Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning
Main Track

Strategic Abilities of Asynchronous Agents: Semantic Side Effects and How to Tame Them

Wojciech Jamroga[1,2] , **Wojciech Penczek**[1] , **Teofil Sidoruk**[1,3]
[1]Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland
[2]Interdisciplinary Centre on Security, Reliability and Trust, SnT, University of Luxembourg
[3]Faculty of Mathematics and Information Science, Warsaw University of Technology
{jamroga, penczek, t.sidoruk}@ipipan.waw.pl

Abstract

Recently, we have proposed a framework for verification of agents' abilities in asynchronous multi-agent systems (MAS), together with an algorithm for automated reduction of models. The semantics was built on the modeling tradition of distributed systems. As we show here, this can sometimes lead to counterintuitive interpretation of formulas when reasoning about the outcome of strategies. First, the semantics disregards finite paths, and yields unnatural evaluation of strategies with deadlocks. Secondly, the semantic representations do not allow to capture the asymmetry between proactive agents and the recipients of their choices. We propose how to avoid the problems by a suitable extension of the representations and change of the execution semantics for asynchronous MAS. We also prove that the model reduction scheme still works in the modified framework.

1 Introduction

Alternating-time temporal logic $\mathbf{ATL}^*$ (Alur, Henzinger, and Kupferman 2002; Schobbens 2004) is probably the most popular logic to describe interaction in multi-agent systems. Formulas of $\mathbf{ATL}^*$ allow to express statements about what agents (or groups of agents) can achieve. For example, $\langle\!\langle taxi \rangle\!\rangle \mathrm{G}\, \neg \mathsf{fatality}$ says that the autonomous cab can drive in such a way that nobody is ever killed, and $\langle\!\langle taxi, passg \rangle\!\rangle \mathrm{F}\, \mathsf{destination}$ expresses that the cab and the passenger have a joint strategy to arrive at the destination, no matter what any other agents do. Such statements allow to express important functionality and safety requirements in a simple and intuitive way. Moreover, the provide input to algorithms and tools for verification, that have been in constant development for over 20 years (Alur et al. 1998; Chen et al. 2013; Busard et al. 2014; Huang and van der Meyden 2014; Cermák et al. 2014; Lomuscio, Qu, and Raimondi 2017; Cermák, Lomuscio, and Murano 2015; Belardinelli et al. 2017b; Belardinelli et al. 2017a; Jamroga et al. 2019; Kurpiewski et al. 2021).

Asynchronous semantics and partial-order reduction. The semantics of strategic logics is traditionally based on synchronous concurrent game models. However, many real-life systems are inherently asynchronous or can be more conveniently modeled as asynchronous.

We have recently proposed how to adapt the semantics of $\mathbf{ATL}^*$ to asynchronous MAS (Jamroga et al. 2018). We also showed that the technique of *partial order reduction (POR)* (Peled 1993; Peled 1994; Lomuscio, Penczek, and Qu 2010b) can be adapted to verification of strategic abilities in asynchronous MAS. In fact, the (almost 30 years old) POR for linear time logic $\mathbf{LTL}$ can be taken off the shelf and applied to a significant part of $\mathbf{ATL}^*_{\mathrm{ir}}$, the variant of $\mathbf{ATL}^*$ based on strategies with imperfect information and imperfect recall. This is very important, as the practical verification of asynchronous systems is often impossible due to the state- and transition-space explosion resulting from interleaving of local transitions. POR allows for a significant, sometimes even exponential, reduction of the models.

Semantic side effects. While the result is appealing, there is a sting in its tail: the $\mathbf{ATL}^*$ semantics in (Jamroga et al. 2018) leads to counterintuitive interpretation of strategic properties. First, it disregards finite paths, and evaluates some intuitively losing strategies as winning (and vice versa). Secondly, it provides a flawed interpretation of the *concurrency fairness* assumption. Thirdly, the representations and their execution semantics do not allow to capture the asymmetry between the agents that control which synchronization branch will be taken, and those influenced by their choices. We tentatively indicated some of the problems in the extended abstract (Jamroga, Penczek, and Sidoruk 2021). In this paper, we demonstrate them carefully, and propose how they can be avoided.

Contribution. Our contribution is threefold. First, we discuss in detail the semantic side effects of adding strategic reasoning on top of classical models of concurrent systems (Priese 1983). We identify the reasons, and demonstrate the problematic phenomena on simple examples. Secondly, we show how to avoid these pitfalls by extending the class of representations and slightly changing the execution semantics of strategies. Specifically, we add "silent" ϵ-transitions in the models and on outcome paths of strategies, and allow for nondeterministic choices in the agents' repertoires. We also identify a family of fairness-style conditions, suitable for the interaction of proactive and reactive agents. No less importantly, we prove that partial order reduction is still correct in the modified framework.

Motivation. The variant of $\mathbf{ATL}^*$ for asynchronous systems in (Jamroga et al. 2018) was proposed mainly as a framework for formal verification. This was backed by the results showing that it submits to partial order reduction. However, a verification framework is only useful if it allows to specify requirements in an intuitive way, so that the property we *think* we are verifying is indeed *the one being verified*. In this paper, we show that this was not the case. We also propose how to overcome the problems without spoiling the efficient reduction scheme. The solutions are not merely technical. In fact, they lead to a better understanding of how strategic activity influences the overall behavior of the system, and how it should be integrated with the traditional models of asynchronous interaction.

2 Models of Multi-agent Systems

We first recall the models of asynchronous interaction in MAS, proposed in (Jamroga et al. 2018) and inspired by (Priese 1983; Lomuscio, Penczek, and Qu 2010b).

