of $\forall x \exists y/x\ x \neq y$ should be played until Verifier wins. The structure of this game is presented in Figure 13.

- Verifier still cannot be guaranteed that he eventually wins: $S[\Upsilon_1], q_0 \ldots, 0 \models \neg\langle\langle V\rangle\rangle_{K(V)} \Diamond win$;
- this time around, however, V's success is much more likely: for each strategy of his, he fails on one path out of infinitely many possible (and Falsifier has to make up his mind *before* V). Intuitively, the probability of eventually bringing about `win` is 1, yet we do not see how this issue can be expressed in ATEL-R* or ATOL at present;

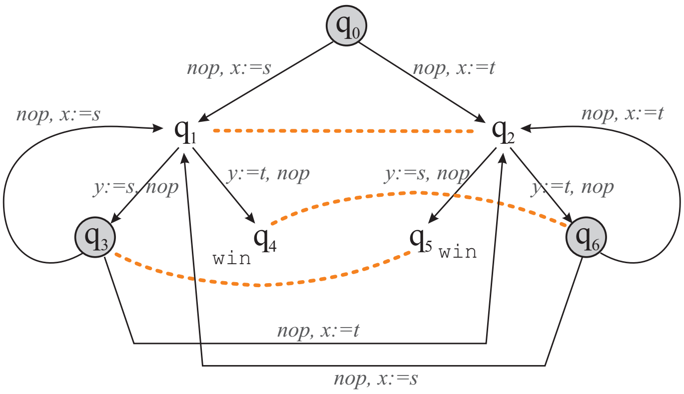


Figure 13. Game structure $S[\Upsilon_1]$ for game $\Upsilon_1 \equiv \forall x \exists y/x\ (x \neq y \vee \Upsilon_1)$

- note that in an analogous model for $\forall x \forall z \exists y/x\ x \neq y$ we have $S[\forall x \forall z \exists y/x\ x \neq y], q_0 \ldots, 0 \models \langle\langle V \rangle\rangle_{K(V)} \Diamond win$, yet this is only because the semantics of ATEL-R* does not treat $\sim_V$ as the epistemic accessibility relation, but rather as a basis from which the relation is generated. Hence, it allows V to remember the value of x anyway – which shows that $S[\forall x \forall z \exists y/x\ x \neq y]$ is not a suitable ATEL-R* model for the formula (although it is still an appropriate ATOL model);

- in order to encode the new game in ATEL-R*, we should split Verifier into two separate players V_1 and V_2. V_1 makes the move at the first and second steps and has a complete information about the state of the environment; V_2 does not see the Falsifier's choice for x at all. What we should ask about then is: $\langle\langle V_1 \rangle\rangle_{K(V_1)} \bigcirc \bigcirc \langle\langle V_2 \rangle\rangle_{K(V_2)} \Diamond win$, which naturally does not hold;

- the above shows that ATOL is much closer to the spirit of *IF* games than ATEL-R*. Why should we care about ATEL-R* for modeling *IF* games at all? Well, consider game $\forall x \Upsilon_2$, where $\Upsilon_2 \equiv \exists y/x\ (x \neq y \vee \Upsilon_2)$; the structure of the game is shown in Figure 14. In ATEL-R*, Verifier has a simple winning strategy: first try $y := s$, and the next time $y := t$, and he is bound to hit the appropriate value – hence, $S[\Upsilon_2], q_0 \ldots, 0 \models \langle\langle V \rangle\rangle_{K(V)} \Diamond win$. At the same time, V has no memoryless strategy: $S[\Upsilon_1], q_0 \ldots, 0 \models \neg \langle\langle V \rangle\rangle^{\bullet}_{Obs(V)} \Diamond win$, because he loses the knowledge what he did with y last time every time he uses y again. In a sense, $\langle\langle V \rangle\rangle_{K(V)}$ is closer to the way variables are treated in mathematical logic than $\langle\langle V \rangle\rangle^{\bullet}_{Obs(V)}$: in $\exists y \exists y\ \varphi$ both quantifiers refer to *different* variables that have the same name only incidentally.

