Reasoning about Strategies under Partial Observability and Fairness Constraints

Simon Busard, Charles Pecheur*
ICTEAM Institute,
Université catholique de Louvain,
Louvain-la-Neuve, Belgium
simon.busard@uclouvain.be
charles.pecheur@uclouvain.be

Hongyang Qu
Dept. of Computer Science,
University of Oxford,
Oxford, United Kingdom
Hongyang.Qu@cs.ox.ac.uk

Franco Raimondi
Dept. of Computer Science,
Middlesex University,
London, United Kingdom
f.raimondi@mdx.ac.uk

A number of extensions exist for Alternating-time Temporal Logic; some of these mix strategies and partial observability but, to the best of our knowledge, no work provides a unified framework for strategies, partial observability and fairness constraints. In this paper we propose $ATLK^F_{po}$, a logic mixing strategies under partial observability and epistemic properties of agents in a system with fairness constraints on states, and we provide a model checking algorithm for it.

1 Introduction

A number of extensions exist for Alternating-time Temporal Logic; starting from [7], partial observability has been investigated by many authors, see for instance [8] and references therein. But, to the best of our knowledge, no work provides a unified framework for strategies, partial observability and fairness constraints. For example, Jamroga and van der Hoek proposed, among other logics, ATOL, mixing partial observability with strategies of agents [10]. Along the same lines, Schobbens studied ATL_{ir}[14], seen as the minimal ATL-based logic for strategies under partial observability [9]. On the other hand, some efforts have been made on bringing fairness to ATL. For instance the work of Alur et al. [1], or the work of Klüppelholz and Baier [11] introduce the notion of fairness constraints on actions, asking for an infinitely often enabled action to be taken infinitely often. For temporal and epistemic logics, however, fairness conditions are normally provided on *states*. Furthermore, it has been shown that (weak, strong or unconditional) fairness constraints on actions, can be reduced to (weak, strong or unconditional, respectively) fairness constraints on states (see [2], for instance). In this paper we propose $ATLK^F_{po}$, a logic mixing strategies under partial observability and epistemic properties of agents in a system with unconditional fairness constraints *on states*, and we provide a model checking algorithm for it.

To motivate the need for fairness constraints in ATL under partial observability, consider the simple card game example in [10]. The game is played between a player and a dealer. It uses three cards, A, K and Q; A wins over K, K wins over Q and Q wins over A. First, the dealer gives one card to the player, keeps one and leaves the last one on table. Then the player can keep his card or swap it with the one on the table. The player wins if his card wins over the dealer's card. Under ATL_{ir} semantics, the player cannot win the game: he cannot distinguish between, for example, $< A,K >$ and $< A,Q >$ (where $< a,b >$ means "player has card a, dealer has card b") and thus has to make the same action in both states, with a different result in each case. Consider now a variation of this game: the game does not terminate after the first round. Instead, if the player does not win, cards are redistributed. In this case, too, the player cannot win the game: for instance, he will have to choose between keeping or swapping cards in $< A,K >$ and $< A,Q >$, so he won't be able to enforce a win because the dealer (that chooses the given cards) can be

*This work is supported by the European Fund for Regional Development and by the Walloon Region.

Submitted to:
SR 2013

unfair and always give the losing pair. But if we add one fairness constraint per intermediate state—i.e. the states in which the player has to choose between swapping or keeping—the player has a strategy to finally win the game. In this case, we only consider paths along which all fairness constraints are met infinitely often: this situation corresponds to a fair dealer, giving the cards randomly. The player can thus finally win because $< A, K >$ will eventually happen—even if he cannot distinguish it from $< A, Q >$—, so he knows a strategy to win at least a round: keeping his card.

Another example of application of fairness constraints in ATL is Multi-Agent Programs [5]. These programs are composed of interleaved agent programs and fairness constraints are used to avoid unfair interleaving. Dastani and Jamroga express fairness as formulae of the logic ATL* [5]; in this paper, instead, we deal only with ATL and therefore fairness constraints cannot be expressed as formulae of the logic. The situation is similar to the case of LTL versus CTL model checking: in the first case model checking fairness is reduced to model checking a more complex formula using the same verification algorithms; in the second case fairness is incorporated into bespoke verification algorithms. In our work we chose ATL over ATL* because of complexity considerations (see Section 3).

The rest of the paper is structured as follows: Section 2 presents the syntax and semantics of $ATLK^F_{po}$ and Section 3 presents two model checking algorithms for the logic. Finally, Section 4 summarizes the contribution and draws some future work.