Asynchronous multi-agent systems. In logical approaches to MAS, one usually assumes synchronous actions of all the agents (Alur, Henzinger, and Kupferman 2002; Schobbens 2004). However, many agent systems are inherently asynchronous, or it is useful to model them without assuming precise timing relationships between the actions of different agents. Such a system can be conveniently represented with a set of automata that execute asynchronously by interleaving local transitions, and synchronize their moves whenever a shared event occurs. The idea is to represent the behavior of each agent by a finite automaton where the nodes and transitions correspond, respectively, to the agent's local states and the events in which it can take part. Then, the global behavior of the system is obtained by the interleaving of local transitions, assuming that, in order for a shared event to occur, all the corresponding agents must execute it in their automata. This motivates the following definition.

Definition 2.1 (Asynchronous MAS). *An* asynchronous multi-agent system (AMAS) S *consists of* n *agents* $\mathbb{A}\mathrm{gt} = \{1, \ldots, n\}$*, each associated with a tuple* $A_i = (L_i, \iota_i, Evt_i, R_i, T_i, \mathcal{PV}_i, V_i)$ *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* events $Evt_i = \{\alpha_i^1, \alpha_i^2, \ldots, \alpha_i^{m_i}\}$*. An agent's* repertoire of choices $R_i : L_i \to 2^{Evt_i} \setminus \{\emptyset\}$ *selects the events available at each local state.* $T_i : L_i \times Evt_i \rightharpoonup L_i$ *is a (partial)* local transition function *such that* $T_i(l_i, \alpha)$ *is defined iff* $\alpha \in R_i(l_i)$*. That is,* $T_i(l, \alpha)$ *indicates the result of executing event* α *in local state* l *from the perspective of agent* i*.*

Let $Evt = \bigcup_{i \in \mathbb{A}\mathrm{gt}} Evt_i$ *be the set of all events, and* $Loc = \bigcup_{i \in \mathbb{A}\mathrm{gt}} L_i$ *be the set of all local states in the system. For each event* $\alpha \in Evt$*,* $Agent(\alpha) = \{i \in \mathbb{A}\mathrm{gt} \mid \alpha \in Evt_i\}$ *is the set of agents which have* α *in their repertoires; events shared by multiple agents are jointly executed by all of them. We assume that each agent* i *in the AMAS is endowed with a disjoint set of its* local propositions $\mathcal{PV}_i$*, and their valuation* $V_i : L_i \to 2^{\mathcal{PV}_i}$*. The overall set of propositions* $\mathcal{PV} = \bigcup_{i \in \mathbb{A}\mathrm{gt}} \mathcal{PV}_i$ *collects all the local propositions.*

As our working example, we use the following scenario.

Example 2.2 (Conference in times of epidemic). *Consider the AMAS in Figure 1, consisting of the Steering Committee Chair (sc), the General Chair (gc), and the Organizing*

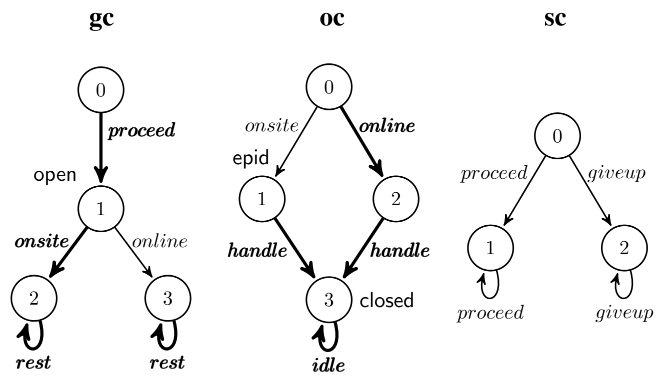

Figure 1: Simple asynchronous MAS: agents gc, oc, and sc. A joint strategy of agents $\{gc, oc\}$ is highlighted.

Committee Chair (oc). Faced with the Covid-19 epidemics, sc *can decide to give up the conference, or send a signal to* gc *to proceed and open the meeting. Then,* gc *and* oc *jointly decide whether the conference will be run on site or online. In the former case, the epidemiologic risk is obviously much higher, indicated by the atomic proposition* epid.

The set of events, the agents' repertoires of choices, and the valuation of atomic propositions can be easily read from the graph. Note that event $proceed$ *is shared by agents* sc *and* gc*, and can only be executed jointly. Similarly,* $onsite$ *and* $online$ *are shared by* gc *and* oc*. All the other events are private, and do not require synchronization.*

Interleaved interpreted systems. To understand the interaction between asynchronous agents, we use the standard execution semantics from concurrency models, i.e., interleaving with synchronization on shared events. To this end, we compose the network of local automata (i.e., AMAS) to a single automaton based on the notions of *global states* and *global transitions*, see below.

Definition 2.3 (Model). *Let* S *be an AMAS with* n *agents. Its* model $IIS(S)$ *extends* S *with: (i) the set of global states* $St \subseteq L_1 \times \ldots \times L_n$*, including the* initial state $\iota = (\iota_1, \ldots, \iota_n)$ *and all the states reachable from* ι *by* T *(see below); (ii) the* global transition function $T : St \times Evt \rightharpoonup St$*, defined by* $T(g_1, \alpha) = g_2$ *iff* $T_i(g_1^i, \alpha) = g_2^i$ *for all* $i \in Agent(\alpha)$ *and* $g_1^i = g_2^i$ *for all* $i \in \mathbb{A}\mathrm{gt} \setminus Agent(\alpha)$*; (iii) the* global valuation *of propositions* $V : St \to 2^{\mathcal{PV}}$*, defined as* $V(l_1, \ldots, l_n) = \bigcup_{i \in \mathbb{A}\mathrm{gt}} V_i(l_i)$*.*

Models, sometimes called *interleaved interpreted systems* (IIS), are used to provide an execution semantics to AMAS. Intuitively, the global states in $IIS(S)$ can be seen as the possible configurations of local states of all the agents. Moreover, the transitions are labeled by events that are simultaneously selected (in the current configuration) by all the agents that have the event in their repertoire.