Proposition 5.4. Finally, the following formulas are examples of ATEL-R* tautologies:

- $\langle\langle A \rangle\rangle_{\mathcal{K}(\Gamma)} \Phi \rightarrow \mathcal{K}_\Gamma \langle\langle A \rangle\rangle_{\mathcal{K}(\Gamma)} \Phi$: if Γ are able to identify A's strategy to bring about Φ, then they also know that A have such a strategy;

- $\langle\langle A \rangle\rangle_{\mathcal{K}(\Gamma)} \Phi \rightarrow \mathcal{K}_\Gamma \langle\langle A \rangle\rangle_{CO(\emptyset)} \Phi$: if A have a strategy *de re*, then they have a strategy *de dicto*;

- having a uniform strategy implies having a complete information strategy: $\langle\langle A \rangle\rangle_{CO(\emptyset)} \Phi \rightarrow \langle\langle A \rangle\rangle \Phi$;

- $\langle\langle A \rangle\rangle^{\bullet}_{\mathcal{K}(\Gamma)} \Phi \rightarrow \langle\langle A \rangle\rangle_{\mathcal{K}(\Gamma)} \Phi$: memoryless strategies are special cases of strategies with recall.

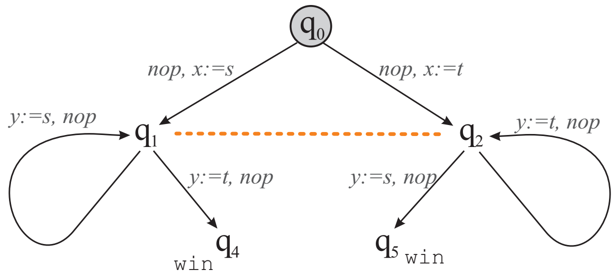


Figure 14. Game structure $S[\forall x \Upsilon_2]$

5.5. Expressivity and Complexity of ATEL-R* and its Subsets

ATEL-R* logic, as defined here, subsumes ATL* and the original ATEL*, as well as Schobbens's ATL_{ir}*, ATL_{iR}* and ATL_{Ir}* logics from [31]. One interesting issue about ATL*, ATL_{ir}*, ATL_{iR}* and ATL_{Ir}* is that they do not seem to be expressible by each other on the language level.[11] This is why we decided to include separate operators for each relevant perspective to epistemic and strategic abilities of agents.

The complexity results for model checking of "vanilla" ATEL-R are rather discouraging. Even parts of it are already intractable: the same problem for ATL_{ir} is NP-hard [31], and model checking of ATL_{iR} (based on cooperation modalities for incomplete information and perfect recall) is generally believed to be undecidable [31]. We would like to stimulate a systematic investigation of the issue by extending the notation from [10]. Let $B\ (P_1, P_2, \ldots \mid T_1, T_2, \ldots \mid M_1, M_2, \ldots)$ be the branching time logic with path quantifiers $P_1, P_2, \ldots$, temporal operators $T_1, T_2, \ldots$ and other modalities $M_1, M_2, \ldots$. Every temporal operator must have a path quantifier as its immediate predecessor (like in CTL). Then:

1. $B\ (E \mid \bigcirc, \Box, \mathcal{U} \mid -)$ is CTL;
2. $B\ (\langle\langle A \rangle\rangle \mid \bigcirc, \Box, \mathcal{U} \mid -)$ is ATL;
3. $B\ (\langle\langle A \rangle\rangle \mid \bigcirc, \Box, \mathcal{U} \mid CO_A, EO_A, DO_A)$ is the original version of ATEL from [17];
4. $B\ (\langle\langle A \rangle\rangle_{CO(\emptyset)} \mid \bigcirc, \Box, \mathcal{U} \mid CO_A, EO_A, DO_A)$ is the ATEL version from [22];
5. $B\ (\langle\langle A \rangle\rangle^{\bullet}_{\Theta(\Gamma)} \mid \bigcirc, \Box, \mathcal{U} \mid CO_A, EO_A, DO_A)$ is ATOL (Section 4);
6. $B\ (\langle\langle A \rangle\rangle_{\mathcal{K}(\Gamma)}, \langle\langle A \rangle\rangle^{\bullet}_{\mathcal{K}(\Gamma)}, \langle\langle A \rangle\rangle, \langle\langle A \rangle\rangle^{\bullet} \mid \bigcirc, \Box, \mathcal{U}, \bigcirc^{-1}, \Box^{-1}, \mathcal{S} \mid C_A, E_A, D_A, CO_A, EO_A, DO_A)$ is "vanilla" ATEL-R.

The model checking problem can be solved in polynomial time for (3). On the other hand, the same problem for (5) is NP-hard and $\Delta_2 P$-easy (Proposition 4.1). It turns out that the authors of the original ATEL proposed the largest tractable member of the family to date. Whether anything relevant can be added to it seems an important question.

