Towards Partial Order Reductions for Strategic Ability

Wojciech Jamroga and Wojciech Penczek*
Institute of Computer Science, Polish Academy of Sciences
01-248 Warsaw, Poland
{jamroga,penczek}@ipipan.waw.pl

Piotr Dembiński and Antoni Mazurkiewicz
Institute of Computer Science, Polish Academy of Sciences
01-248 Warsaw, Poland
{piotrd,amaz}@ipipan.waw.pl

ABSTRACT

We propose a general semantics for strategic abilities of agents in asynchronous systems, with and without perfect information. Based on the semantics, we show some general complexity results for verification of strategic abilities in asynchronous interaction. More importantly, we develop a methodology for *partial order reduction* in verification of agents with imperfect information. We show that the reduction preserves an important subset of strategic properties, both with and without the fairness assumption. Interestingly, the reduction does not work for strategic abilities under perfect information.

KEYWORDS

Alternating-time temporal logic; asynchronous systems; model checking; partial order reduction; Mazurkiewicz traces

ACM Reference Format:
Wojciech Jamroga and Wojciech Penczek and Piotr Dembiński and Antoni Mazurkiewicz. 2018. Towards Partial Order Reductions for Strategic Ability. In *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), Stockholm, Sweden, July 10–15, 2018,* IFAAMAS, 10 pages.

1 INTRODUCTION

Alternating-time temporal logic **ATL*** and its fragment **ATL** [5, 6] extend temporal logic with the notion of *strategic ability.* They allow to express statements about what agents (or groups of agents) can achieve. For example, $\langle\langle i\rangle\rangle$F$\mathsf{win}_i$ says that agent i can eventually win no matter what the other agents do, while $\langle\langle i, j\rangle\rangle$Gsafe expresses that agents i and j together can force the system to always remain in a safe state. Such properties can be useful for specification, verification, and reasoning about interaction in agent systems. Moreover, algorithms and tools for verification of strategic abilities have been in constant development for almost 20 years [2, 3, 9, 10, 16, 18, 19, 21, 36, 41, 50, 51, 63]. However, there are two caveats.

First, many tools and algorithmic solutions focus on agents with perfect information. This is clearly unrealistic in all but the simplest multi-agent scenarios. Still, the tendency is somewhat easy to understand, since model checking of **ATL** variants with imperfect information is $\mathbf{\Delta_2^P}$- to **PSPACE**-complete for agents playing memoryless strategies [13, 38, 66] and **EXPTIME**-complete to undecidable for agents with perfect recall of the past [11, 28, 34]. Moreover, the imperfect information semantics of **ATL** does not admit alternation-free fixpoint characterizations [14, 26, 27], which makes incremental synthesis of strategies difficult to achieve [16, 17, 36, 39, 63].

Secondly, the semantics of strategic logics are almost exclusively based on synchronous concurrent game models. That is, one implicitly assumes the existence of a global clock that triggers subsequent global events in the system. At each tick of the clock, all the agents choose their actions, and the system proceeds accordingly with the corresponding global transition. However, many real-life systems are inherently asynchronous. No less importantly, many systems that are synchronous at the implementation level can be more conveniently modeled as asynchronous on a more abstract level.

In this paper, we make the first step towards strategic analysis of such systems. Our contribution is threefold. First, we define a semantics of strategic abilities for agents in asynchronous systems, with and without perfect information. Secondly, we present some general complexity results for verification of strategic abilities in such systems. Thirdly, and most importantly, we adapt *partial order reduction (POR)* to model checking of strategic abilities for agents with imperfect information. We also demonstrate that POR allows to significantly reduce the size of the model, and thus to make the verification more feasible. In fact, we show that the most efficient variant of POR, defined for linear time logic **LTL**, can applied be almost directly. The (nontrivial) proof that the **LTL** reductions work also for the more expressive strategic operators is the main contribution of this paper. Interestingly, the scheme does *not* work for verification of agents with perfect information.

The outline of the paper is as follows. In Section 2, we introduce the structures to represent and reason about asynchronous multi-agent systems. In Section 3, we define the semantics of **ATL** for asynchronous systems. In Section 4, we show the general complexity results. Sections 5 and 6 put forward the theoretical foundations and the algorithms for partial order reduction. We conclude in Section 8.

Related work. Relevant related work is relatively scarce. Asynchronous semantics and partial order reduction for distributed systems were extensively studied in [31–33, 43, 45, 57–60, 62]. The most recent approaches include dynamic POR [1, 20, 30] and combine POR with symbolic methods [42, 44]. The only efficient approach to partial order reduction in a MAS context [48, 49] concerns standard temporal-epistemic logics.

Alur, Henzinger and Kupferman mentioned asynchronous systems in their seminal paper on **ATL** [6], but they modeled them as a special case of synchronous systems. Asynchronous omega-regular games were also considered in [64]. Reactive modules [2–4] feature several modes of asynchronous execution, but – to the best of our knowledge – this aspect has never been given a more systematic analysis. The work that comes closest to our new proposal is [25]

*Also affiliated with Siedlce University, Faculty of Science, Institute of Computer Science, 3-Maja 54, 08-110 Siedlce, Poland.

Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. André, S. Koenig (eds.), July 10–15, 2018, Stockholm, Sweden.

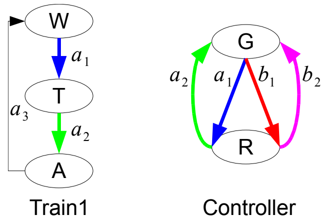

Figure 1: Asynchronous MAS for the TGC benchmark

where a variant of **ATL** was proposed for agent-oriented agent programs written in 2APL with asynchronous execution semantics.

2 MODELS OF MULTI-AGENT SYSTEMS

One can model multi-agent systems as networks of automata that execute asynchronously by interleaving local transitions, and synchronize their moves whenever a shared action is executed [29, 48].

Definition 2.1 (Asynchronous MAS). An *asynchronous multi-agent system (AMAS)* consists of n agents $\mathcal{A} = \{1, \ldots, n\}$, each associated with a tuple $A_i = (L_i, \iota_i, Act_i, P_i, T_i)$ including a set of *local states* $L_i = \{l_i^1, l_i^2, \ldots, l_i^{n_i}\}$, an *initial state* $\iota_i \in L_i$, and a set of *actions* $Act_i = \{a_i^1, a_i^2, \ldots, a_i^{m_i}\}$. Notice that the sets Act_i do not need to be disjoint. $Act = \bigcup_{i \in \mathcal{A}} Act_i$ is the set of all actions, and $Loc = \bigcup_{i \in \mathcal{A}} L_i$ is the set of all local states in the system. For each $a \in Act$, the set $Agent(a) = \{i \in \mathcal{A} \mid a \in Act_i\}$ contains the agents which have a in their sets of actions.

A *local protocol* $P_i : L_i \to 2^{Act_i}$ selects the actions available at each local state. Moreover, $T_i : L_i \times Act_i \rightharpoonup L_i$ is a (partial) *local transition function* such that $T_i(l_i, a) \neq undef$ iff $a \in P_i(l_i)$.

Example 2.2 (TGC). Figure 1 presents the Train-Gate-Controller (TGC) benchmark [3, 35]. The system consists of three agents: a controller c and two trains t_1, t_2. The trains run on separate circular tracks that jointly pass through a narrow tunnel. Each train can be waiting for the permission to enter (state W), riding inside the tunnel (T), or riding somewhere away of the tunnel (A). The controller switches between green light (state G) and red light (R). Initially, both trains are waiting and the controller displays Green.

2.1 Interleaved Interpreted Systems

To understand the interaction between asynchronous agents, we use *global states* and *global transitions*, defined formally below.

Definition 2.3 (Interleaved Interpreted System). Let $\mathcal{PV}$ be a set of propositional variables. An *interleaved interpreted system (IIS)*, or a *model*, is an asynchronous MAS extended with the following elements: a set $St \subseteq L_1 \times \cdots \times L_n$ of global states, an initial state $\iota \in St$, a global transition function $T : St \times Act \rightharpoonup St$, and a valuation of propositions $V : St \to 2^{\mathcal{PV}}$. For state $g = (l_1, \ldots, l_n)$, we denote the local component of agent i by $g^i = l_i$. Also, we will sometimes write $g_1 \xrightarrow{a} g_2$ instead of $T(g_1, a) = g_2$.

We say that action $a \in Act$ is *enabled* at $g \in St$ if $g \xrightarrow{a} g'$ for some $g' \in St$. The global transition function is assumed to be total, i.e., at each $g \in St$ there exists at least one enabled action.

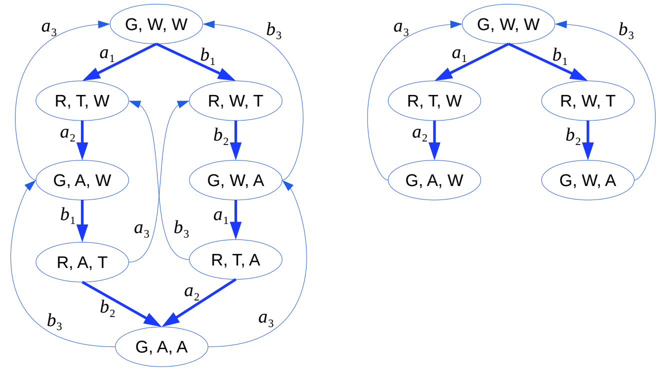

Figure 2: IIS for TGC: (a) full model, (b) reduced model. Visible transitions are depicted by blue bold arrows

An infinite sequence of global states and actions $\pi = g_0 a_0 g_1 a_1 g_2 \ldots$ is called a *path* if $g_i \xrightarrow{a_i} g_{i+1}$ for every $i \geq 0$. $Act(\pi) = a_0 a_1 a_2 \ldots$ is the sequence of actions in π, and $\pi[i] = g_i$ is the i-th global state of π. $\Pi_M(g)$ denotes the set of all paths in M starting at g.