2 Syntax and Semantics

This section presents the syntax and semantics of $ATLK^F_{po}$, an extension of ATL with partial observability under fairness constraints on states. An extension with full observability under the same fairness constraints $ATLK^F_{fo}$ is also presented because the model checking algorithm for $ATLK^F_{po}$ relies on the one for $ATLK^F_{fo}$.

Syntax Both logics share the same syntax, composed of the standard Boolean connectors ($\vee$, $\wedge$, $\neg$, etc.), CTL operators (EX, EU, EG, etc.) [4], knowledge operators (K_{ag}, E_Γ, D_Γ, C_Γ) [6] and strategic operators ($\langle\Gamma\rangle X$, $\langle\Gamma\rangle G$, $\langle\Gamma\rangle U$, $\langle\Gamma\rangle W$ and their $[\Gamma]$ counterparts) [1].

Models and notation $ATLK^F_{fo}$ and $ATLK^F_{po}$ formulae are interpreted over models $M = \langle Ag, S, Act, T, I, \{\sim_i\}, V, F\rangle$ where (1) Ag is a set of n agents; (2) $S = S_1 \times ... \times S_n$ is a set of global states, each of which is composed of n local states, one for each agent; (3) $Act = Act_1 \times ... \times Act_n$ is a set of joint actions, each of which is composed of n actions, one for each agent; (4) $T \subseteq S \times Act \times S$ is a transition relation between states in S and labelled with joint actions (we write $s \xrightarrow{a} s'$ if $(s,a,s') \in T$); (5) $I \subseteq S$ is the a set of initial states; (6) $\{\sim_i\}$ is a set of equivalence relations between states, and $\sim_i$ partitions the set of states in terms of knowledge of agent i—$s \sim_i s'$ iff $s_i = s'_i$, i.e two states are indistinguishable for agent i if they share the same local state for i; (7) $V : S \to 2^{AP}$ labels states with atomic propositions of AP; (8) $F \subseteq 2^S$ is a set of fairness constraints, each of which is a subset of states.

A joint action $a = (a_1, ..., a_n)$ *completes* a partially joint action $a_\Gamma = (a'_i, ..., a'_j)$ composed of actions of agents in $\Gamma \subseteq Ag$—written $a_\Gamma \sqsubseteq a$—if actions in a for agents in Γ correspond to actions in a_Γ. Furthermore, we define the function $img : S \times Act \to 2^S$ as $img(s,a) = \{s' \in S | s \xrightarrow{a} s'\}$, i.e. $img(s,a)$ is the set of states reachable in one step from s through a.

A model M represents a non-deterministic system where each agent has an imperfect information about the current global state. One restriction is made on T: $\forall s, s' \in S, s \sim_i s' \implies enabled(s,i) = enabled(s',i)$ where $enabled(s,i) = \{a_i \in Act_i | \exists s' \in S, a \in Act \text{ s.t. } (a_i) \sqsubseteq a \wedge s \xrightarrow{a} s'\}$. This means that the actions

an agent can perform in two epistemically equivalent states are the same. The *enabled* function is straightforwardly extended to groups of agents.

A *path* in a model M is a sequence $\pi = s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} ...$ of elements of T. We use $\pi(d)$ for s_d. A state s is *reachable* in M if there exist a path π and $d \geq 0$ such that $\pi(0) \in I$ and $\pi(d) = s$. A path π is *fair* according to a set of fairness conditions $F = \{f_1, ..., f_k\}$ if for each fairness condition f, there exist infinitely many positions $d \geq 0$ such that $\pi(d) \in f$. A state s is *fair* if there exists a fair path starting at s.

A *strategy* for agent i is a function $f_i : S \to Act_i$ where, for any state s, $f_i(s) \in enabled(s,i)$; a strategy maps each state to an enabled action. We call these strategies *global strategies*. A *uniform strategy* for agent i is a global strategy f_i where $\forall s, s' \in S, s' \sim_i s \implies f_i(s) = f_i(s')$, i.e. agent i cannot choose two different actions for two indistinguishable states. The *strategy outcomes* from a state s for a strategy f_i, denoted with $out(s, f_i)$, is the set of paths a strategy can enforce, i.e. $out(s, f_i) = \{\pi = s_0 \xrightarrow{a_1} s_1 ... | s_0 = s \wedge \forall d \geq 0, s_{d+1} \in img(s_d, a_{d+1}) \wedge (f_i(s_d)) \sqsubseteq a_{d+1}\}$. The definition of outcomes is naturally extended to sets of strategies for a subset of agents.