Example 2.4 (Conference). *The model for the asynchronous MAS of Example 2.2 is shown in Figure 2.*

We say that event $\alpha \in Evt$ is *enabled* at $g \in St$ if $T(g, \alpha) = g'$ for some $g' \in St$. The set of events enabled at

g is denoted by $enabled(g)$. The global transition function is assumed to be serial, i.e., at each $g \in St$ there exists at least one enabled event.

Discussion. This modeling approach is standard in theory of concurrent systems, where it dates back to the early 1980s and the idea of APA Nets (asynchronous, parallel automata nets) (Priese 1983). Note that APA Nets and their models were *not* proposed with causal interpretation in mind. In particular, they were *not* meant to capture the interaction of purposeful agents that freely choose their strategies, but rather a set of reactive components converging to a joint behavior. Despite superficial differences, the same applies to process-algebraic approaches to concurrency, such as CSP (Hoare 1978), CCS (Milner 1980), ACP (Bergstra and Klop 1985), and π-calculus (Milner, Parrow, and Walker 1992).

Definition 2.1 extends that with the repertoire functions from synchronous models of MAS (Lomuscio, van der Meyden, and Ryan 2000; Alur, Henzinger, and Kupferman 2002). Agent i's repertoire lists the events available to i, and is supposed to define the space of i's strategies. As we show further, this is not enough in case of asynchronous MAS.

3 Reasoning About Abilities: ATL*

Alternating-time temporal logic $\mathbf{ATL}^*$ (Alur, Henzinger, and Kupferman 2002; Schobbens 2004) introduces *strategic modalities* $\langle\!\langle A \rangle\!\rangle\gamma$, expressing that agents A can enforce the temporal property γ. A variant for asynchronous MAS was proposed recently (Jamroga et al. 2018). We summarize the main points in this section.

Syntax. Let $\mathcal{PV}$ be a set of propositional variables and $\mathbb{A}\mathrm{gt}$ the set of all agents. The language of $\mathbf{ATL}^*$ is defined as below.

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

where $\mathsf{p} \in \mathcal{PV}$, $A \subseteq \mathbb{A}\mathrm{gt}$, X stands for "next", and U for "strong until". The other Boolean operators and constants are defined as usual. "Release" can be defined as $\gamma_1\,\mathrm{R}\,\gamma_2 \equiv \neg((\neg\gamma_1)\,\mathrm{U}\,(\neg\gamma_2))$. "Eventually" and "always" can be defined as $\mathrm{F}\,\gamma \equiv \mathit{true}\,\mathrm{U}\,\gamma$ and $\mathrm{G}\,\gamma \equiv \mathit{false}\,\mathrm{R}\,\gamma$.

Example 3.1 (Conference)**.** *Formula* $\langle\!\langle sc \rangle\!\rangle\mathrm{F}\,\mathsf{open}$ *expresses that the Steering Chair can enforce that the conference is eventually opened. Moreover, formula* $\langle\!\langle gc, oc \rangle\!\rangle\mathrm{G}\,\neg\mathsf{epid}$ *says that the General Chair and the Organizing Chair have a joint strategy to avoid high epidemiological risk.*

Strategies and outcomes. An *imperfect* ***i****nformation/imperfect* ***r****ecall strategy (*ir*-strategy) for* i is a function $\sigma_i \colon L_i \to Evt_i$ s.t. $\sigma_i(l) \in R_i(l)$ for each $l \in L_i$. We denote the set of such strategies by Σ_i^{ir}. A *collective strategy* σ_A for a coalition $A = (1, \ldots, m) \subseteq \mathbb{A}\mathrm{gt}$ is a tuple of strategies, one per agent $i \in A$. The set of A's collective ir strategies is denoted by Σ_A^{ir}. We will sometimes use $\sigma_A(g) = (\sigma_{a_1}(g), \ldots, \sigma_{a_m}(g))$ to denote the tuple of A's selections at state g.

Example 3.2 (Conference)**.** *A collective strategy for the General Chair and the OC Chair is shown in Figure 1.*

An infinite sequence of global states and events $\pi = g_0\alpha_0 g_1\alpha_1 g_2 \ldots$ is called a *path* if $g_i \xrightarrow{\alpha_i} g_{i+1}$ for every $i \geq 0$. $Evt(\pi) = \alpha_0\alpha_1\alpha_2 \ldots$ is the sequence of events 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 model M starting at g. Intuitively, the outcome of σ_A in g is the set of all the paths that can occur when the agents in A follow σ_A and the agents in $\mathbb{A}\mathrm{gt} \setminus A$ freely choose events from their repertoires. To define it formally, we first refine the concept of an enabled event, taking into account the choices of A in strategy σ_A.

Definition 3.3 (Enabled events)**.** *Let* $A = (1, \ldots, m)$, $g \in St$, *and let* $\overrightarrow{\alpha}_A = (\alpha_1, \ldots, \alpha_m)$ *be a tuple of events such that every* $\alpha_i \in R_i(g^i)$. *That is, every* α_i *can be selected by its respective agent* i *at state* g. *We say that event* $\beta \in Evt$ *is* enabled by $\overrightarrow{\alpha}_A$ at $g \in St$ *iff*

- *for every* $i \in Agent(\beta) \cap A$, *we have* $\beta = \alpha_i$, *and*
- *for every* $i \in Agent(\beta) \setminus A$, *it holds that* $\beta \in R_i(g^i)$.