IIS can be used to provide an execution semantics to AMAS.

Definition 2.4 (Canonical IIS). Let S be an asynchronous MAS with n agents. Its *canonical model* $IIS_V(S)$ extends S with the valuation V, global states $St = L_1 \times \ldots \times L_n$, initial state $\iota = (\iota_1, \ldots, \iota_n)$, and transition function T defined as follows: $T(g_1, a) = g_2$ iff $T_i(g_1^i, a) = g_2^i$ for all $i \in Agent(a)$, and $g_1^i = g_2^i$ for all $i \in \mathcal{A} \setminus Agent(a)$.

The state/transition structure of the canonical interleaved interpreted system for TCG is depicted in Figure 2a. Additionally, let us assume $\mathcal{PV} = \{\mathsf{in}_1, \mathsf{in}_2\}$ with $\mathsf{in}_i \in V(g)$ iff $g^i = T$. That is, proposition in_i denotes that train t_i is currently in the tunnel. It is easy to see that the global state space grows exponentially with the number of agents. In some cases, it suffices to consider a subset of states and transitions, i.e., a submodel of $IIS(S)$.

Definition 2.5 (Submodel). Let M, M' be two models extending the same AMAS, such that $St' \subseteq St$, $\iota \in St'$, T is an extension of T', and $V' = V|_{St'}$. Then, we write $M' \subseteq M$ and call M' a *submodel* of M or a *reduced model* of M.

An example submodel of the IIS for TCG is shown in Figure 2b. It is easy to see that, for each $g \in St'$, we have $\Pi_{M'}(g) \subseteq \Pi_M(g)$.

In order to generate reduced models, we need a notion of *invisibility* and *independency* of actions. Intuitively, an action is invisible iff it does not change the valuations of the propositions. Note that this concept of invisibility is technical, and is not connected to the view of any agent in the sense of [52]. Additionally, we can designate a subset of agents A whose actions are visible by definition. Furthermore, two actions are weakly independent iff they are not actions of the same agent, and strongly independent iff they are weakly independent and at least one of them is invisible.

Definition 2.6 (Invisible actions). Consider a model M, a subset of agents $A \subseteq \mathcal{A}$, and a subset of propositions $PV \subseteq \mathcal{PV}$. An action $a \in Act$ is *invisible* wrt. A and PV if $Agent(a) \cap A = \emptyset$ and for each two global states $g, g' \in St$ we have that $g \xrightarrow{a} g'$ implies

$V(g) \cap PV = V(g') \cap PV$. The set of all invisible actions for A, PV is denoted by $Invis_{A,PV}$, and its closure – of visible actions – by $Vis_{A,PV} = Act \setminus Invis_{A,PV}$.

Definition 2.7 (Independent actions). Weak independence $WI \subseteq Act \times Act$ is defined as: $WI = \{(a, b) \in Act \times Act \mid Agent(a) \cap Agent(b) = \emptyset\}$. *Strong independence* (or *independence*) $I_{A,PV} \subseteq Act \times Act$ is defined as: $I_{A,PV} = WI \setminus (Vis_{A,PV} \times Vis_{A,PV})$.

We assume in the rest of the paper that a suitable subset PV is given, and omit the subscript PV whenever clear from the context.

3 REASONING ABOUT AGENTS' ABILITIES

Many important properties in a MAS can be specified in terms of the strategic ability of some agents to achieve a given goal. Such properties can be specified by formulas of the strategic logic **ATL**. The semantics of **ATL** is typically defined for models of synchronous systems. In this section, we show how it can be adapted to asynchronous MAS.

3.1 Alternating-Time Temporal Logic: Syntax

Alternating-time temporal logic [5, 6] generalizes the branching-time temporal logic **CTL** [22] by replacing the path quantifiers E, A with *strategic modalities* $\langle\langle A\rangle\rangle$. Informally, $\langle\langle A\rangle\rangle\gamma$ expresses that the group of agents A has a collective strategy to enforce the temporal property γ. The formulas make use of temporal operators: "X" ("next"), "G" ("always from now on"), "F" ("now or sometime in the future"), U ("strong until"), and R ("release"). The logic comes in several syntactic variants, the most popular of which are **ATL*** and "vanilla **ATL**" (the latter often called simply "**ATL**").

Definition 3.1 (Syntax of **ATL****).* Let $\mathcal{PV}$ be a set of propositional variables and $\mathcal{A}$ the set of all agents. The language of **ATL*** is defined by the following grammar (where $\mathsf{p} \in \mathcal{PV}$ and $A \subseteq \mathcal{A}$):

$$\varphi ::= \mathsf{p} \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\langle A\rangle\rangle\gamma,$$
$$\gamma ::= \varphi \mid \neg\gamma \mid \gamma \wedge \gamma \mid \mathrm{X}\gamma \mid \gamma\,\mathrm{U}\,\gamma.$$

The other Boolean operators are defined as usual. "Release" can be defined as $\gamma_1 \,\mathrm{R}\, \gamma_2 \equiv \neg((\neg\gamma_1)\,\mathrm{U}\,(\neg\gamma_2))$. The "sometime" and "always" operators can be defined as $\mathrm{F}\gamma \equiv true\,\mathrm{U}\,\gamma$ and $\mathrm{G}\gamma \equiv false\,\mathrm{R}\,\gamma$. Moreover, "for all paths" can be defined as $\mathrm{A}\gamma \equiv \langle\langle\emptyset\rangle\rangle\gamma$.

Definition 3.2 (Syntax of **ATL***).* In "vanilla **ATL**," every occurrence of a strategic modality is immediately followed by a temporal operator. In that case, "release" is not definable from "until" anymore [47], and it must be added explicitly to the syntax:

$$\varphi ::= \mathsf{p} \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\langle A\rangle\rangle\mathrm{X}\varphi \mid \langle\langle A\rangle\rangle\varphi\,\mathrm{U}\,\varphi \mid \langle\langle A\rangle\rangle\varphi\,\mathrm{R}\,\varphi.$$

In the rest of the paper, we are mainly interested in formulas that do not use the next step operator X, and do not contain nested strategic modalities. We denote the corresponding subsets of **ATL*** and **ATL** by **sATL*** ("simple **ATL***") and **sATL** ("simple **ATL**"). Moreover, **1ATL*** is the fragment of **sATL*** that admits only formulas consisting of a single strategic modality followed by an **LTL** formula (i.e., $\langle\langle A\rangle\rangle\gamma$, where $\gamma \in$ **LTL**), and analogously for **1ATL**.

Example 3.3. The following formulas of **sATL*** specify interesting properties of the TCG system: $\langle\langle c\rangle\rangle\mathrm{F}\,\mathsf{in}_1$ (the controller can let train t_1 in), $\langle\langle c\rangle\rangle\mathrm{G}\neg\mathsf{in}_1$ (the controller can keep t_1 out forever), $\langle\langle c\rangle\rangle\mathrm{F}(\mathsf{in}_1 \wedge \mathrm{F}\neg\mathsf{in}_1)$ (the controller can let t_1 through), $\neg\langle\langle t_1, t_2\rangle\rangle\mathrm{F}(\mathsf{in}_1 \vee \mathsf{in}_2)$ (neither train can get in without the help of the controller, even if it collaborates with the other train).

We claim that most of practically interesting specifications of strategic ability can be expressed in **sATL***, possibly extended with epistemic operators. Nested strategic modalities allow to express an agent's ability to endow another agent with ability (or deprive the other agent of ability), which is seldom of practical interest.

3.2 Strategies and Outcomes

Let M be a model. A *strategy* of agent $i \in \mathcal{A}$ in M is a conditional plan that specifies what i is going to do in any potential situation. Here, we follow Schobbens [66], and adopt his taxonomy of four "canonical" strategy types: IR, iR, Ir, and ir. In the notation, R (resp. r) stands for perfect (resp. imperfect) *recall*, and I (resp. i) refers to perfect (resp. imperfect) *information*. Note that verification of **ATL** for agents with perfect recall is in general undecidable [28]. Because of that, we focus on memoryless strategies. Formally:

- A *memoryless perfect information strategy for agent i* is a function $\sigma_i : St \to Act_i$ st. $\sigma_i(g) \in P_i(g^i)$ for each $g \in St$.
- A *memoryless imperfect information strategy for i* is a function $\sigma_i : L_i \to Act_i$ st. $\sigma_i(l) \in P_i(l)$ for each $l \in L_i$.

Thus, a perfect information strategy can assign different actions to any two global states, while under imperfect information the agent's choices depend only on the local state of the agent.[1] A *joint strategy* σ_A for a coalition $A \subseteq \mathcal{A}$ is a tuple of strategies, one per agent $i \in A$. We denote the set of A's collective memoryless perfect (resp. imperfect) information strategies by Σ_A^{Ir} (resp. Σ_A^{ir}). Additionally, let $\sigma_A = (\sigma_1, \ldots, \sigma_k)$ be a joint strategy for $A = \{i_1, \ldots, i_k\}$. For each $g \in St$, we define $\sigma_A(g) = (\sigma_1(g), \ldots, \sigma_k(g))$.

Definition 3.4 (Outcome paths). Let $Y \in \{Ir, \mathrm{ir}\}$. The *outcome* of strategy $\sigma_A \in \Sigma_A^Y$ in state $g \in St$ is the set $out_M(g, \sigma_A) \subseteq \Pi_M(g)$ such that $\pi = g_0 a_0 g_1 a_1 g_2 \cdots \in out_M(g, \sigma_A)$ iff $g_0 = g$ and $\forall i \in \mathcal{N}$ $\forall j \in A$ if $j \in Agent(a_i)$, then $a_i \in \sigma_j(\pi[i])$ for $Y = \mathrm{Ir}$, and $a_i \in \sigma_j(\pi[i]^j)$ for $Y = \mathrm{ir}$.