Semantics The semantics of both logics are defined over states of a model M by defining the relations $M, s \models^F_{fo} \phi$ and $M, s \models^F_{po} \phi$, for $ATLK^F_{fo}$ and $ATLK^F_{po}$, respectively. M can be omitted when clear from the context. Both relations share a part of their semantics; we write $s \models^F \phi$ if $s \models^F_{fo} \phi$ and $s \models^F_{po} \phi$. The $s \models^F_{fo} \phi$ and $s \models^F_{po} \phi$ relations are recursively defined over the structure of ϕ and follow the standard interpretation for most of the operators. $s \models^F p$ if $p \in V(s)$; $\vee$ and $\neg$ are interpreted in the natural way. $s \models^F K_i \phi$ if ϕ is true in all fair reachable states indistinguishable from s for agent i, $s \models^F E_\Gamma \phi$ if all agents in Γ know ϕ, $s \models^F D_\Gamma \phi$ if, by putting all their knowledge in common, agents of Γ would know ϕ, and $s \models^F C_\Gamma \phi$ if ϕ is common knowledge among agents of Γ [6]. $s \models^F E\psi$ if there is a path π starting at s satisfying ψ, $\pi \models^F X\phi$ if $\pi(1)$ satisfies ϕ, $\pi \models^F \phi_1 U \phi_2$ if ϕ_1 is true along the path until ϕ_2 is true, $\pi \models G\phi$ if ϕ is always true along π, and $\pi \models \phi_1 W \phi_2$ if $\pi \models (\phi_1 U \phi_2) \vee G\phi_1$ [4].

The meaning of the $\langle\Gamma\rangle$ operator is different in the two semantics:
(i) $s \models^F_{fo} \langle\Gamma\rangle\psi$ iff there exists a set of **global strategies** f_Γ, one for each agent in Γ, such that for all **fair paths** $\pi \in out(s, f_\Gamma), \pi \models^F \psi$;
(ii) $s \models^F_{po} \langle\Gamma\rangle\psi$ iff there exists a set of **uniform strategies** f_Γ, one for each agent in Γ, such that for all $s' \sim_\Gamma s$, for all **fair paths** $\pi \in out(s', f_\Gamma), \pi \models^F \psi$.

The $[\Gamma]$ operator is the dual of $\langle\Gamma\rangle$: $s \models^F [\Gamma]\psi$ iff $s \models^F \neg\langle\Gamma\rangle\neg\psi$.

3 Model Checking $ATLK^F_{fo}$ and $ATLK^F_{po}$

Model checking $ATLK^F_{fo}$ The model checking algorithm for $ATLK^F_{fo}$ is defined by the function $[\![.]\!]^F_{fo} : ATLK^F_{fo} \to 2^S$ returning the set of states of a given model M satisfying a given $ATLK^F_{fo}$ property. This function is defined in the standard way for Boolean connectors, CTL and knowledge operators [4, 13]. The $[\Gamma]$ operators are evaluated as follows:

$$[\![[\Gamma]X\phi]\!]^F_{fo} = Pre_{[\Gamma]}([\![\phi]\!]^F_{fo} \cap Fair_{[\Gamma]})$$

$$[\![[\Gamma]\phi_1 U \phi_2]\!]^F_{fo} = \mu Z.([\![\phi_2]\!]^F_{fo} \cap Fair_{[\Gamma]}) \cup ([\![\phi_1]\!]^F_{fo} \cap Pre_{[\Gamma]}(Z))$$

$$[\![[\Gamma]G\phi]\!]^F_{fo} = \nu Z.[\![\phi]\!]^F_{fo} \cap \bigcap_{f \in F} Pre_{[\Gamma]}(\mu Y.(Z \cap f) \cup ([\![\phi]\!]^F_{fo} \cap Pre_{[\Gamma]}(Y)))$$

$$[\![[\Gamma]\phi_1 W \phi_2]\!]^F_{fo} = \begin{array}{l} \nu Z.([\![\phi_2]\!]^F_{fo} \cap Fair_{[\Gamma]}) \\ \quad \cup ([\![\phi_1]\!]^F_{fo} \cap \bigcap_{f \in F} Pre_{[\Gamma]}(\mu Y.([\![\phi_2]\!]^F_{fo} \cap Fair_{[\Gamma]}) \cup (Z \cap f) \cup ([\![\phi_1]\!]^F_{fo} \cap Pre_{[\Gamma]}(Y)))) \end{array}$$