[11]Except for ATL and ATL_{Ir} – but without the star! – which are equivalent

6. Conclusions

We have briefly re-introduced Alternating-time Temporal Logic and discussed two kinds of models for the language (concurrent game structures and alternating transition systems), trying to stress that – when one wants to reason about knowledge in such systems – it is important to distinguish the computational structure from the behavioral structure, and to decide in what way the first one unravels into the latter. We argue that the initial approach to Alternating-time Temporal Epistemic Logic [17] offered too weak a notion of a strategy. In order to say that agent a can enforce a property φ, it was required that there existed a sequence of a's actions at the end of which φ held – whether he had knowledge to recognize the sequence was not taken into account. Moreover, even the requirement that the agent's strategy must be uniform proved too weak: it would still enable plans in which the agent was allowed to "guess" the opening action. We suggest that it is not enough that the agent knows that some strategy will help him out; it is more appropriate to require that the agent can identify the winning strategy itself. In other words, the agent should be required to have a strategy *de re* rather than *de dicto*. Under such a constraint, the agent "knows how to play".

This is still not enough to give the meaning of a cooperation modality for coalitional planning under uncertainty. Even if a group of agents can collectively identify a winning strategy, they are prone to fail in case there are other competing strategies as well. Thus, we propose several different operators instead of one to distinguish subtle cases here.

The assumption that agents can use the complete history to make their subsequent decisions is also investigated in this paper. Two paradigms are studied here in consequence. First, agents can be assumed to have no (or only limited) memory. In this case, they make their decisions only on the basis of what they can *observe* (albeit in the broadest possible sense of the word); a language for ATL + Observations, dubbed ATOL, is proposed for specification of such agents in this paper. The other paradigm is formalized via a richer system, called Alternating-time Temporal Epistemic Logic with Recall (ATEL-R*). We believe that both approaches can be equally interesting and useful.

This paper is only a step towards clarifying epistemic issues in ATL, and it leaves many questions open. For instance, although only recently a complete axiomatization for ATL has been given [13], this is still unexplored area for ATEL and ATOL. Also, more non-trivial examples of game-like scenario's should be looked for in which a logic of knowledge and time may help to reveal interesting properties, and which are good cases for automated planning via model checking.

Acknowledgments The authors would like to thank Marek Bednarczyk, Valentin Goranko, Geert Jonker, Mark Ryan, Pierre Yves Schobbens, Rafal Somla and Mike Wooldridge, for their valuable remarks and discussions.

References

[1] Alur, R., de Alfaro, L., Henzinger, T. A., Krishnan, S., Mang, F., Qadeer, S., Rajamani, S., Tasiran, S.: *MOCHA User Manual,* Technical report, University of Berkeley, 2000.

[2] Alur, R., Henzinger, T. A., Kupferman, O.: Alternating-Time Temporal Logic, *Proceedings of the 38th Annual Symposium on Foundations of Computer Science (FOCS),* IEEE Computer Society Press, 1997, Available at `citeseer.ist.psu.edu/170985.html`.

[3] Alur, R., Henzinger, T. A., Kupferman, O.: Alternating-Time Temporal Logic, *Lecture Notes in Computer Science*, **1536**, 1998, 23–60, Available at `citeseer.ist.psu.edu/174802.html`.

[4] Alur, R., Henzinger, T. A., Kupferman, O.: Alternating-Time Temporal Logic, *Journal of the ACM*, **49**, 2002, 672–713, Updated, improved, and extended text. Available at `www.cis.upenn.edu/ alur/Jacm02.pdf`.

[5] van Benthem, J.: Games in Dynamic Epistemic Logic, *Bulletin of Economic Research*, **53**(4), 2001, 219–248, Proceedings of LOFT-4.

[6] van Benthem, J.: The Epistemic Logic of *IF* Games (Hintikka Self-Applied), 2002, Under preparation. To appear in the Hintikka Volume, *Library of Living Philosophers*. Available at `http://staff.science.uva.nl/~johan/H-H.pdf`.

[7] van Ditmarsch, H.: *Knowledge Games*, Ph.D. Thesis, University of Groningen, 2000.

[8] Dunin-Keplicz, B., Verbrugge, R.: A Tuning Machine for Collective Commitments, *Proceedings of Formal Approaches to Multi-Agent Systems (FAMAS 2003)* (B. Dunin-Keplicz, R. Verbrugge, Eds.), 2003.

[9] Emerson, E. A.: Temporal and Modal Logic, in: *Handbook of Theoretical Computer Science* (J. van Leeuwen, Ed.), vol. B, Elsevier Science Publishers, 1990, 995–1072.