Intuitively, the outcome of a joint strategy σ_A in a global state g is the set of all the infinite paths that can occur when in each state of the paths either some agents (an agent) in A execute(s) an action according to σ_A or some agents (an agent) in $\overline{A}$ execute(s) an action following their protocols. Clearly, each action a has to be executed by all agents which have a in their sets of actions. In reasoning about asynchronous systems, one often wants to look only at *fair* paths, i.e., ones that do not consistently ignore an agent whose action is always enabled. Formally, a path π satisfies the *concurrency-fairness condition* (**CF**) if there is no action enabled in all states of π from $\pi[i]$ on, and at the same time weakly independent from all the actions actually executed in $\pi[i], \pi[i+1], \pi[i+2], \ldots$. We denote the set of all such paths starting at g by $\Pi_M^{CF}(g)$.

*Definition 3.5 (**CF**-outcome).* The *concurrency-fair outcome* of $\sigma_A \in \Sigma_A^Y$ is defined as $out_M^{CF}(g, \sigma_A) = out_M(g, \sigma_A) \cap \Pi_M^{CF}(g)$.

Note that, in an arbitrary IIS, not every action admitted by agent i's protocol at some local state l_i must be enabled at all the global

[1] Alternatively, we can require the agent's choices to be the same for the global states that share the same local states.

states g with $g^i = l_i$. Moreover, it may be the case that the outcome of a strategy is empty, because all the paths consistent with the strategy get "stuck" at some global state. In Sections 5 and 6, we show how to construct reduced models so that this does not create problems. For the moment, we state the following two lemmas.

Lemma 3.6. *Let M' be a submodel of M. For each* ir*-joint strategy σ_A we have $out_{M'}(\iota, \sigma_A) = out_M(\iota, \sigma_A) \cap \Pi_{M'}(\iota)$. and $out^{CF}_{M'}(\iota, \sigma_A) = out^{CF}_M(\iota, \sigma_A) \cap \Pi^{CF}_{M'}(\iota)$.*

Proof. Notice that each ir-strategy in M is also a well defined ir-strategy in M' as it is defined on the local states of AMAS which is extended by M and M'. The lemma follows directly from Definitions 3.4 and 3.5, together with the fact that $\Pi_{M'}(\iota) \subseteq \Pi_M(\iota)$. □

Lemma 3.7. *Let M be a model, $\pi, \pi' \in \Pi_M(\iota)$, and for some $i \in \mathcal{A}$: $Act(\pi) \mid_{Act_i} = Act(\pi') \mid_{Act_i}$. Then, for each* ir*-strategy σ_i, we have $\pi \in out_M(\iota, \sigma_i)$ iff $\pi' \in out_M(\iota, \sigma_i)$.*

Proof sketch. Let $Act(\pi) \mid_{Act_i} = b_0 b_1 \ldots$. For each b_j let $\pi[b_j]$ denote the global state from which b_j is executed in π. By induction we can show that for each $j \geq 0$ we have $\pi[b_j]^i = \pi'[b_j]^i$. For $j = 0$ it is easy to notice that $\pi[b_0]^i = \pi[b_0]^i = \iota^i$.

Assume that the thesis holds for $j = k$. The induction step follows from the fact the local evolution T_i is a function, so if $\pi[b_k]^i = \pi'[b_k]^i = l$ for some $l \in L_i$, then $\pi[b_{k+1}]^i = \pi'[b_{k+1}]^i = T_i(l, b_k)$. This means that for each ir-strategy σ_i, we have $\pi \in out_M(\iota, \sigma_i)$ iff $\pi' \in out_M(\iota, \sigma_i)$, which concludes the proof. □

The lemma can be easily generalized to joint strategies $\sigma_A \in \Sigma^{\text{ir}}_A$. Note that the same property does not hold for perfect information strategies. This is because the current local state l_i can only change through the execution of an action by agent i, but the current global state can possibly change because of another agent's transition. Similarly, the analogue of Lemma 3.7 does not hold in synchronous models of MAS, since the local transitions of i in a synchronous model can be influenced by the actions selected by the other agents.

3.3 Asynchronous Semantics of ATL and ATL*

Our semantics of **ATL*** for asynchronous interaction, parameterised with the strategy type $Y \in \{\text{Ir}, \text{ir}\}$, is defined as follows:

- $M, g \models_Y \mathsf{p}$ iff $\mathsf{p} \in V(g)$, for $\mathsf{p} \in \mathcal{PV}$;
- $M, g \models_Y \neg\varphi$ iff $M, g \not\models_Y \varphi$;
- $M, g \models_Y \varphi_1 \wedge \varphi_2$ iff $M, g \models_Y \varphi_1$ and $M, g \models_Y \varphi_2$;
- $M, g \models_Y \langle\langle A \rangle\rangle\gamma$ iff there is a strategy $\sigma_A \in \Sigma^Y_A$ such that $out_M(g, \sigma_A) \neq \emptyset$ and, for each path $\pi \in out_M(g, \sigma_A)$, we have $M, \pi \models_Y \gamma$;
- $M, \pi \models_Y \varphi$ iff $M, \pi[0] \models_Y \varphi$;
- $M, \pi \models_Y \neg\gamma$ iff $M, \pi \not\models_Y \gamma$;
- $M, \pi \models_Y \gamma_1 \wedge \gamma_2$ iff $M, \pi \models_Y \gamma_1$ and $M, \pi \models_Y \gamma_2$;
- $M, \pi \models_Y \mathrm{X}\gamma$ iff $M, \pi[1, \infty] \models_Y \gamma$;
- $M, \pi \models_Y \gamma_1 \mathrm{U} \gamma_2$ iff $M, \pi[i, \infty] \models_Y \gamma_2$ for some $i \geq 0$ and $M, \pi[j, \infty] \models_Y \gamma_1$ for all $0 \leq j < i$.

The semantics of "vanilla **ATL**" can be given entirely with respect to states in the usual way.

Example 3.8. We leave it to the reader to check that all the formulas in Example 3.3 hold in the TGC model from Figure 2a for both the Ir and the ir semantics.

Remark 1. *We observe that the relation $\models_{\text{ir}}$ captures the "objective" notion of ability under imperfect information [15, 37]. That is, $\langle\langle A \rangle\rangle\gamma$ holds iff A have a collective strategy to enforce γ from the current global state of the system. We expect to obtain analogous results for the semantics based on "subjective" ability [15, 40, 66], but a detailed study is outside the scope of this paper.*

Remark 2. *Notice also that the semantics constrains the abilities behind $\langle\langle A \rangle\rangle$ to "no-deadlock" paths and strategies. That is, we only consider infinite execution paths, and only strategies whose outcomes are nonempty sets of such paths. This is in line with the standard approach to distributed systems. An interesting alternative would be to model executions with deadlock by paths ending with an infinite sequence of "silent" actions, looping in the deadlock state. We plan to study the resulting semantics of* **ATL*** *in the future.*

We obtain the *concurrency-fair semantics* $\models_{\text{IrF}}$ and $\models_{\text{irF}}$ by replacing $out_M(g, \sigma_A)$ with $out^{CF}_M(g, \sigma_A)$ in the clauses for $\langle\langle A \rangle\rangle$.

For the set of formulas $\mathcal{L}$ and the semantic relation $\models_Y$, we denote the logical system $(\mathcal{L}, \models_Y)$ by $\mathcal{L}_Y$. Thus, $\mathbf{ATL}_{\text{Ir}}$ is the "vanilla **ATL**" with memoryless perfect information semantics, $\mathbf{sATL}^*_{\text{irF}}$ is the "simple **ATL***" with memoryless imperfect information strategies and concurrency-fairness assumption, and so on.

4 MODEL CHECKING sATL AND sATL* FOR ASYNCHRONOUS AGENT SYSTEMS

In this work, we focus on simple specifications of strategic ability, i.e., ones that can be formally characterized without nesting strategic modalities. We believe that an overwhelming majority of properties, relevant in actual application domains, follow that pattern. One usually wants to require (or ask if) a given player has a strategy to eventually win the game ($\langle\langle i \rangle\rangle \mathrm{F}\mathsf{win_i}$), two trains can persistently avoid the crash ($\langle\langle t_1, t_2 \rangle\rangle \mathrm{G}\neg(\mathsf{in}_1 \wedge \mathsf{in}_2)$), etc. Moreover, in all realistic scenarios, players have only partial knowledge of the current global state of the world. Thus, we focus here on the semantics based on imperfect information strategies.

In this section, we establish the complexity of model checking for relevant fragments of $\mathbf{sATL}^*_{\text{ir}}$ and $\mathbf{sATL}^*_{\text{irF}}$. We observe that the complexity can be given with respect to the *model* of the system (i.e., an interleaved interpreted system, cf. Section 2.1), or the compact *representation* of the system (in our case, an asynchronous MAS, cf. Section 2). We give both kinds of results. Note that $IIS_V(S)$ has usually exponentially many global states and transitions in the number of agents in S. Thus, the model checking results relative to the size of $IIS_V(S)$ "hide" the part of the complexity already included in the blowup. On the other hand, POR reduces models and not representations, so the complexity wrt the size of the model tells us how much gain we can expect when the model is reduced.

4.1 Model Checking $\mathbf{1ATL}_{\text{ir}}$ and $\mathbf{1ATL}_{\text{irF}}$

We begin by looking at the verification complexity for simplest specifications, consisting of a single strategic modality $\langle\langle A \rangle\rangle$ immediately followed by a single temporal modality.

Proposition 4.1. *Model checking $\mathbf{1ATL}_{\text{ir}}$ and $\mathbf{1ATL}_{\text{irF}}$ is* **NP***-complete in the size of the model and the length of the formula. It remains* **NP***-complete even for formulas of bounded length.*

Proof sketch. Analogous to the result in [66] for $\langle\langle\Gamma\rangle\rangle$-$\mathbf{ATL}_{\mathrm{ir}}$.