where $Pre_{[\Gamma]}(Z) = \{s | \forall a_\Gamma \in enabled(s,\Gamma), \exists a \text{ s.t. } a_\Gamma \sqsubseteq a \wedge img(s,a) \cap Z \neq \emptyset\}$ and $Fair_{[\Gamma]} = [\![[\Gamma] G\, true]\!]^F_{fo}$. $\mu Z.\tau(Z)$ and $\nu Z.\tau(Z)$ are the least and greatest fix points of function $\tau(Z)$. Intuitively, the $Pre_{[\Gamma]}(Z)$ operator returns the set of states in which Γ cannot avoid to reach a state of Z. Thus, $[\![[\Gamma] G\phi]\!]^F_{fo}$ returns the set of states in which Γ cannot avoid a path of states of $[\![\phi]\!]^F_{fo}$ going through all fairness constraints infinitely often; $Fair_{[\Gamma]}$ is the set of states in which Γ cannot avoid a fair path. Note that the $\langle\Gamma\rangle$ operators can be computed using the $[\Gamma]$ and $\neg$ operators, but can also be computed directly using the dual forms from the ones above. For example $[\![\langle\Gamma\rangle G\phi]\!]^F_{fo} = \nu Z.([\![\phi]\!]^F_{fo} \cup \overline{Fair_{[\Gamma]}}) \cap Pre_{\langle\Gamma\rangle}(Z)$, where $Pre_{\langle\Gamma\rangle}(Z) = \overline{Pre_{[\Gamma]}(\overline{Z})} = \{s | \exists a_\Gamma \in enabled(s,\Gamma)$ such that $\forall a, a_\Gamma \sqsubseteq a \implies img(s,a) \subseteq Z\}$. $\overline{Z} \subseteq S$ is the complement of the set $Z \subseteq S$.

The correctness of the model checking algorithm for $ATLK^F_{fo}$ follows from Theorem 1.

Theorem 1. *For all states $s \in S$, $s \models^F_{fo} \phi$ if and only if $s \in [\![\phi]\!]^F_{fo}$.*

Proof sketch. First, $Reach_{[\Gamma]}(P_1, P_2) = \mu Y.P_2 \cup (P_1 \cap Pre_{[\Gamma]}(Y))$ computes the set of states in which Γ cannot avoid a finite path of states of P_1 to a state of P_2. We can prove it by induction over the computation of the least fix point. It is true by definition of the least fix point and the $Pre_{[\Gamma]}$ operation.

Then, for the $[\Gamma]G\phi$ operator, $[\![[\Gamma] G\phi]\!]^F_{fo} = \nu Z.[\![\phi]\!]^F_{fo} \cap \bigcap_{f\in F} Pre_{[\Gamma]}(\mu Y.(Z \cap f) \cup ([\![\phi]\!]^F_{fo} \cap Pre_{[\Gamma]}(Y)))$ $= \nu Z.[\![\phi]\!]^F_{fo} \cap \bigcap_{f\in F} Pre_{[\Gamma]}(Reach_{[\Gamma]}([\![\phi]\!]^F_{fo}, Z \cap f))$ computes the set of states in which Γ cannot avoid a fair path (i.e. going through each $f \in F$ infinitely often) that satisfies $G\phi$. We prove it by induction over the computation of the greatest fix point and by using what has been proved just above.

Thanks to this, we can easily prove that $Fair_{[\Gamma]} = [\![[\Gamma] G true]\!]^F_{fo}$ computes the set of states in which Γ cannot avoid a fair path (it is just a particular case of the $[\Gamma]G$ operator).

Then, $[\Gamma]X$ and $[\Gamma]U$ operators compute the set of states in which Γ cannot avoid a successor in $[\![\phi]\!]^F_{fo}$ in which Γ cannot avoid a fair path, respectively in which Γ cannot avoid a finite path through states of $[\![\phi_1]\!]^F_{fo}$ to a state of $[\![\phi_2]\!]^F_{fo}$, in which Γ cannot avoid a fair path. In particular, the proof for $[\Gamma]U$ directly follows from the proof for $Reach_{[\Gamma]}$.

Finally, the proof for the $[\Gamma]W$ operator is similar to the one for $[\Gamma]G$ operator. The proof of correctness of the algorithms for $\langle\Gamma\rangle$ operators follows from the proof for $[\Gamma]$ operators, the duality of these operators and standard fix point properties. □

Model checking $ATLK^F_{po}$ – basic algorithm A basic algorithm is presented in Algorithm 1. It relies on the model checking algorithm for $ATLK^F_{fo}$. It uses two sub-algorithms: *Split* and $[\![.]\!]^F_{fo}|_{strat}$, where *strat* is a strategy represented as a set of state/action pairs. The latter is a modified version of the algorithm described in the previous section with $Pre_{\langle\Gamma\rangle}|_{strat}$ replacing $Pre_{\langle\Gamma\rangle}$ where $Pre_{\langle\Gamma\rangle}|_{strat}(Z) = \{s | \exists a_\Gamma \in enabled(s,\Gamma)$ such that $\langle s, a_\Gamma\rangle \in strat \wedge \forall a, a_\Gamma \sqsubseteq a \implies img(s,a) \subseteq Z\}$, i.e., $Pre_{\langle\Gamma\rangle}|_{strat}(Z)$ is $Pre_{\langle\Gamma\rangle}(Z)$ restricted to states and actions allowed by *strat*. Furthermore, $[\![.]\!]^F_{fo}|_{strat}$ recursively calls $[\![.]\!]^F_{po}$ on sub-formulae, instead of $[\![.]\!]^F_{fo}$.