[10] Emerson, E. A., Halpern, J. Y.: "Sometimes" and "Not Never" Revisited: On Branching versus Linear Time Temporal Logic, *Journal of the ACM*, **33**(1), 1986, 151–178.

[11] Fagin, R., Halpern, J. Y., Moses, Y., Vardi, M. Y.: *Reasoning about Knowledge*, MIT Press: Cambridge, MA, 1995.

[12] Goranko, V.: Coalition Games and Alternating Temporal Logics, *Proc. of the 8th Conference on Theoretical Aspects of Rationality and Knowledge (TARK VIII)* (J. van Benthem, Ed.), Morgan Kaufmann, 2001, Corrected version. Available at `http://general.rau.ac.za/maths/goranko/papers/gltl31.pdf`.

[13] Goranko, V., van Drimmelen, G.: Complete Axiomatization and Decidability of the Alternating-time Temporal Logic, 2003, Submitted.

[14] Goranko, V., Jamroga, W.: Comparing Semantics of Logics for Multi-agent Systems, *Synthese*, 2004, **139**(2), pp. 241–280.

[15] Hafer, T., Thomas, W.: Computation tree logic CTL* and Path Quantifiers in the Monadic Theory of the Binary Tree, in: *Proceedings of 14th ICALP, Karlsruhe* (F. Baader, G. Brewka, T. Eiter, Eds.), vol. 267 of *Lecture Notes in Computer Science*, Springer, 1987, 269–279.

[16] Hintikka, J., Sandu, G.: Game-Theoretical Semantics, in: *Handbook of Logic and Language* (J. van Benthem, A. ter Meulen, Eds.), Elsevier: Amsterdam, 1997, 361–410.

[17] van der Hoek, W., Wooldridge, M.: Tractable Multiagent Planning for Epistemic Goals, *Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-02)* (C. Castelfranchi, W. Johnson, Eds.), ACM Press, New York, 2002.

[18] van der Hoek, W., Wooldridge, M.: Cooperation, Knowledge and Time: Alternating-time Temporal Epistemic Logic and its Applications, *Studia Logica*, **75**(1), 2003, 125–157.

[19] van der Hoek, W., Wooldridge, M.: Model Checking Cooperation, Knowledge and Time – A Case Study, *Research in Economics*, **57**(3), 2003, 235–265.

[20] van der Hoek, W., Wooldridge, M.: Towards a Logic of Rational Agency, *Logic Journal of the IGPL*, **11**(2), 2003, 135–160.

[21] Huth, M., Ryan, M.: *Logic in Computer Science: Modeling and reasoning about systems*, Cambridge University Press, 2000.

[22] Jamroga, W.: Some Remarks on Alternating Temporal Epistemic Logic, *Proceedings of Formal Approaches to Multi-Agent Systems (FAMAS 2003)* (B. Dunin-Keplicz, R. Verbrugge, Eds.), 2003.

[23] Jonker, G.: Feasible Strategies in Alternating-time Temporal Epistemic Logic, 2003, Master thesis.

[24] Laroussinie, F.: About the expressive power of CTL combinators, *Information Processing Letters*, **54**(6), 1995, 343–345.

[25] Laroussinie, F., Schnoebelen, P.: A hierarchy of temporal logics with past, *Theoretical Computer Science*, **148**(2), 1995, 303–324.

[26] Moore, R.: A Formal Theory of Knowledge and Action, in: *Formal Theories of the Commonsense World* (J. Hobbs, R. Moore, Eds.), Ablex Publishing Corp., 1985.

[27] Morgenstern, L.: Knowledge and the Frame Problem, *International Journal of Expert Systems*, **3**(4), 1991.

[28] von Neumann, J., Morgenstern, O.: *Theory of Games and Economic Behaviour*, Princeton University Press: Princeton, NJ, 1944.

[29] Quine, W.: Quantifiers and Propositional Attitudes, *Journal of Philosophy*, **53**, 1956, 177–187.

[30] Schnoebelen, P.: The Complexity of Temporal Model Checking, *Advances in Modal Logics, Proceedings of AiML 2002*, World Scientific, 2003.

[31] Schobbens, P. Y.: Alternating-Time Logic with Imperfect Recall, *Electronic Notes in Theoretical Computer Science*, **85**(2), 2004.

[32] Wooldridge, M.: *Reasoning about Rational Agents*, MIT Press : Cambridge, Mass, 2000.