For the upper bound, observe that model checking of $\langle\langle A\rangle\rangle\gamma$ in M, g can be done by (1) guessing an ir-joint strategy σ_A, (2) pruning M according to σ_A, and (3) model checking the **CTL** formulas: $\neg\mathrm{AG}\bot$ ("the set of paths is nonempty") and $\mathrm{A}\gamma$ ("for all paths, γ") in state g of the resulting model. Since σ_A is of at most linear size with respect to $|M|$, and model checking of $\mathrm{A}\gamma$ can be done in deterministic polynomial time w.r.t. $|M|$, both with and without fairness assumptions [7], we obtain the bound.

For the lower bound, we use the reduction of [66] of SAT to model checking of the formula $\langle\langle 1\rangle\rangle\mathrm{F}yes$ in a single-agent model (note that single-agent systems can be seen as special cases of both synchronous and asynchronous systems, and the semantics with and without fairness assumptions coincide on such models). Notice that the lower bound does not rely on the length of the formula. □

Proposition 4.2. *Model checking* $\mathbf{1ATL}_{\mathrm{ir}}$ *and* $\mathbf{1ATL}_{\mathrm{irF}}$ *is* **PSPACE**-*complete in the size of the representation (even for formulas of bounded length).*

Proof sketch. For the upper bound, observe that model checking of $\langle\langle A\rangle\rangle\gamma$ in an AMAS S can be done by: (1) guessing an ir strategy σ_A as a deterministic restriction of the protocols P_i, $i \in \mathcal{A}$; (2) pruning M; (3) model checking, in the resulting representation S', the **LTL** formulas: (3a) $\mathrm{G}\bot$ and (3b) γ (for $\mathbf{1ATL}_{\mathrm{ir}}$) or $\gamma \wedge fair$ (for $\mathbf{1ATL}_{\mathrm{ir}}$) where *fair* is an **LTL** characterization of fairness. The algorithm returns *true* iff the output of (3a) was *false* and that of (3b) was *true*. Since the size of σ_A is linear wrt $|S|$, and model checking **LTL** is in **PSPACE** wrt $|S|$ [65], we obtain the bound.

For the lower bound, we adapt the construction from [46]. Given a Turing machine T with space complexity $s(n)$, we construct the concurrent program $P(T)$ as in [46, Theorem 6.1]. According to that theorem, there exists a computation of T on the empty tape which eventually reaches an accepting state iff $P(T) \models_{\mathrm{CTL}} \mathrm{EF\,accept}$. Now, we observe that $P(T)$ is in fact an asynchronous MAS in the sense of Definition 2.1. Thus, $P(T) \models_{\mathrm{CTL}} \mathrm{EF\,accept}$ iff $IIS(P(T)) \not\models_{\mathrm{ir}} \langle\langle\emptyset\rangle\rangle\mathrm{G}\neg\mathrm{accept}$. This way we obtain the **co-PSPACE**-hardness for $\mathbf{1ATL}_{\mathrm{ir}}$ (recall that **co-PSPACE** = **PSPACE**).

For $\mathbf{1ATL}_{\mathrm{irF}}$, we observe that all the paths in $IIS(P(T))$ are fair, so the same construction can be used.

Again, the reduction does not rely on the length of the formula. □

4.2 Model Checking $\mathbf{sATL}_{\mathrm{ir}}$ and $\mathbf{sATL}_{\mathrm{irF}}$

The verification complexity for Boolean combinations of formulas from **1ATL** is almost the same.

Proposition 4.3. *Model checking* $\mathbf{sATL}_{\mathrm{ir}}$ *and* $\mathbf{sATL}_{\mathrm{irF}}$ *is* **NP**-*hard and in* $\Theta_2^{\mathbf{P}}$ *in the size of the model and the length of the formula (even for formulas of bounded length).*[2]

Proof sketch. The lower bounds follow from Proposition 4.1.

The following algorithm for checking φ in M, q demonstrates the upper bound. First, the non-deterministic algorithm in Proposition 4.1 is used as an oracle that determines the truth value for each subformula $\langle\langle A\rangle\rangle\gamma$ of φ. Clearly, the oracle is called at most $|\varphi|$ times, and the input in the next call does not depend on the output of the preceding calls. Finally, based on the output of the calls, the value of φ is calculated in the standard way. □

Proposition 4.4. *Model checking* $\mathbf{sATL}_{\mathrm{ir}}$ *and* $\mathbf{sATL}_{\mathrm{irF}}$ *is* **PSPACE**-*complete in the size of the representation and the length of the formula (even for formulas of bounded length).*

Proof sketch. The lower bounds follow from Proposition 4.2. For the upper bounds, we use the algorithm from Proposition 4.3, but with the algorithm from Proposition 4.2 as the oracle. Since $\mathbf{P}^{\mathbf{PSPACE}} = \mathbf{PSPACE}$, we obtain the result. □

4.3 Model Checking $\mathbf{sATL}^*_{\mathrm{ir}}$ and $\mathbf{sATL}^*_{\mathrm{irF}}$

Finally, we examine the complexity of verification for specifications with arbitrary **LTL** subformulas.

Proposition 4.5. *The following statements hold:*

1. *Model checking* $\mathbf{sATL}^*_{\mathrm{ir}}$, $\mathbf{1ATL}^*_{\mathrm{ir}}$, $\mathbf{sATL}^*_{\mathrm{irF}}$, $\mathbf{1ATL}^*_{\mathrm{irF}}$ *is* **PSPACE**-*complete in the size of the model and the formula.*
2. *For formulas of bounded length, the problem is* **NP**-*complete for* $\mathbf{1ATL}^*_{\mathrm{ir}}$ *and* $\mathbf{1ATL}^*_{\mathrm{irF}}$, *and between* **NP** *and* $\Theta_2^{\mathbf{P}}$-*complete for* $\mathbf{sATL}^*_{\mathrm{ir}}$ *and* $\mathbf{sATL}^*_{\mathrm{irF}}$.

Proof sketch. For (1), the lower bound follows from **PSPACE**-completeness of **LTL** model checking [65]. The upper bound for $\mathbf{1ATL}^*_{\mathrm{ir}}$ and $\mathbf{1ATL}^*_{\mathrm{irF}}$ can be obtained by guessing the strategy, pruning the model, and verifying the **LTL** formulas from Propositions 4.2 (note that $\mathbf{NP}^{\mathbf{PSPACE}} = \mathbf{PSPACE}$). For $\mathbf{sATL}^*_{\mathrm{ir}}$ and $\mathbf{sATL}^*_{\mathrm{irF}}$, we repeat this for each subformula, and compute the Boolean combination.

For (2), the lower bound follows from Proposition 4.1. The inclusion in **NP** for $\mathbf{1ATL}^*_{\mathrm{ir}}$ and $\mathbf{1ATL}^*_{\mathrm{irF}}$ can be obtained by an algorithm similar to that in Proposition 4.1, only an **LTL** rather than **CTL** model checker is called. Since **LTL** model checking is **NLOGSPACE**-complete for formulas of bounded size [65], and **NLOGSPACE** $\subseteq$ **P**, the upper bound follows.

The upper bound for $\mathbf{sATL}^*_{\mathrm{ir}}$ and $\mathbf{sATL}^*_{\mathrm{irF}}$ is obtained by an algorithm similar to that in Proposition 4.3, only an **LTL** rather than **CTL** model checker is called inside the oracle. □

Proposition 4.6. *Model checking* $\mathbf{sATL}^*_{\mathrm{ir}}$, $\mathbf{1ATL}^*_{\mathrm{ir}}$, $\mathbf{sATL}^*_{\mathrm{irF}}$, *and* $\mathbf{1ATL}^*_{\mathrm{irF}}$ *is* **PSPACE**-*complete in the size of the representation and the formula (even for the formulas of bounded length).*

Proof sketch. The lower bounds follow from Proposition 4.2. The upper bounds are obtained analogously to Proposition 4.4. □

4.4 Discussion

The above complexity results show that model checking fragments of $\mathbf{sATL}^*_{\mathrm{ir}}$ and $\mathbf{sATL}^*_{\mathrm{irF}}$ wrt compact representations (i.e., asynchronous MAS) is hard, and the size of the representation is the main factor for this hardness. Moreover, they suggest that there is no general method better than unfolding the representation to an explicit model (i.e., an interleaved interpreted system), and then verifying the IIS. This is because, with **PSPACE**-complete problems, one should expect exponential running time in practice. Thus, it is essential for the unfolding to *produce as small models as possible.* In what follows, we recall the idea of *partial order reduction*, very

[2] Where $\Theta_2^{\mathbf{P}} = \mathbf{P}^{||\mathbf{NP}}$ is the class of problems solvable by a deterministic polynomial-time Turing machine making polynomially many *nonadaptive* calls to an **NP** oracle.

important in verification of temporal properties, and show how it can be used to model-check formulas of $\mathbf{sATL}^*_{\mathrm{ir}}$ and $\mathbf{sATL}^*_{\mathrm{irF}}$.

5 PARTIAL ORDER REDUCTIONS

Partial order reductions have been defined for various configurations of temporal and temporal-epistemic logics without the "next step" operator X [31, 48, 49, 57, 62]. The idea is to generate reduced models that either preserve some kind of model equivalence, or preserve representatives of Mazurkiewicz traces. The former method was used, for instance, to construct POR for $\mathbf{LTL}_{-\mathbf{X}}$ and $\mathbf{LTLK}_{-\mathbf{X}}$ based on stuttering trace equivalence [48, 49], and to obtain reductions for $\mathbf{CTL}^*_{-\mathbf{X}}$ and $\mathbf{CTLK}_{-\mathbf{X}}$ based on stuttering bisimulation [31, 48, 49]. The latter method was applied e.g. to prove correctness of reduction for $\mathbf{LTL}_{-\mathbf{X}}$ formulas under the concurrency-fair semantics [57].