The *Split* algorithm is given in Algorithm 2. $Split(S \times Act_\Gamma)$ returns the set of uniform strategies of the system (a uniform strategy is represented by the action for group Γ allowed in each state, and this action needs to be the same for each state in the same equivalence class).

Intuitively, Algorithm 1 computes, for each possible uniform strategy *strat*, the set of states for which the strategy is winning, and then keeps only the states s for which the strategy is winning for all states equivalent to s.

Before proving the correctness of the basic algorithm, let's prove the correctness of the *Split* algorithm.

Theorem 2. *$Split(Strats)$ computes the set of all the largest subsets SA of $Strats \subseteq S \times Act_\Gamma$ such that no conflicts appear in SA.*

Algorithm 1: $[\![\langle\Gamma\rangle\psi]\!]^F_{po}$

Data: M a given (implicit) model, Γ a subset of agents of M, ψ an $ATLK^F_{po}$ path formula.
Result: The set of states of M satisfying $\langle\Gamma\rangle\psi$.

$sat = \{\}$
for $strat \in Split(S \times Act_\Gamma)$ **do**
  $winning = [\![\langle\Gamma\rangle\psi]\!]^F_{fo}|_{strat}$
  $sat = sat \cup \{s \in winning | \forall s' \sim_\Gamma s, s' \in winning\}$
return sat

Algorithm 2: $Split(Strats)$

Data: $Strats \subseteq S \times Act_\Gamma$.
Result: The set of all the largest subsets SA of $Strats \subseteq S \times Act_\Gamma$ such that no conflicts appear in SA.

$C = \{\langle s, a_\Gamma\rangle \in Strats | \exists \langle s', a'_\Gamma\rangle \in Strats \; s.t. \; s' \sim_\Gamma s \wedge a_\Gamma \neq a'_\Gamma\}$
if $C = \emptyset$ **then return** $\{Strats\}$
else
  $\langle s, a_\Gamma\rangle =$ pick one in C
  $E = \{\langle s', a'_\Gamma\rangle \in Strats | s' \sim_\Gamma s\}$
  $A = \{a_\Gamma \in Act_\Gamma | \exists \langle s, a_\Gamma\rangle \in E\}$
  $strats = \{\}$
  for $a_\Gamma \in A$ **do**
    $S = \{\langle s', a_\Gamma\rangle \in E | a'_\Gamma = a_\Gamma\}$
    $strats = strats \cup Split(S \cup (Strats \backslash E))$
  return $strats$

Remark 1. *A conflict appears in $SA \subseteq S \times Act_\Gamma$ if there exist two elements $\langle s, a_\Gamma\rangle$ and $\langle s', a'_\Gamma\rangle$ in SA such that $s' \sim_\Gamma s$ and $a_\Gamma \neq a'_\Gamma$, i.e. there is a conflict if SA proposes two different actions in two equivalent states.*

Proof sketch of Theorem 2. *Split* gets all the conflicting elements of *Strats*. If there are no such elements, then *Strats* is its own largest non-conflicting subset; otherwise, *Split* takes one conflicting equivalence class E and, for each of its largest non-conflicting subsets S—i.e. subsets of states using the same action—it calls *Split* on the rest of *Strats* augmented with the non-conflicting subset S.

We can prove the correctness of *Split* by induction over the number of conflicting equivalence classes of *Strats*. If *Strats* does not contain any conflicting equivalence classes, *Strats* is its own single largest subset in which no conflicts appear. Otherwise, let's assume that $Split(Starts \backslash E)$ with E a conflicting equivalence class of *Strats* returns the set of all the largest non-conflicting subsets of $Strats \backslash E$; then, by what has been explained above, *Split* returns the cartesian product between all the largest non-conflicting subsets of E and all the largest non-conflicting subsets of $Strats \backslash E$. Because these cannot be conflicting as they belong to different equivalence classes, we can conclude that *Split* returns the set of the largest non-conflicting subsets of *Strats*. □

The correctness of Algorithm 1 is then given by the following theorem.