Thus, β *is enabled by* $\overrightarrow{\alpha}_A$ *if all the agents that "own"* β *can choose* β *for execution, even when* $\overrightarrow{\alpha}_A$ *has been selected by the coalition A. We denote the set of such events by* $enabled(g, \overrightarrow{\alpha}_A)$. *Clearly,* $enabled(g, \overrightarrow{\alpha}_A) \subseteq enabled(g)$.

Example 3.4 (Conference)**.** *Consider state* $g = 000$ *and the choices of agents* $A = \{gc, oc\}$ *shown in Figure 1, i.e.,* $\overrightarrow{\alpha}_A = (proceed, online)$. *The only events enabled by* $\overrightarrow{\alpha}_A$ *are* $proceed$ *and* $giveup$. *Event* $onsite$ *is not enabled because A chose different events for execution;* $online$ *is not enabled because it requires synchronization which is impossible at* 000.

Definition 3.5 (Outcome paths)**.** *The* outcome *of strategy* $\sigma_A \in \Sigma_A^{\mathrm{ir}}$ *in state* $g \in St$ *is the set* $out_M(g, \sigma_A) \subseteq \Pi_M(g)$ *such that* $\pi = g_0\alpha_0 g_1\alpha_1 g_2 \cdots \in out_M(g, \sigma_A)$ *iff* $g_0 = g$, *and* $\forall i \geq 0 \quad \alpha_i \in enabled(\pi[i], \sigma_A(\pi[i]))$.

One often wants to look only at paths that do not consistently ignore agents whose choice is always enabled. Formally, a path π satisfies *concurrency-fairness* (**CF**) if there is no event α enabled in all states of π from $\pi[n]$ on and such that for every α_i actually executed in $\pi[i]$, $i = n, n+1, \ldots$, we have $Agent(\alpha) \cap Agent(\alpha_i) = \emptyset$. We denote the set of all such paths starting at g by $\Pi_M^{\mathbf{CF}}(g)$.

Definition 3.6 (**CF**-outcome)**.** *The* **CF**-outcome *of* $\sigma_A \in \Sigma_A^{\mathrm{ir}}$ *is defined as* $out_M^{\mathbf{CF}}(g, \sigma_A) = out_M(g, \sigma_A) \cap \Pi_M^{\mathbf{CF}}(g)$.

Strategic ability for asynchronous systems. The semantics of $\mathbf{ATL}_{\mathrm{ir}}^*$ in AMAS is defined by the following clause for strategic modalities (Jamroga et al. 2018):

$M, g \models_{\mathrm{ir}} \langle\!\langle A \rangle\!\rangle\gamma$ iff there is a strategy $\sigma_A \in \Sigma_A^{\mathrm{ir}}$ s.t. $out_M(g, \sigma_A) \neq \emptyset$ and, for each path $\pi \in out_M(g, \sigma_A)$, we have $M, \pi \models_{\mathrm{ir}} \gamma$.

The clauses for Boolean and temporal operators are standard. Moreover, the *concurrency-fair semantics* $\models_{\mathrm{ir}}^{\mathbf{CF}}$ is obtained by replacing $out_M(g, \sigma_A)$ with $out_M^{\mathbf{CF}}(g, \sigma_A)$ in the above clause.

Example 3.7 (Conference)**.** *Clearly, formula* $\langle\!\langle gc, oc \rangle\!\rangle\mathrm{G}\,\neg\mathsf{epid}$ *holds in* $(M_{conf}, 000)$, *in both* $\models_{\mathrm{ir}}$ *and* $\models_{\mathrm{ir}}^{\mathbf{CF}}$ *semantics. To see that, fix* $\sigma_{gc}(0) = proceed$ *and*

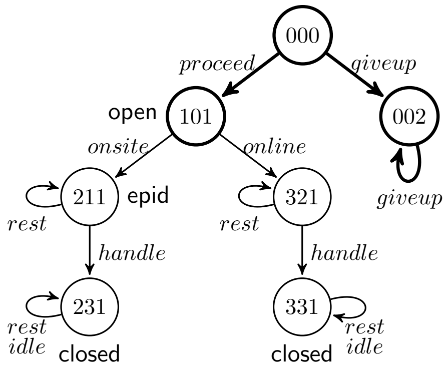

Figure 2: Model M_{conf} for the conference scenario. We highlight the transitions enabled by the strategy in Figure 1, and the resulting reachable states.

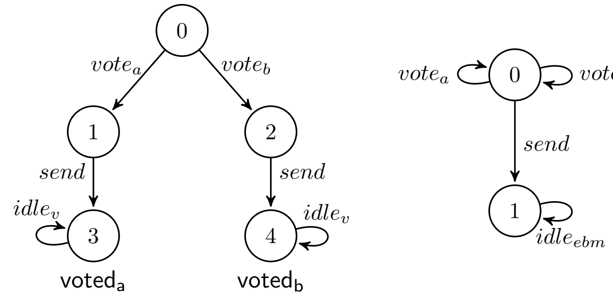

Figure 3: Casting a ballot: voter v (left) and EBM ebm (right)

$\sigma_{gc}(1) = \sigma_{oc}(0) = online$ *in the collective strategy of* $\{gc, oc\}$. *Note also that* $M_{conf}, 000 \models_{\mathrm{ir}} \neg\langle\langle gc, oc\rangle\rangle \mathrm{F}\,\mathsf{closed}$ *because, after executing* $proceed$ *and* $online$ *(or* $onsite$*), event* $rest$ *may be selected forever. On the other hand, such paths are not concurrency-fair, and thus* $M_{conf}, 000 \models^{\boldsymbol{CF}}_{\mathrm{ir}} \langle\langle gc, oc\rangle\rangle \mathrm{F}\,\mathsf{closed}$.