It is essential to notice that the practical value of a reduction scheme depends on how discriminative the underlying notion of equivalence is. Since $\mathbf{CTL}_{-\mathbf{X}}$ equivalences are more discriminative than those for $\mathbf{LTL}_{-\mathbf{X}}$, partial order reductions preserving $\mathbf{LTL}_{-\mathbf{X}}$ produce smaller models than these for $\mathbf{CTL}_{-\mathbf{X}}$. $\mathbf{ATL}^*_{-\mathbf{X}}$ has even more distinguishing power than $\mathbf{CTL}_{-\mathbf{X}}$. Thus, one can expect that equivalences preserving full $\mathbf{ATL}^*_{-\mathbf{X}}$ ($\mathbf{ATL}^*$ without the next step operator X) would be very discriminative, and result in very inefficient reductions. Aware of this and motivated by practical applications, we look for *subsets of* $\mathbf{ATL}^*_{-\mathbf{X}}$ *for which the most efficient known partial order reduction methods* (i.e., those for $\mathbf{LTL}_{-\mathbf{X}}$) *can be applied.*

In what follows, we show that the reductions for $\mathbf{LTL}_{-\mathbf{X}}$ can be adapted to $\mathbf{sATL}^*_{\mathrm{ir}}$, both with and without the **CF** assumption. We begin by introducing the relevant notions of equivalence (Sections 5.1 and 5.2). Then, we propose conditions on reduced models that preserve the equivalences (Sections 5.3 and 5.4). Finally, we present algorithms for POR and show their correctness (Section 6).

Interestingly, it turns out that our approach does not apply to $\mathbf{sATL}^*_{\mathrm{Ir}}$, cf. Section 6.3. This suggests that **ATL** with imperfect information, besides conceptual advantage, can possibly offer some technical benefits over **ATL** with perfect information.

5.1 Stuttering Equivalences

Let M be a model, $M' \subseteq M$, and $PV \subseteq \mathcal{PV}$ a subset of propositions. Stuttering equivalence says that two paths can be divided into corresponding finite segments, each satisfying exactly the same propositions. Stuttering path equivalence[3] requires two models to always have stuttering-equivalent paths.

Definition 5.1 (Stuttering equivalence [23]). Two paths $\pi \in \Pi_M(\iota)$ and $\pi' \in \Pi_{M'}(\iota)$ are *stuttering equivalent*, denoted $\pi \equiv_s \pi'$, if there exists a partition $B_0 = (\pi[0], \dots, \pi[i_1 - 1])$, $B_1 = (\pi[i_1], \dots, \pi[i_2 - 1])$, ... of the states of π, and an analogous partition $B'_0, B'_1, \dots$ of the states of π', such that for each $j \geq 0 : B_j$ and B'_j are nonempty and finite, and $V(g) \cap PV = V'(g') \cap PV$ for every $g \in B_j$ and $g' \in B'_j$.

Definition 5.2 (Stuttering path equivalence [23]). Models M and M' are *stuttering path equivalent*, denoted $M \equiv_s M'$ if for each path $\pi \in \Pi_M(\iota)$, there is a path $\pi' \in \Pi_{M'}(\iota)$ such that $\pi \equiv_s \pi'$.[4]

Theorem 5.3 ([23]). *If $M \equiv_s M'$, then we have $M, \iota \models \varphi$ iff $M', \iota' \models \varphi$, for any $\mathbf{LTL}_{-\mathbf{X}}$ formula φ over PV.*

5.2 Independence-Based Equivalences

Partial order reductions for concurrency-fair $\mathbf{LTL}_{-\mathbf{X}}$ are based on *Mazurkiewicz traces* [53–55]. Consider two finite sequences of actions $w, w' \in Act^*$. We say that $w \sim_I w'$ iff $w = w_1abw_2$ and $w' = w_1baw_2$, for some $w_1, w_2 \in Act^*$ and $(a, b) \in I_\emptyset$. Let $\equiv_I$ be the reflexive and transitive closure of $\sim_I$. By (finite) traces we mean the equivalence classes of $\equiv_I$, denoted by $[w]_{\equiv_I}$.

Let $v, v' \in Act^\omega$, and let $Pref(v)$ denote the set of the finite prefixes of v. Now, $v \leq_I v'$ iff $\forall u \in Pref(v) \exists \hat{u} \in Pref(v) \exists u' \in Pref(v')(u \in Pref(\hat{u}) \wedge \hat{u} \equiv_I u')$. That is, each finite prefix of v can be extended to a permutation (under commuting adjacent independent actions) of some prefix of v'. Moreover, let $v \equiv^\omega_I v'$ iff $v \leq_I v'$ and $v' \leq_I v$. Infinite traces are defined as equivalence classes of $\equiv^\omega_I$, denoted by $[v]_{\equiv^\omega_I}$.

Theorem 5.4 ([61]). *Let M be a model. If $\pi, \pi' \in \Pi_M(\iota)$ such that $Act(\pi) \equiv^\omega_I Act(\pi')$, then $\pi \equiv_s \pi'$.*

Thus, paths over representatives of the same infinite trace cannot be distinguished by any $\mathbf{LTL}_{-\mathbf{X}}$ formula over PV. Note that Mazurkiewicz traces preserve **CF**, i.e., if $\pi \in \Pi^{CF}_M(\iota)$, then for each π' such that $Act(\pi) \equiv^\omega_I Act(\pi')$ we have $\pi' \in \Pi^{CF}_M(\iota)$.

5.3 Preserving Traces for $\mathbf{sATL}^*_{\mathrm{irF}}$

Rather than generating the full model $M = IIS(S)$, one can generate a reduced model M' satisfying the following property:

$$\textbf{AE-CF}: \ (\forall \pi \in \Pi^{CF}_M(\iota))(\exists \pi' \in \Pi^{CF}_{M'}(\iota)) Act(\pi) \equiv^\omega_I Act(\pi').$$

Then, M' preserves the $\mathbf{LTL}_{-\mathbf{X}}$ formulas under **CF** over PV [61]. We will now prove that this also works for $\mathbf{sATL}^*_{\mathrm{irF}}$.

We first show that each set $out_M(g, \sigma_A)$ is trace-complete in the sense that with each path π such that $Act(\pi) = w$, it contains a path over any $w' \in [w]_{\equiv^\omega_I}$.

Lemma 5.5. *Let $\pi \in out_M(\iota, \sigma_A)$ and $Act(\pi) = w$. Then, $\forall w' \in [w]_{\equiv^\omega_I} \exists \pi' \in out_M(\iota, \sigma_A)$ such that $Act(\pi') = w'$.*

Proof. Let M' be obtained from M by fixing $P_i(l_i) = \{\sigma_i(l_i)\}$ for each $i \in A, l_i \in L_i$, and pruning the transitions accordingly. The set of paths $\Pi_{M'}(\iota)$ of M' must be trace-complete [61]. But $\Pi_{M'}(\iota) = out_M(\iota, \sigma_A)$, which ends the proof. □

The above lemma implies the following.

Lemma 5.6. *Let M be a model and M' its submodel satisfing the property **AE-CF**. Then, for each* ir*-strategy σ_A, $\forall \pi \in out^{CF}_M(\iota, \sigma_A)$ $\exists \pi' \in out^{CF}_{M'}(\iota, \sigma_A)$ such that $Act(\pi) \equiv^\omega_I Act(\pi')$.*

Proof. Assume that $\pi \in out^{CF}_M(\iota, \sigma_A)$. Then there is $\pi' \in \Pi^{CF}_{M'}(\iota)$ such that $Act(\pi) \equiv^\omega_I Act(\pi')$ (by **AE-CF**). Since M' is a submodel

[3] The property is usually called *stuttering trace equivalence*. We opt for a slightly different name to avoid confusion with Mazurkiewicz traces, also used in this paper.

[4] Typically, the definition contains also the symmetric condition which in our case always holds for M and its submodel M', as $\Pi_{M'}(\iota) \subseteq \Pi_M(\iota)$.

of M, we have that $\pi' \in \Pi_M^{CF}(\iota)$. This implies that $\pi' \in out_M^{CF}(\iota, \sigma_A)$ by Lemma 5.5. Since $\pi' \in \Pi_{M'}^{CF}(\iota)$ by Definition 3.4, we obtain that $\pi' \in out_{M'}^{CF}(\iota, \sigma_A)$, which together with the fact that $Act(\pi) \equiv_I^{\omega} Act(\pi')$ completes the proof. □

Theorem 5.7. *Let M be a model and M' its submodel satisfying **AE-CF**. For each* $\mathbf{sATL}^*_{\mathrm{irF}}$ *formula φ over PV we have:*

$$M, \iota \models_{\mathrm{irF}} \varphi \quad \textit{iff} \quad M', \iota' \models_{\mathrm{irF}} \varphi.$$

Proof. Proof by induction on the structure of φ. We show the case $\varphi = \langle\langle A \rangle\rangle \gamma$. The cases for $\neg, \wedge$ are straightforward.

(⇒) Follows from the fact that for each σ_A we have $out_{M'}^{CF}(\iota, \sigma_A) = out_M^{CF}(\iota, \sigma_A) \cap \Pi_{M'}^{CF}(\iota)$, so $out_{M'}^{CF}(\iota, \sigma_A) \subseteq out_M^{CF}(\iota, \sigma_A)$.