Theorem 3. *$[\![\langle\Gamma\rangle\psi]\!]^F_{po}$ computes the set of states of M satisfying $\langle\Gamma\rangle\psi$, i.e.*

$$\forall s \in S, s \in [\![\langle\Gamma\rangle\psi]\!]^F_{po} \text{ iff } s \models^F_{po} \langle\Gamma\rangle\psi.$$

Proof sketch. First, $Split(S \times Act_\Gamma)$ returns all the possible uniform strategies of the system, where a uniform strategy is represented by the only action allowed in each equivalence class of states—states equivalent in terms of the knowledge of Γ—, this action being the same for every state of the class.

Indeed, the set of the largest non-conflicting subsets of $S \times Act_\Gamma$ is the set of possible uniform strategies. A non-conflicting subset of $S \times Act_\Gamma$ provides at most one action for each equivalence class of states, otherwise it would not be non-conflicting; second, a largest non-conflicting subset of $S \times Act_\Gamma$ provides exactly one action for each equivalence class of states, otherwise there would be a larger subset giving one action for the missing equivalence classes and this subset would not be conflicting. Finally, a largest non-conflicting subset of $S \times Act_\Gamma$ is a uniform strategy because it is exactly the definition of a uniform strategy: giving one possible action for each equivalence class. This thus ends the proof that $Split$ returns the set of all possible uniform strategies.

Second, $winning = [\![\Gamma]\!]\psi]\!]^F_{fo}\psi|_{strat}$ returns the set of states for which the strategy $strat$ is winning. Indeed, it uses $ATLK^F_{fo}$ model checking algorithm, restricted to actions in $strat$. It thus returns the set of states for which there is a (global) winning strategy in $strat$. As $strat$ is, by construction, a uniform strategy, $winning$ is the set of states for which there exists a uniform winning strategy—in fact, it is $strat$ itself.

Finally, the set $\{s \in winning | \forall s' \sim_\Gamma s, s' \in winning\}$ is the set of states s for which $strat$ is a winning strategy for all $s' \sim_\Gamma s$. sat thus accumulates all the states s for which there is a winning strategy for all states indistinguishable from s. As this is exactly the semantics of the property, i.e. sat is exactly the set of states of the system satisfying the property, the proof is done. □

Improving the basic algorithm The first improvement proposed for the basic algorithm is the pre-filtering of states to the ones satisfying the property under $ATLK^F_{fo}$; we can filter them because if a state s does not satisfy $\langle\Gamma\rangle\psi$ under $ATLK^F_{fo}$, s cannot satisfy $\langle\Gamma\rangle\psi$ under $ATLK^F_{po}$. The second one is the alternation between filtering and splitting the strategies. Both improvements are aimed at reducing the number of uniform strategies to consider. The improved algorithm is presented in Algorithm 3. Using this algorithm, we can compute $[\![\langle\Gamma\rangle\psi]\!]^F_{po}$ as $Improved[\![\langle\Gamma\rangle\psi]\!]^F_{po}|_{S\times Act_\Gamma}$. The intuition behind Algorithm 3 is to start by computing the set of states satisfying the property and the associated actions (line 1), then get all conflicts (line 2) and, if there are conflicts, choose one conflicting equivalence class of states and possible actions (lines 6 to 8) and for each possible action a_Γ, recursively call the algorithm with the strategies following a_Γ (lines 11 and 12)—i.e. split the class into uniform strategies for this class and recursively call the algorithm on each strategy.

More in detail, Algorithm 3 returns the set of states satisfying the property in $Strats$. So, to get the final result, we have to take all the states satisfying the property in $S \times Act_\Gamma$. Algorithm 3 uses the function $[\![.]\!]^{F,ac}_{fo}|_{strats}$. This function is a modification of the $[\![.]\!]^F_{fo}|_{strats}$ function where actions are linked to states. More precisely, every sub-call to $[\![.]\!]^F_{po}$ or $\overline{Fair_{[\Gamma]}}$ is enclosed by $StatesActions_\Gamma|_{strats}$ to get all enabled actions in these states, restricted to $strats$—$StatesActions_\Gamma|_{strats}(Z) = \{\langle s, a_\Gamma\rangle \in strats | s \in Z \wedge a_\Gamma \in enabled(s,\Gamma)\}$—, and $Pre_{\langle\Gamma\rangle}|_{strats}$ is replaced by $Pre^{ac}_{\langle\Gamma\rangle}|_{strats}(Z) = \{\langle s, a_\Gamma\rangle \in strats | a_\Gamma \in enabled(s,\Gamma) \wedge \forall a, a_\Gamma \sqsubseteq a \implies img(s,a) \subseteq Z\}$. For example, $[\![[\Gamma]G\phi]\!]^{F,ac}_{fo}|_{Strats} = \nu Z.(StatesActions_\Gamma|_{Strats}([\![\phi]\!]^F_{po} \cup \overline{Fair_{[\Gamma]}})) \cap Pre^{ac}_{\langle\Gamma\rangle}|_{Strats}(Z)$.