Discussion. Strategic play assumes proactive attitude: the agents in $\langle\langle A\rangle\rangle$ are free to choose *any* available strategy σ_A. This is conceptually consistent with the notion of agency (Bratman 1987). At the same time, it is somewhat at odds with the standard semantics of concurrent processes, where the components cannot stubbornly refuse to synchronize if that is the only way to proceed with a transition. This seems a minor problem, but it is worrying that a strategy can have the empty set of outcomes, and equally worrying that such strategies are treated differently from the other ones. Indeed, as we will show in the subsequent sections, the semantics proposed in (Jamroga et al. 2018) leads to a counterintuitive interpretation of strategic formulas.

4 Semantic Problems and How to Avoid Them

Starting with this section, we describe some problematic phenomena that follow from the straightforward combination of strategic ability with models of concurrent systems, proposed in (Jamroga et al. 2018). We also show how to extend the representations and modify their execution semantics to avoid the counterintuitive interpretation of formulas.

4.1 Deadlock Strategies and Finite Paths

An automata network is typically required to produce no deadlock states, i.e., every global state in its composition must have at least one outgoing transition. Then, all the maximal paths are infinite, and it is natural to refer to only infinite paths in the semantics of temporal operators. In case of AMAS, the situation is more delicate. Even if the AMAS as a whole produces no deadlocks, some strategies might, which makes the interpretation of strategic modalities cumbersome. We illustrate this on the following example.

Example 4.1 (Conference)**.** *Recall the 3-agent AMAS of Figure 1, together with its model* M_{conf} *(Figure 2). Clearly,* M_{conf} *has no deadlock states. Let us now look at the collective strategies of coalition* $\{gc, oc\}$*, with agent* sc *serving as the opponent. It is easy to see that the coalition has no way to prevent the opening of the conference, i.e., it cannot prevent the system from reaching state* 101*. However, the strategy depicted in Figure 1 produces only one* infinite *path:* $(000\,giveup\,002\,giveup\,\ldots)$*. Since the* $\mathbf{ATL}^*$ *semantics in Section 3 disregards finite paths, we get that* $M_{conf}, 000 \models \langle\langle gc, oc\rangle\rangle \mathrm{G}\,\neg\mathsf{open}$*, which is counterintuitive.*

Things can get even trickier. In particular, the outcome of a strategy can be empty – in fact, it may even happen that a coalition has only strategies with empty outcomes.

Example 4.2 (Voting)**.** *Consider the AMAS in Figure 3 that depicts a simple voting scenario. A voter* v *can fill in an electronic ballot with a vote for candidate* a *or* b*, and then push the* send *button. The Electronic Ballot Machine* ebm *duly registers the choices of the voter. Note that all the* joint *strategies of* $\{v, ebm\}$ *produce only finite runs. This is because* ebm *must choose a single event at location* 0 *in a memoryless strategy, and thus* v *and* ebm *are bound to "miscoordinate" either at the first or at the second step. Since finite paths are not included in the outcome sets, and the semantics in Section 3 rules out strategies with empty outcomes, we get that* $IIS(S_{vote}), 00 \models \neg\langle\langle v, ebm\rangle\rangle \mathrm{F}\,\top$*, which is quite strange.*

Notice that removing the non-emptiness requirement from the semantic clause in Section 3 does not help. In that case, any joint strategy of $\{v, ebm\}$ could be used to demonstrate that $\langle\langle v, ebm\rangle\rangle \mathrm{G}\,\bot$.

4.2 Solution: Adding Silent Transitions

To deal with the problem, we augment the model of the system with special "silent" transitions, labeled by ϵ, that are fired whenever no "real" transition can occur. In our case, the ϵ-transitions account for the possibility that some agents miscoordinate and thus block the system. Moreover, we redefine the outcome set of a strategy so that an ϵ-transition is taken whenever such miscoordination occurs.

Definition 4.3 (Undeadlocked IIS)**.** *Let* S *be an AMAS, and assume that no agent in* S *has* ϵ *in its alphabet of events. The* undeadlocked model of S*, denoted* $M^{\epsilon} = IIS^{\epsilon}(S)$*, extends the model* $M = IIS(S)$ *as follows:*

- $Evt_{M^\epsilon} = Evt_M \cup \{\epsilon\}$*, where* $Agent(\epsilon) = \emptyset$*;*

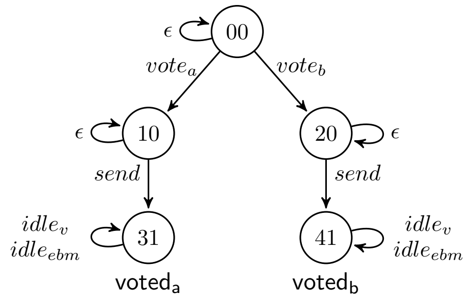

Figure 4: Undeadlocked IIS for the voting scenario

- *For each $g \in St$, we add the transition $g \xrightarrow{\epsilon} g$ iff there is a selection of agents' choices $\overrightarrow{\alpha}_A = (\alpha_1, \ldots, \alpha_k)$, $\alpha_i \in R_i(g)$, such that $enabled_M(g, \overrightarrow{\alpha}_A) = \emptyset$. In that case, we also fix $enabled_{M^\epsilon}(g, \overrightarrow{\alpha}_A) = \{\epsilon\}$.*

In other words, "silent" loops are added in the states where a combination of the agents' actions can block the system.

Paths are defined as in Section 2. The following is trivial.