(⇐) Assume that $M', \iota \models_{\mathrm{irF}} \langle\langle A \rangle\rangle \gamma$. From the semantics, there is an ir-joint strategy σ_A such that for each $\pi \in out_{M'}^{CF}(\iota, \sigma_A)$ we have $M', \pi \models_{\mathrm{irF}} \gamma$. In order to prove the thesis, we show that for each $\pi \in out_M^{CF}(\iota, \sigma_A) \setminus out_{M'}^{CF}(\iota, \sigma_A)$ we have $M, \pi \models_{\mathrm{irF}} \gamma$. It follows from Lemma 5.6 and Theorem 5.4 that for each $\pi \in out_M^{CF}(\iota, \sigma_A)$ there is $\pi' \in out_{M'}^{CF}(\iota, \sigma_A)$ such that $\pi \equiv_s \pi'$. So, $M', \pi' \models_{\mathrm{irF}} \gamma$ implies that $M, \pi \models_{\mathrm{irF}} \gamma$. Thus, we can conclude that $M, \iota \models_{\mathrm{irF}} \langle\langle A \rangle\rangle \gamma$. □

5.4 Stuttering Equivalence without CF

The method based on Mazurkiewicz traces works well for $\mathbf{sATL}^*_{\mathrm{irF}}$, and we will present an algorithm generating reduced models that satisfy condition **AE-CF** in Section 6. The same cannot be easily applied to the semantics without fairness. In particular, it is unclear how to generate reduced models that satisfy the analogue of **AE-CF** in all paths. However, a similar result can be obtained through stuttering equivalence, based on the following structural property:

$$\mathbf{AE}_A\colon \forall \sigma_A \in \Sigma_A^{\mathrm{ir}} \forall \pi \in out_M(\iota, \sigma_A)\ \exists \pi' \in out_{M'}(\iota, \sigma_A)\colon \pi \equiv_s \pi'$$

Theorem 5.8. *Let $A \subseteq \mathcal{A}$, and let M' be a submodel of M satisfying $\mathbf{AE}_A$. For each* $\mathbf{sATL}^*_{\mathrm{ir}}$ *formula φ over PV, that refers only to coalitions $\hat{A} \subseteq A$:* $\quad M, \iota \models_{\mathrm{ir}} \varphi \quad$ *iff* $\quad M', \iota' \models_{\mathrm{ir}} \varphi$.

Proof. Proof by induction on the structure of φ. We show the case $\varphi = \langle\langle \hat{A} \rangle\rangle \gamma$. The cases for $\neg, \wedge$ are straightforward.

Notice that $out_{M'}(\iota, \sigma_{\hat{A}}) \subseteq out_M(\iota, \sigma_{\hat{A}})$, which together with $\mathbf{AE}_A$ implies that the sets $out_M(\iota, \sigma_{\hat{A}})$ and $out_{M'}(\iota, \sigma_{\hat{A}})$ are stuttering path equivalent. So, the thesis follows from Theorem 5.3. □

Thus, we have proved that the structural conditions **AE-CF** and $\mathbf{AE}_A$ are sufficient to obtain correct reductions with and without fairness (Theorems 5.7 and 5.8). We will discuss algorithms that generate such reduced models in Section 6.

6 ALGORITHMS FOR POR

As mentioned above, the idea of model checking with POR is to reduce the size of models while preserving satisfaction for a class of formulas. Traditionally, the reduction algorithm is based either on depth-first-search (DFS, see [31]), or on double-depth-first-search (DDFS [24]). In this paper, we use the former.

6.1 DFS Algorithm

In the following, the stack represents a path $\pi = g_0 a_0 g_1 a_1 \cdots g_n$ that is currently being visited. For the top element of the stack g_n the following three operations are computed in a loop:

(1) Identify the set $en(g_n) \subseteq Act$ of enabled actions.
(2) Heuristically select a subset $E(g_n) \subseteq en(g_n)$ of possible actions (see Section 6.2).
(3) For any action $a \in E(g_n)$, compute the successor state g' such that $g_n \xrightarrow{a} g'$, and add g' to the stack thereby generating the path $\pi' = g_0 a_0 g_1 a_1 \cdots g_n a g'$. Recursively proceed to explore the submodel originating at g'.
(4) Remove g_n from the stack.

The algorithm begins with the stack comprising of the initial state of $M = IIS(S)$, and terminates when the stack is empty. Notice that the model generated by the algorithm must be a submodel of the M. Moreover, it is generated *directly from the AMAS S, without ever generating the full model M*. Finally, the size of the reduced model crucially depends on the ratio $E(g)/en(g)$. The choice of $E(g)$ is discussed in the next subsection.

6.2 Heuristics for $\mathbf{sATL}^*_{\mathrm{irF}}$ and Subsets of $\mathbf{sATL}_{\mathrm{ir}}$

Let $A \subseteq \mathcal{A}$. The conditions **C1** − **C3** below, inspired by [23], define a heuristics for a selection of $E(g) \subseteq en(g)$ in the algorithm of Sect. 6.1.

- **C1** Along each path π in M that starts at g, each action that is dependent on an action in $E(g)$ cannot be executed in π without an action in $E(g)$ is executed first in π. Formally, $\forall \pi \in \Pi_M(g)$ such that $\pi = g_0 a_0 g_1 a_1 \ldots$ with $g_0 = g$, and $\forall b \in Act$ such that $(b, c) \notin I_A$ for some $c \in E(g)$, if $a_i = b$ for some $i \geq 0$, then $a_j \in E(g)$ for some $j < i$.
- **C2** If $E(g) \neq en(g)$, then $E(g) \subseteq Invis_A$.
- **C3** For every cycle in M' there is at least one node g in the cycle for which $E(g) = en(g)$, i.e., for which all the successors of g are expanded.

Theorem 6.1. *Let $M = IIS(S)$, and $M' \subseteq M$ be the reduced model generated by DFS with the choice of $E(g')$ for $g' \in St'$ given by conditions* **C1, C3** *and the independence relation I_A, where $A = \emptyset$. Then, M' satisfies* ***AE-CF****.*

Proof. See [61, Theorem 3.3]. □

Theorem 6.2. *Let $A \subseteq \mathcal{A}$, $M = IIS(S)$, and $M' \subseteq M$ be the reduced model generated by DFS with the choice of $E(g')$ for $g' \in St'$ given by conditions* **C1, C2, C3** *and the independence relation I_A. Then, M' satisfies $\mathbf{AE}_A$.*

Proof. Although the setting is slightly different, it can be shown similarly to [23, Theorem 12] that the conditions **C1, C2, C3** guarantee that the models M and M' are stuttering path equivalent. More precisely, for each path $\pi = g_0 a_0 g_1 a_1 \cdots$ with $g_0 = \iota$ in M there is a stuttering equivalent path $\pi' = g_0' a_0' g_1' a_1' \cdots$ with $g_0' = \iota$ in M' such that $Act(\pi)|_{Vis_A} = Act(\pi')|_{Vis_A}$, i.e., π and π' have the same maximal sequence of visible actions for A.

To show that M' satisfies $\mathbf{AE}_A$, consider an ir-joint strategy σ_A and $\pi \in out_M(\iota, \sigma_A)$. Since $M \equiv_s M'$, we have that there is $\pi' \in \Pi_{M'}(\iota)$ such that $\pi \equiv_s \pi'$ and $Act(\pi)|_{Vis_A} = Act(\pi')|_{Vis_A}$. Since $Act_i \subseteq Vis_A$ for each $i \in A$, the same sequence of actions of each Act_i is executed in π and π'. Thus, by the generalization of Lemma 3.7 to ir-joint strategies we get $\pi' \in out_M(\iota, \sigma_A)$. So, by Lemma 3.6 we have $\pi' \in out_M(\iota, \sigma_A)$. □

Thus, we have obtained a general method of POR for fragments of $\mathbf{ATL}^*$ with imperfect information. The method is in fact a reformulation of the reduction for $\mathbf{LTL}_{-\mathbf{X}}$. This has at least two welcome implications. First, the actual reductions are likely to be substantial – much more than one would expect with the expressivity of $\mathbf{sATL}^*$. Secondly, one can reuse or adapt existing algorithms and tools performing reductions for $\mathbf{LTL}_{-\mathbf{X}}$. Algorithms generating reduced models, in which the choice of $E(g)$ is given by **C1, C2, C3** or **C1, C3** can be found for instance in [23, 31, 49, 57, 58, 62].

6.3 Bad News for Agents with Perfect Information

Here, we briefly show that the adaptation of $\mathbf{LTL}_{-\mathbf{X}}$ reduction does not work for $\mathbf{sATL}^*$ with memoryless perfect information. We begin with a counterexample to Lemma 5.5 which was essential to our formal construction (Example 6.3). Then, we show that the whole method does not preserve formulas of $\mathbf{sATL}^*_{\mathrm{Ir}}$ (Example 6.4).

Example 6.3. Consider the MAS composed of two agents $\{1,2\}$ such that: $L_1 = \{l_1^1, l_1^2\}$, $L_2 = \{l_2^1, l_2^2\}$, $Act_1 = \{\epsilon, a\}$, $Act_2 = \{\epsilon, b\}$, $P_1(l_1^1) = \{a, \epsilon\}$, $P_1(l_1^2) = \{\epsilon\}$, $P_2(l_2^1) = \{b\}$, $P_2(l_2^2) = \{\epsilon\}$, and $T_1(l_1^1, a) = l_1^2$, $T_2(l_2^1, b) = l_2^2$.

Define an Ir-strategy $\sigma_{\{1,2\}}$ as follows: $\sigma_1(l_1^1, l_2^1) = a$, $\sigma_1(l_1^1, l_2^2) = \sigma_1(l_1^2, l_2^2) = \epsilon$; $\sigma_2(l_1^1, l_2^1) = \sigma_2(l_1^2, l_2^1) = b$, $\sigma_2(l_1^2, l_2^2) = \epsilon$. It is easy to see that $out((l_1^1, l_2^1), \sigma_{\{1,2\}})$ is not trace complete. Note that $(a, b) \in I$, but while $out((l_1^1, l_2^1), \sigma_{\{1,2\}})$ contains the path over $ab(\epsilon)^\omega$, it does not contain any path over $ba(\epsilon)^\omega$.

Example 6.4. Consider formula $\langle\langle c\rangle\rangle(\mathrm{F}\,\mathsf{in}_1 \wedge \mathrm{F}\,\mathsf{in}_2)$, interpreted with the Ir semantics. Clearly, the formula holds in the TGC model in Figure 2a, but not in the reduced model in Figure 2b.