Intuitively, $StatesActions_\Gamma|_{strats}(Z)$ returns all the states of Z with their enabled actions allowed by $strats$ and $Pre^{ac}_{\langle\Gamma\rangle}|_{strats}(Z)$ returns the states that can enforce to reach Z in one step, and the actions that

Algorithm 3: $Improved[\![\langle\Gamma\rangle\psi]\!]^F_{po}|_{Strats}$

Data: M a given (implicit) model, Γ a subset of agents of M, ψ an $ATLK^F_{po}$ path formula, $Strats \subseteq S \times Act_\Gamma$.

Result: The set of states of M satisfying $\langle\Gamma\rangle\psi$ in $Strats$.

1 $Z = [\![\langle\Gamma\rangle\psi]\!]^{F,ac}_{fo}|_{Strats}$
2 $C = \{\langle s, a_\Gamma\rangle \in Z | \exists \langle s', a'_\Gamma\rangle \in Z$ such that $s \sim_\Gamma s' \wedge a_\Gamma \neq a'_\Gamma\}$
if $C = \emptyset$ **then**
4 **return** $\{s \in S | \exists a_\Gamma \in Act_\Gamma$ s.t. $\forall s' \sim_\Gamma s, \langle s', a_\Gamma\rangle \in Z\}$
else
6 $\langle s, a_\Gamma\rangle =$ pick one in C
7 $E = \{\langle s', a'_\Gamma\rangle \in Z | s \sim_\Gamma s'\}$
8 $A = \{a_\Gamma \in Act_\Gamma | \exists \langle s, a_\Gamma\rangle \in E\}$
$sat = \{\}$
for $a_\Gamma \in A$ **do**
11 $strat = \{\langle s', a'_\Gamma\rangle \in E | a'_\Gamma = a_\Gamma\} \cup (Z \backslash E)$
12 $sat = sat \cup Improved[\![\langle\Gamma\rangle\psi]\!]^F_{po}|_{strat}$
return sat

allow them to do so, restricted to actions in $strats$. $[\![\langle\Gamma\rangle\psi]\!]^{F,ac}_{fo}|_{strats}$ thus returns the states satisfying $\langle\Gamma\rangle\psi$ associated to the actions of $strats$ that allow them to do so.

The correctness of Algorithm 3 is given by the following theorem.

Theorem 4. *$Improved[\![\langle\Gamma\rangle\psi]\!]^F_{po}|_{S\times Act_\Gamma}$ computes the set of states of M satisfying $\langle\Gamma\rangle\psi$, i.e.*

$$\forall s \in S, s \in Improved[\![\langle\Gamma\rangle\psi]\!]^F_{po}|_{S\times Act_\Gamma} \text{ iff } s \models^F_{po} \langle\Gamma\rangle\psi.$$

Proof sketch. First, $[\![\langle\Gamma\rangle\psi]\!]^{F,ac}_{fo}|_{Strats}$ returns the set of states s (and associated actions) such that there exists a global strategy in $Strats$ allowing Γ to enforce the property in s. This means that if a state/action pair is not returned, Γ has no global strategy to enforce the property from the given state by using the action given in the pair. By extension, there is no uniform strategy to enforce the property neither. Thus, only state/action pairs returned by $[\![\langle\Gamma\rangle\psi]\!]^{F,ac}_{fo}|_{Strats}$ have to be considered when searching for a uniform strategy in $Strats$. This also means that $[\![\langle\Gamma\rangle\psi]\!]^{F,ac}_{fo}|_{Strats}$ filters $Strats$ to winning global strategies; if the result is also a uniform strategy, all the states in the returned set have a uniform strategy to enforce the property.

Second, $Improved[\![\langle\Gamma\rangle\psi]\!]^F_{po}|_{Strats}$ returns the set of states satisfying the property in $Strats$. We can prove this by induction on the number of conflicting equivalence classes of $Strats$: this is true if there are no conflicting classes because Line 1 computes a winning uniform strategy—as discussed above—and Line 4 returns the set of states for which the strategy is winning for all indistinguishable states. This is also true in the inductive case because (1) filtering with $[\![\langle\Gamma\rangle\psi]\!]^{F,ac}_{fo}|_{Strats}$ doesn't lose potential state/action pairs and (2) the algorithm takes one conflicting class and tries all the possibilities for this class.