Proposition 4.4. *For any AMAS S, any state $g \in IIS^\epsilon(S)$, and any strategy σ_A, we have that $enabled_{IIS^\epsilon(S)}(g, \sigma_A(state)) \neq \emptyset$.*

Example 4.5 (Conference)**.** *The undeadlocked model M^ϵ_{conf} of the conference scenario (Example 2.2) extends the model in Figure 2 with one ϵ-loop at state* 101*. The loop models the situation when the agents choose $(onsite, online, proceed)$ or $(online, onsite, proceed)$. We leave it for the reader to check that, at the other states, all the combinations of choices enable at least one transition.*

For the strategy in Example 4.1, notice that its outcome in M^ϵ_{conf} contains two *infinite paths: not only $(000\, giveup\, 002\, giveup\, 002 \ldots)$, but also $(000\, proceed\, 101\, \epsilon\, 101 \ldots)$. Since the latter path invalidates $\mathrm{G}\,\neg\mathsf{open}$, we get that $M_{conf}, 000 \not\models \langle\langle gc, oc\rangle\rangle \mathrm{G}\,\neg\mathsf{open}$, as expected.*

Example 4.6 (Voting)**.** *The undeadlocked model for the voting scenario is presented in Figure 4. Note that formula $\neg\langle\langle v, ebm\rangle\rangle \mathrm{F}\,\top$ does not hold anymore, because the joint strategies of $\{v, ebm\}$ have nonempty outcomes in $IIS^\epsilon(S_{vote})$. On the other hand, the formula $\langle\langle v\rangle\rangle \mathrm{F}\,\mathsf{voted_a}$ (and even $\langle\langle v, ebm\rangle\rangle \mathrm{F}\,\mathsf{voted_a}$) does not hold, which is contrary to the intuition behind the modeling. We will come back to this issue in Section 7.*

Discussion. Adding "silent" transitions to account for the control flow when no observable event occurs is pretty standard. The crucial issue is *where* to add them. Here, we add the ϵ-transitions whenever a subset of agents might choose to miscoordinate (and stick to their choices). Again, this is in line with the notion of agency and strategic play in MAS (Bratman 1987; Pauly 2001b). In the next section, we will discuss a concept of "agent fairness" where the addition of ϵ-transitions is constrained by the assumption that only a given subset of agents is fully proactive.

The examples used in this section expose an important feature of agent systems. The execution semantics of concurrent processes is often defined by a state-transition graph (or, alternatively, by the tree of paths generated by the graph, i.e., the tree unfolding of the graph). For systems that involve proactive agents, this is not enough. Rather, the execution semantics should map from the possible coalitions and their available strategies to the outcome sets of those strategies. In this sense, the possible behaviors of an agent system should be understood via the *set of possible execution trees*, rather than a single tree. This is consistent with the theoretical model of MAS in (Goranko and Jamroga 2015), based on path effectivity functions.

An alternative way out of the problem is to include finite maximal paths in the outcomes of strategies. However, the interpretation of strategic modalities over finite paths is rather nonstandard (Belardinelli et al. 2018) and may pose new problems in the asynchronous setting. Moreover, our approach allows to reuse the existing techniques and tools, which are typically built for infinite path semantics, including the verification and partial order reduction functionalities of tools like SPIN (Holzmann 1997) and STV (Kurpiewski et al. 2021). In general, this is a design dilemma between changing the logical semantics of the formulas vs. updating the execution semantics of the representations. Here, we choose the latter approach.

5 Playing Against Reactive Opponents

The solution proposed in Section 4.2 is based on the assumption that an agent is free to choose any event in its repertoire – even one that prevents the system from executing anything. The downside is that, for most systems, only safety goals can be achieved (i.e., properties specified by $\langle\langle A\rangle\rangle \mathrm{G}\,\varphi$). For reachability, there is often a combination of the opponents' choices that blocks the execution early on, and prevents the coalition from reaching their goal. In this section, we define a fairness-style condition that constrains the choices of more "reactive" opponents. We also show a construction to verify the abilities of the coalition over the resulting paths in a technically simpler way.

5.1 Opponent-Reactiveness

Given a strategy σ_A, the agents in A are by definition assumed to be proactive. Below, we propose an execution semantics for σ_A which assumes that A cannot be stalled forever by miscoordination on the part of the opponents.

Definition 5.1 (Opponent-reactiveness)**.** *A path $\pi = g_0\alpha_0 g_1\alpha_1 g_2 \ldots$ in $IIS^\epsilon(S)$ is* opponent-reactive *for strategy σ_A iff we have that $\alpha_n = \epsilon$ implies $enabled(g_n, \sigma_A(g_n)) = \{\epsilon\}$. In other words, whenever the agents outside A have a way to proceed, they must proceed. The* reactive outcome *of σ_A in g, denoted $out^{\mathrm{React}}(g, \sigma_A)$, is the restriction of $out(g, \sigma_A)$ to its opponent-reactive paths.*

Example 5.2 (Conference)**.** *Consider the undeadlocked model M^ϵ_{conf} of Example 4.5. Path $(000\, proceed\, 101\, \epsilon\, 101 \ldots)$ is opponent-reactive for the strategy of agents $\{gc, oc\}$ shown in Figure 1.*

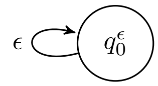

Figure 5: The auxiliary agent added in S^ϵ

On the other hand, consider coalition $\{gc, sc\}$, and the following strategy of theirs: $\sigma_{gc}(0) = proceed, \sigma_{gc}(1) = onsite, \sigma_{sc}(0) = proceed$. The same path is not *opponent-reactive for the strategy because the only opponent (oc) has a response at state* 101 *that enables a "real" transition (onsite).*

Proposition 5.3. *In $out^{\text{React}}_{IIS^\epsilon(S)}(g, \sigma_A)$, the only possible occurrence of ϵ is as an infinite sequence of ϵ-transitions following a finite prefix of "real" transitions.*