7 HOW BIG IS THE GAIN?

The efficiency of our method follows from the efficiency of partial order reductions for $\mathbf{LTL}_{-\mathbf{X}}$, which has been documented in many papers [49, 56, 60]. We refer to those papers for experimental results, and present here only a quick estimation of the savings that are obtained for the Trains and Controller scenario from Section 2. Let TGC_n be the asynchronous MAS consisting of the controller c and n trains $(t_1, \ldots, t_n)$. Take $PV = \{\mathsf{in}_1, \ldots, \mathsf{in}_n\}$, and let $\mathsf{in}_i \in V(g)$ iff $g^i = T$. That is, in_i holds iff train t_i is in the tunnel.

Note that each action of the controller changes one of the in_i variables. Hence, all the actions of c are visible. It is easy to check that both variants of the POR algorithm from Section 6 generate the reduced model M'_n in Figure 2b. For instance, in the global state (G, A, W), two transitions are enabled: b_1 and a_3. The set $\{a_3\}$ satisfies conditions **C1, C2, C3**, whereas b_1 is a visible transition. Thus, state (R, A, T) is not visited. Similarly, in (G, W, A), transitions a_1 and b_3 are enabled. The set $\{b_3\}$ satisfies conditions **C1, C2, C3**, whereas a_1 is a visible transition. Therefore, state (R, T, A) is not visited.

By Theorems 6.1 and 5.7, the reduced model M'_n satisfies exactly the same $\mathbf{sATL}^*_{\mathrm{irF}}$ formulas over PV as $IIS(TGC_n)$. Moreover, by Theorems 6.2 and 5.8, M'_n and $IIS(TGC_n)$ satisfy the same formulas of $\mathbf{sATL}^*_{\mathrm{irF}}$ using only the strategic operators $\langle\langle c\rangle\rangle, \langle\langle\emptyset\rangle\rangle$. So, for example, one can-model check formula $\langle\langle c\rangle\rangle\mathrm{G}\neg\mathsf{in}_1$ in M'_n instead of TGC_n, and get the same output.

How big is the gain? Quoting the estimates from [48, 49], the size of the full state space is $|St_{IIS(TGC_n)}| \geq 2^{n+1}$, while the size of the reduced model is $|St_{M'_n}| = 2n + 1$. Thus, the reduced state space is *exponentially smaller* than the size of the full model. Of course, such optimistic results are by no means guaranteed. For many AMAS, the reduction may remove a smaller fraction of states. Still, it is important to note that the complexity of $\mathbf{ATL}_{\mathrm{ir}}$ model checking is **NP**-hard *in the size of the model* (and not the size of the representation!), and all the attempts at actual algorithms so far run in exponential time. So, even a linear reduction of the state space is likely to produce an exponential improvement of the performance.

8 CONCLUSIONS AND FUTURE WORK

Many important properties of multi-agent systems are underpinned by the ability of some agents (or groups) to achieve a given goal. In this paper, we propose a general semantics of strategic ability for asynchronous MAS, and study the model checking problem for relevant subsets of alternating-time temporal logic. We concentrate on imperfect information strategies, and consider two semantic variants: one looking at all the infinite executions of strategies, and the other taking into account only the *fair* execution paths.

The theoretical complexity results follow the same pattern as those for synchronous MAS, though proving them required careful treatment. Consequently, model checking of strategic abilities under imperfect information for asynchronous systems is as hard as in the synchronous case. This makes model reductions essential for practical verification. The most important result of this paper consists in showing that the partial order reduction for $\mathbf{LTL}_{-\mathbf{X}}$ can be almost directly applied to $\mathbf{ATL}_{\mathrm{ir}}$ without nested strategic modalities. The importance of the result stems from the fact that $\mathbf{LTL}_{-\mathbf{X}}$ has relatively weak distinguishing power, and therefore admits strong reductions, clustering paths into relatively few equivalence classes.

Interestingly, it turns out that the scheme does *not* work for $\mathbf{ATL}^*$ with perfect information strategies. Until now, virtually all the results have suggested that verification of strategic abilities is significantly easier for agents with perfect information. Thus, we identify an aspect of verification that might be in favor of imperfect information strategies in some contexts.

The ideas presented in this paper open many exciting paths for future research. We will have a closer look at some alternative semantics for $\mathbf{ATL}_{\mathrm{ir}}$ in asynchronous MAS, including the "deadlock-friendly" semantics and the one based on "subjective" ability. We also plan to extend our method to a larger subset of $\mathbf{ATL}^*$ specifications, a subset of Strategy Logic [12], and to $\mathbf{sATL}^*$ with epistemic operators using possibly techniques reported in [18]. Experimental evaluation of the reductions, on known benchmarks and randomly generated models, is also on the list. Adapting the POR scheme to combinations of strategic and epistemic modalities is another interesting path for future work. Finally, we would like to investigate if our partial order reduction scheme can be combined with the bisimulation-based reduction for $\mathbf{ATL}_{\mathrm{ir}}$, proposed recently in [8].

Acknowledgements. We thank the anonymous reviewers for their insightful comments. W. Jamroga and W. Penczek acknowledge the support of the National Centre for Research and Development Poland (NCBiR) under the PolLux grant VoteVerif (POL-LUX-IV/1/2016).

REFERENCES

[1] P. A. Abdulla, S. Aronis, B. Jonsson, and K. F. Sagonas. 2014. Optimal dynamic partial order reduction. In *The 41st Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL '14, San Diego, CA, USA, January 20-21, 2014*. 373–384.

[2] R. Alur, L. de Alfaro, R. Grossu, T.A. Henzinger, M. Kang, C.M. Kirsch, R. Majumdar, F.Y.C. Mang, and B.-Y. Wang. 2001. jMocha: A Model-Checking Tool that Exploits Design Structure. In *Proceedings of ICSE*. IEEE Computer Society Press, 835–836.

[3] R. Alur, T. Henzinger, F. Mang, S. Qadeer, S. Rajamani, and S. Tasiran. 1998. MOCHA: Modularity in Model Checking. In *Proceedings of CAV (Lecture Notes in Computer Science)*, Vol. 1427. Springer, 521–525.

[4] R. Alur and T. A. Henzinger. 1999. Reactive Modules. *Formal Methods in System Design* 15, 1 (1999), 7–48.

[5] R. Alur, T. A. Henzinger, and O. Kupferman. 1997. Alternating-Time Temporal Logic. In *Proceedings of the 38th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE Computer Society Press, 100–109.

[6] R. Alur, T. A. Henzinger, and O. Kupferman. 2002. Alternating-Time Temporal Logic. *J. ACM* 49 (2002), 672–713. https://doi.org/10.1145/585265.585270

[7] C. Baier and J.-P. Katoen. 2008. *Principles of Model Checking*. MIT Press.

[8] F. Belardinelli, R. Condurache, C. Dima, W. Jamroga, and A.V. Jones. 2017. Bisimulations for Verification of Strategic Abilities with Application to ThreeBallot Voting Protocol. In *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems AAMAS 2017*. IFAAMAS. To appear.

[9] F. Belardinelli, A. Lomuscio, A. Murano, and S. Rubin. 2017. Verification of Broadcasting Multi-Agent Systems against an Epistemic Strategy Logic. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*. 91–97.

[10] F. Belardinelli, A. Lomuscio, A. Murano, and S. Rubin. 2017. Verification of Multiagent Systems with Imperfect Information and Public Actions. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2017, São Paulo, Brazil, May 8-12, 2017*. 1268–1276.

[11] R. Berthon, B. Maubert, and A. Murano. 2017. Decidability Results for ATL* with Imperfect Information and Perfect Recall. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2017, São Paulo, Brazil, May 8-12, 2017*. 1250–1258.

[12] R. Berthon, B. Maubert, A. Murano, S. Rubin, and M. Y. Vardi. 2017. Strategy logic with imperfect information. In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20-23, 2017*. 1–12.

[13] N. Bulling, J. Dix, and W. Jamroga. 2010. Model Checking Logics of Strategic Ability: Complexity. In *Specification and Verification of Multi-Agent Systems*, M. Dastani, K. Hindriks, and J.-J. Meyer (Eds.). Springer, 125–159.

[14] N. Bulling and W. Jamroga. 2011. Alternating Epistemic Mu-Calculus. In *Proceedings of IJCAI-11*. 109–114.

[15] N. Bulling and W. Jamroga. 2014. Comparing Variants of Strategic Ability: How Uncertainty and Memory Influence General Properties of Games. *Journal of Autonomous Agents and Multi-Agent Systems* 28, 3 (2014), 474–518.

[16] S. Busard, C. Pecheur, H. Qu, and F. Raimondi. 2014. Improving the Model Checking of Strategies under Partial Observability and Fairness Constraints. In *Formal Methods and Software Engineering*. Lecture Notes in Computer Science, Vol. 8829. Springer, 27–42. https://doi.org/10.1007/978-3-319-11737-9_3

[17] S. Busard, C. Pecheur, H. Qu, and F. Raimondi. 2015. Reasoning about memoryless strategies under partial observability and unconditional fairness constraints. *Information and Computation* 242 (2015), 128–156. https://doi.org/10.1016/j.ic.2015.03.014

[18] P. Cermák, A. Lomuscio, F. Mogavero, and A. Murano. 2014. MCMAS-SLK: A Model Checker for the Verification of Strategy Logic Specifications. In *Proc. of CAV'14 (Lecture Notes in Computer Science)*, Vol. 8559. Springer, 525–532.

[19] P. Cermák, A. Lomuscio, and A. Murano. 2015. Verifying and Synthesising Multi-Agent Systems against One-Goal Strategy Logic Specifications. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*. 2038–2044.

[20] K. Chatterjee, A. Pavlogiannis, N. Sinha, and K. Vaidya. 2016. Data-centric Dynamic Partial Order Reduction. *CoRR* abs/1610.01188 (2016).

[21] T. Chen, V. Forejt, M. Kwiatkowska, D. Parker, and A. Simaitis. 2013. PRISM-games: A Model Checker for Stochastic Multi-Player Games. In *Proceedings of TACAS (Lecture Notes in Computer Science)*, Vol. 7795. Springer, 185–191.