The final result thus is correct since it returns the set of states s for which there is a uniform strategy in $S \times Act_\Gamma$ that is winning for all states equivalent to s. □

Complexity considerations Model checking ATL with perfect recall and partial observability is an undecidable problem [14], while model checking ATL_{ir} is a Δ^P_2-complete problem [9]. $ATLK^F_{po}$ subsumes

ATL_{ir} and its model checking problem is therefore Δ_2^P-hard. Algorithm 1 performs a call to $[[.]]^F_{fo}$ for each uniform strategy: $[[.]]^F_{fo}$ is in **P**, but in the worst case there could be exponentially many calls to this procedure, as there could be up to $\prod_{i\in\Gamma} |Act_i|^{|S_i|}$ uniform strategies to consider.

4 Conclusion

A number of studies in the past have investigated the problem of model checking strategies under partial observability and, separately, some work has provided algorithms for including fairness constraints on *actions* in the case of full observability. To the best of our knowledge, the issue of fairness constraints and partial observability have never been addressed together.

In this paper we presented $ATLK^F_{po}$, a logic combining partial observability and fairness constraints on *states* (which is the standard approach for temporal and epistemic logics), and we have provided a model checking algorithm.The proposed algorithm is similar to the one of Calta et al. [3]. They also split possible actions into uniform strategies, but they do not provide a way to deal with fairness constraints.

Finally, the structure of our algorithm is compatible with symbolic model checking using OBDDs, and we are working on its implementation in the model checker MCMAS [12], where fairness constraints are only supported for temporal and epistemic operators.

References

[1] Rajeev Alur, Thomas A. Henzinger & Orna Kupferman (2002): *Alternating-time temporal logic*. *J. ACM* 49(5), pp. 672–713, doi:10.1145/585265.585270.

[2] Christel Baier & Joost-Pieter Katoen (2008): *Principles of Model Checking*. The MIT Press.

[3] Jan Calta, Dmitry Shkatov & Holger Schlingloff (2010): *Finding Uniform Strategies for Multi-agent Systems*. In Jürgen Dix, João Leite, Guido Governatori & Wojtek Jamroga, editors: *Computational Logic in Multi-Agent Systems*, *Lecture Notes in Computer Science* 6245, Springer Berlin / Heidelberg, pp. 135–152, doi:10.1007/978-3-642-14977-1_12.

[4] E. M. Clarke, O. Grumberg & D. Peled (1999): *Model Checking*. MIT Press.

[5] Mehdi Dastani & Wojciech Jamroga (2010): *Reasoning about strategies of multi-agent programs*. In: *Proceedings of AAMAS 10*, pp. 997–1004.

[6] Ronald Fagin, Joseph Y. Halpern, Yoram Moses & Moshe Y. Vardi (1995): *Reasoning about Knowledge*. MIT Press, Cambridge.

[7] Wiebe van der Hoek & Michael Wooldridge (2003): *Cooperation, Knowledge, and Time: Alternating-time Temporal Epistemic Logic and its Applications*. *Studia Logica* 75, pp. 125–157, doi:10.1023/A:1026185103185.

[8] W. Jamroga & T. Ågotnes (2007): *Constructive knowledge: what agents can achieve under imperfect information*. *Journal of Applied Non-Classical Logics* 17(4), pp. 423–475, doi:10.3166/jancl.17.423-475.

[9] Wojciech Jamroga & Jürgen Dix (2006): *Model Checking Abilities under Incomplete Information Is Indeed Δ_2^P-complete*. In: *EUMAS'06*.

[10] Wojciech Jamroga & Wiebe van der Hoek (2004): *Agents that Know How to Play*. *Fundamenta Informaticae* Volume 63(2), pp. 185–219.

[11] Sascha Klüppelholz & Christel Baier (2008): *Alternating-Time Stream Logic for Multi-agent Systems*. In: *Coordination Models and Languages*, LNCS 5052, Springer, pp. 184–198, doi:10.1007/978-3-540-68265-3_12.

[12] A. Lomuscio, H. Qu & F. Raimondi (2009): *MCMAS: A Model Checker for the Verification of Multi-Agent Systems*. In: *Proceedings of CAV 2009*, *LNCS* 5643, Springer, pp. 682–688, doi:10.1007/978-3-642-02658-4_55.

[13] Alessio Lomuscio & Wojciech Penczek (2007): *Symbolic model checking for temporal-epistemic logics*. *SIGACT News* 38(3), pp. 77–99, doi:10.1145/1324215.1324231.

[14] Pierre-Yves Schobbens (2004): *Alternating-time logic with imperfect recall*. *Electronic Notes in Theoretical Computer Science* 85(2), pp. 82 – 93, doi:10.1016/S1571-0661(05)82604-0.