Proof. Take any $\pi = g_0\alpha_0g_1\alpha_1g_2\cdots \in out^{\text{React}}_{IIS^\epsilon(S)}(g, \sigma_A)$ such that ϵ occurs on π, and let i be the first position on π st. $\alpha_i = \epsilon$. By Definition 5.1, we get that $enabled(g_i, \sigma_A(g_i)) = \{\epsilon\}$. Moreover, $state_{i+1} = g_i$, so also $enabled(g_{i+1}, \sigma_A(g_{i+1})) = \{\epsilon\}$. Thus, $\alpha_{i+1} = \epsilon$. It follows by simple induction that $\alpha_j = \epsilon$ for every $j \geq i$. ☐

The *opponent-reactive semantics* $\models^{\text{React}}_{\text{ir}}$ of $\mathbf{ATL}^*$ is obtained by replacing $out_M(g, \sigma_A)$ with $out^{\text{React}}_M(g, \sigma_A)$ in the semantic clause presented in Section 3.

5.2 Encoding Strategic Deadlock-Freeness Under Opponent-Reactiveness in AMAS

If we adopt the assumption of opponent-reactiveness for coalition A, there is an alternative, technically simpler way to obtain the same semantics of strategic ability as in Section 4.2. The idea is to introduce the "silent" transitions already at the level of the AMAS.

Definition 5.4 (Undeadlocked AMAS)**.** *The* undeadlocked variant *of S is constructed from S by adding an auxiliary agent A_ϵ with $L_\epsilon = \{q_0^\epsilon\}$, $\iota_\epsilon = q_0^\epsilon$, $Evt_\epsilon = \{\epsilon\}$, $R_\epsilon(q_0^\epsilon) = \{\epsilon\}$, $T_i(q_0^\epsilon, \epsilon) = q_0^\epsilon$, and $\mathcal{PV}_\epsilon = \emptyset$. In other words, we add a module with a single local state and a "silent" loop labeled by ϵ, as in Figure 5. We will denote the undeadlocked variant of S by S^ϵ. Note that S^ϵ can be seen as a special case of AMAS. Thus, the outcome sets and reactive outcomes of strategies in $IIS(S^\epsilon)$ are defined exactly as before.*

Obviously, the extra agent adds ϵ-loops to the model of S, i.e., to $IIS(S)$. We show now that, under the assumption of opponent-reactiveness, the view of A's strategic ability in the undeadlocked AMAS S^ϵ corresponds precisely to A's abilities in the undeadlocked model of the original AMAS S, i.e., $IIS^\epsilon(S)$. This allows to deal with deadlocks and finite paths without redefining the execution semantics for AMAS, set in Definition 2.3, and thus use the existing tools such as SPIN (Holzmann 1997) in a straightforward way.

Proposition 5.5. *Let $A \subseteq \mathbb{A}\text{gt}$. In $out^{\text{React}}_{IIS(S^\epsilon)}(g, \sigma_A)$, the only possible occurrence of ϵ is as an infinite suffix of ϵ-transitions.*

Proof. Analogous to Proposition 5.3. ☐

Theorem 5.6. *For every strategy σ_A in S, we have that*

$$out^{\text{React}}_{IIS^\epsilon(S)}(g, \sigma_A) = out^{\text{React}}_{IIS(S^\epsilon)}(g, \sigma_A).$$

Proof. $\boldsymbol{out^{\text{React}}_{IIS^\epsilon(S)}(g, \sigma_A) \subseteq out^{\text{React}}_{IIS(S^\epsilon)}(g, \sigma_A)}$: Consider any $\pi = g_0\alpha_0g_1\alpha_1g_2\cdots \in out^{\text{React}}_{IIS^\epsilon(S)}(g, \sigma_A)$. If there are no ϵ-transitions on π, we have that $\pi \in out^{\text{React}}_{IIS(S)}(g, \sigma_A) \subseteq out^{\text{React}}_{IIS(S^\epsilon)}(g, \sigma_A)$, QED. Suppose that π includes ϵ-transitions, with α_i being the first one. Then, we have that $\alpha_j \neq \epsilon$ and $\alpha_j \in enabled_{IIS^\epsilon(S)}(g_j, \sigma_A(g_j))$ for every $j < i$, hence also $\alpha_j \in enabled_{IIS(S)}(g_j, \sigma_A(g_j)) \subseteq enabled_{IIS(S^\epsilon)}(g_j, \sigma_A(g_j))$. **(*)**
By Proposition 5.3, $g_j = g_i$ and $\alpha_j = \epsilon$ for every $j \geq i$. By Definition 5.1, $enabled_{IIS^\epsilon(S)}(g_j, \sigma_A(g_j)) = \{\epsilon\}$. Hence, $enabled_{IIS(S)}(g_j, \sigma_A(g_j)) = \emptyset$ and $enabled_{IIS(S^\epsilon)}(g_j, \sigma_A(g_j)) = \{\epsilon\}$. **(**)**
Thus, by **(*)** and **(**)**, $\pi \in out^{\text{React}}_{IIS(S^\epsilon)}(g, \sigma_A)$, QED.

$\boldsymbol{out^{\text{React}}_{IIS(S^\epsilon)}(g, \sigma_A) \subseteq out^{\text{React}}_{IIS^\epsilon(S)}(g, \sigma_A)}$: Analogous, with Proposition 5.5 used instead of Proposition 5.3. ☐

Discussion. Opponent-reactiveness is to strategic properties what fairness conditions are to temporal properties of asynchronous systems. If an important property cannot be satisfied in all possible executions, it may at least hold under some reasonable assumptions about which events can be selected by whom in response to what. Clearly, the condition can be considered intuitive by some and problematic by others. The main point is, unlike in the previous semantics, now it is made explicit, and can be adopted or rejected depending on the intuition.