[22] E.M. Clarke and E.A. Emerson. 1981. Design and Synthesis of Synchronization Skeletons Using Branching Time Temporal Logic. In *Proceedings of Logics of Programs Workshop (Lecture Notes in Computer Science)*, Vol. 131. 52–71.

[23] E. M. Clarke, O. Grumberg, and D. A. Peled. 1999. *Model Checking*. The MIT Press, Cambridge, Massachusetts.

[24] C. Courcoubetis, M. Vardi, P. Wolper, and M. Yannakakis. 1992. Memory-Efficient Algorithms for the Verification of Temporal Properties. *Formal Methods in System Design* 1, 2/3 (1992), 275–288.

[25] M. Dastani and W. Jamroga. 2010. Reasoning about Strategies of Multi-Agent Programs. In *Proceedings of AAMAS2010*. 625–632.

[26] C. Dima, B. Maubert, and S. Pinchinat. 2014. The Expressive Power of Epistemic μ-Calculus. *CoRR* abs/1407.5166 (2014).

[27] C. Dima, B. Maubert, and S. Pinchinat. 2015. Relating Paths in Transition Systems: The Fall of the Modal Mu-Calculus. In *Proceedings of MFCS (Lecture Notes in Computer Science)*, Vol. 9234. Springer, 179–191. https://doi.org/10.1007/978-3-662-48057-1_14

[28] C. Dima and F.L. Tiplea. 2011. Model-checking ATL under Imperfect Information and Perfect Recall Semantics is Undecidable. *CoRR* abs/1102.4225 (2011).

[29] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. 1995. *Reasoning about Knowledge*. MIT Press.

[30] C. Flanagan and P. Godefroid. 2005. Dynamic partial-order reduction for model checking software. In *Proceedings of the 32nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2005, Long Beach, California, USA, January 12-14, 2005*. 110–121.

[31] R. Gerth, R. Kuiper, D. Peled, and W. Penczek. 1999. A Partial Order Approach to Branching Time Logic Model Checking. *Information and Computation* 150 (1999), 132–152.

[32] P. Godefroid. 1991. Using Partial Orders to Improve Automatic Verification Methods. In *Proceedings of the 2nd International Conference on Computer Aided Verification (CAV'90) (ACM/AMS DIMACS Series)*, E. M. Clarke and R. P. Kurshan (Eds.), Vol. 3. 321–340.

[33] P. Godefroid and P. Wolper. 1994. A Partial Approach to Model Checking. *Information and Computation* 110, 2 (1994), 305–326.

[34] D.P. Guelev, C. Dima, and C. Enea. 2011. An alternating-time temporal logic with knowledge, perfect recall and past: axiomatisation and model-checking. *Journal of Applied Non-Classical Logics* 21, 1 (2011), 93–131.

[35] W. van der Hoek and M. Wooldridge. 2002. Tractable multiagent planning for epistemic goals. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'02)*, M. Gini, T. Ishida, C. Castelfranchi, and W. L. Johnson (Eds.). ACM Press, 1167–1174.

[36] X. Huang and R. van der Meyden. 2014. Symbolic Model Checking Epistemic Strategy Logic. In *Proceedings of AAAI*. 1426–1432.

[37] W. Jamroga. 2003. Some Remarks on Alternating Temporal Epistemic Logic. In *Proceedings of Formal Approaches to Multi-Agent Systems (FAMAS 2003)*, B. Dunin-Keplicz and R. Verbrugge (Eds.). 133–140.

[38] W. Jamroga and J. Dix. 2006. Model Checking ATL_{ir} is Indeed Δ_2^P-complete. In *Proceedings of EUMAS'06 (CEUR Workshop Proceedings)*, Vol. 223. CEUR-WS.org.

[39] W. Jamroga, M. Knapik, and D. Kurpiewski. 2017. Fixpoint Approximation of Strategic Abilities under Imperfect Information. In *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems AAMAS 2017*. IFAAMAS.

[40] W. Jamroga and W. van der Hoek. 2004. Agents that Know how to Play. *Fundamenta Informaticae* 63, 2–3 (2004), 185–219.

[41] M. Kacprzak and W. Penczek. 2004. Unbounded Model Checking for Alternating-Time Temporal Logic. In *Proceedings of AAMAS-04*. IEEE Computer Society, 646–653. https://doi.org/10.1109/AAMAS.2004.10089

[42] V. Kahlon, C. Wang, and A. Gupta. 2009. Monotonic Partial Order Reduction: An Optimal Symbolic Partial Order Reduction Technique. In *Computer Aided Verification, 21st International Conference, CAV 2009, Grenoble, France, June 26 - July 2, 2009. Proceedings*. 398–413.

[43] I. Kokkarinen, D. Peled, and A. Valmari. 1997. Relaxed Visibility Enhances Partial Order Reductions. In *Proceedings of the 9th International Conference on Computer Aided Verification (CAV'97) (LNCS)*, Vol. 1254. Springer-Verlag, 328–340.

[44] I. Konnov, H. Veith, and J. Widder. 2015. SMT and POR Beat Counter Abstraction: Parameterized Model Checking of Threshold-Based Distributed Algorithms. In *Computer Aided Verification - 27th International Conference, CAV 2015, San Francisco, CA, USA, July 18-24, 2015, Proceedings, Part I*. 85–102.

[45] L. M. Kristensen and A. Valmari. 2000. Improved Question-Guided Stubborn Set Methods for State Properties. In *Proceedings of the 21st International Conference on Applications and Theory of Petri Nets (ICATPN'00) (LNCS)*, Vol. 1825. Springer-Verlag, 282–302.

[46] O. Kupferman, M.Y. Vardi, and P. Wolper. 2000. An automata-theoretic approach to branching-time model checking. *Journal of the ACM* 47, 2 (2000), 312–360.

[47] F. Laroussinie, N. Markey, and G. Oreiby. 2008. On the Expressiveness and Complexity of ATL. *Logical Methods in Computer Science* 4 (2008), 7.

[48] A. Lomuscio, W. Penczek, and H. Qu. 2010. Partial order reductions for model checking temporal epistemic logics over interleaved multi-agent systems. In *9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010), Toronto, Canada, May 10-14, 2010, Volume 1-3*. 659–666.

[49] A. Lomuscio, W. Penczek, and H. Qu. 2010. Partial Order Reductions for Model Checking Temporal-epistemic Logics over Interleaved Multi-agent Systems. *Fundam. Inform.* 101, 1-2 (2010), 71–90.

[50] A. Lomuscio, H. Qu, and F. Raimondi. 2015. MCMAS: An Open-Source Model Checker for the Verification of Multi-Agent Systems. *International Journal on Software Tools for Technology Transfer* (2015). https://doi.org/10.1007/s10009-015-0378-x Availabe online.

[51] A. Lomuscio and F. Raimondi. 2006. MCMAS : A Model Checker for Multi-Agent Systems. In *Proceedings of TACAS (Lecture Notes in Computer Science)*, Vol. 4314.

Springer, 450–454.
[52] V. Malvone, A. Murano, and L. Sorrentino. 2017. Hiding Actions in Multi-Player Games. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2017, São Paulo, Brazil, May 8-12, 2017.* 1205–1213.
[53] A. Mazurkiewicz. 1977. Concurrent program schemes and their interpretations. *DAIMI Report Series* 6, 78 (1977).
[54] A. Mazurkiewicz. 1986. Trace Theory. In *Advances in Petri Nets 1986*. LNCS, Vol. 255. Springer-Verlag, 279–324.
[55] A. Mazurkiewicz. 1988. Basic Notions of Trace Theory. In *Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency*. LNCS, Vol. 354. Springer-Verlag, 285–363.
[56] José Vander Meulen and Charles Pecheur. 2011. Combining Partial-Order Reduction and Symbolic Model Checking to Verify LTL Properties. In *NASA Formal Methods - Third International Symposium, NFM 2011, Pasadena, CA, USA, April 18-20, 2011. Proceedings.* 406–421.
[57] D. Peled. 1993. All From One, One For All: On Model Checking Using Representatives. In *Proceedings of the 5th International Conference on Computer Aided Verification (LNCS 697)*. Springer-Verlag, 409–423.
[58] D. Peled. 1994. Combining Partial Order Reductions with On-the-fly Model-Checking. In *Proceedings of the 6th International Conference on Computer Aided Verification (LNCS 818)*. Springer-Verlag, 377–390.
[59] D. Peled. 1996. Partial Order Reductions: Model Checking Using Representatives. In *Proceedings of the 21st International Symposium on Mathematical Foundations of Computer Science (MFCS'96) (LNCS)*, Vol. 1113. Springer-Verlag, 93–112.
[60] D. Peled. 1998. Ten Years of Partial-Order Reductions. In *Proceedings of the 10th International Conference on Computer Aided Verification (CAV'98) (LNCS)*, Vol. 1427. Springer-Verlag, 17–28.
[61] Doron A. Peled. 1996. Combining Partial Order Reductions with On-the-Fly Model-Checking. *Formal Methods in System Design* 8, 1 (1996), 39–64.
[62] W. Penczek, M. Szreter, R. Gerth, and R. Kuiper. 2000. Improving Partial Order Reductions for Universal Branching Time Properties. *Fundamenta Informaticae* 43 (2000), 245–267.
[63] J. Pilecki, M.A. Bednarczyk, and W. Jamroga. 2014. Synthesis and Verification of Uniform Strategies for Multi-Agent Systems. In *Proceedings of CLIMA XV (Lecture Notes in Computer Science)*, Vol. 8624. Springer, 166–182.
[64] B. Puchala. 2010. Asynchronous Omega-Regular Games with Partial Information. In *Proceedings of MFCS*. 592–603. https://doi.org/10.1007/978-3-642-15155-2_52
[65] Ph. Schnoebelen. 2003. The Complexity of Temporal Model Checking. In *Advances in Modal Logics, Proceedings of AiML 2002.* World Scientific.
[66] P. Y. Schobbens. 2004. Alternating-Time Logic with Imperfect Recall. *Electronic Notes in Theoretical Computer Science* 85, 2 (2004), 82–93.