Note that, under the reactiveness assumption, we have that $M^\epsilon_{conf}, 000 \models^{\text{React}}_{\text{ir}} \langle\langle gc, sc\rangle\rangle \mathrm{F}\,\mathsf{epid}$ and $M^\epsilon_{conf}, 000 \models^{\text{React}}_{\text{ir}} \langle\langle oc\rangle\rangle \mathrm{G}\,\neg\mathsf{epid}$. This seems to contradict the commonly accepted requirement of *regularity* in games (Pauly 2001a). However, the contradiction is only superficial, as the two formulas are evaluated *under different execution assumptions*: for the former, we assume agent oc to be reactive, whereas the latter assumes gc and sc to react to the strategy of oc.

6 Concurrency-Fairness Revisited

In Def. 3.6, we recalled the notion of concurrency-fair outcome of (Jamroga et al. 2018). The idea was to remove from $out(g, \sigma_A)$ the paths that consistently ignore agents whose events are enabled *at the level of the whole model*. Unfortunately, the definition has unwelcome side effects, too.

6.1 Problems with Concurrency-Fairness

We first show that, contrary to intuition, Definition 3.6 automatically disregards *deadlock paths*, i.e., paths with finitely many "real" transitions.

Proposition 6.1. *Consider an AMAS S and a path π in $IIS^\epsilon(S)$ such that, from some point i on, π includes only ϵ-transitions. Then, for every strategy σ_A in S, we have that $\pi \notin out^{\boldsymbol{CF}}_{IIS^\epsilon(S)}(g, \sigma_A)$.*

Proof. Take π as above, i.e., $\pi = g_0\alpha_0 g_1\alpha_1 \dots g_i\epsilon g_i\epsilon g_i \dots$. Since the transition function in $IIS^\epsilon(S)$ is serial, there must be some event $\beta \neq \epsilon$ enabled in g_i. In consequence, β is always enabled from i on, but none of its "owners" in $Agent(\beta)$ executes an event on π after i. Hence, π does not satisfy **CF**, and does not belong to $out^{\mathbf{CF}}_{IIS^\epsilon(S)}(g, \sigma_A)$ for any strategy σ_A. □

Thus, the **CF** condition eliminates all the deadlock paths from the outcome of a strategy (for instance, the path $(000\, proceed\, 101\, \epsilon\, 101 \dots)$ in Example 4.5). In consequence, reasoning about concurrency-fair paths suffers from the problems that we identified in Section 4.1, even for undeadlocked models. Moreover, combining the temporal and strategic fairness (i.e., **CF** and React) collapses the undeadlocked execution semantics altogether, see below.

Proposition 6.2. *Reasoning about reactive* and *fair outcomes in an undeadlocked model reduces to reasoning about the fair executions in the original model without ϵ-transitions. Formally, let $out^{\mathrm{React},\boldsymbol{CF}}_M(g, \sigma_A) = out^{\mathrm{React}}_M(g, \sigma_A) \cap out^{\boldsymbol{CF}}_M(g, \sigma_A)$. For any AMAS S and any strategy σ_A in S, we have:*

$$out^{\mathrm{React},\boldsymbol{CF}}_{IIS^\epsilon(S)}(g, \sigma_A) = out^{\boldsymbol{CF}}_{IIS(S)}(g, \sigma_A).$$

Proof. Clearly, we have $out^{\mathbf{CF}}_{IIS(S)}(g, \sigma_A) \subseteq out^{\mathrm{React},\mathbf{CF}}_{IIS^\epsilon(S)}(g, \sigma_A)$, since $out^{\mathrm{React},\mathbf{CF}}_{IIS^\epsilon(S)}(g, \sigma_A)$ can only add to $out^{\mathbf{CF}}_{IIS(S)}(g, \sigma_A)$ new paths that include ϵ-transitions.

For the other direction, take any $\pi \in out^{\mathrm{React},\mathbf{CF}}_{IIS^\epsilon(S)}(g, \sigma_A)$, and suppose that it contains an ϵ-transition. By Proposition 5.3, it must have an infinite suffix consisting only of ϵ-transitions. Then, by Proposition 6.1, $\pi \notin out^{\mathbf{CF}}_{IIS^\epsilon(S)}(g, \sigma_A)$, which leads to a contradiction. Thus, π contains only transitions from $IIS(S)$, and hence $\pi \in out^{\mathbf{CF}}_{IIS(S)}(g, \sigma_A)$, QED. □

6.2 Strategic Concurrency-Fairness

So, how should fair paths be properly defined for strategic reasoning? The answer is simple: in relation to the outcome of the strategy being executed.

Definition 6.3 (Strategic **CF**). $\pi = g_0\alpha_0 g_1\alpha_1 g_2 \dots$ *is a* concurrency-fair path for strategy σ_A and state g *iff $g_0 = g$, and there is no event α s.t., for some n and all $i \geq n$, we have $\alpha \in enabled(\pi[i], \sigma_A(\pi[i]))$ and $Agent(\alpha) \cap Agent(\alpha_i) = \emptyset$. That is, agents with an event always enabled* by σ_A *cannot be ignored forever.*

The **SCF**-outcome *of $\sigma_A \in \Sigma^{\mathrm{ir}}_A$ is defined as $out^{\boldsymbol{SCF}}_M(g, \sigma_A) = \{\pi \in out_M(g, \sigma_A) \mid \pi$ is concurrency-fair for $\sigma_A, g\}$.*

The following formal results show that **SCF** does not suffer from the problems demonstrated in Section 6.1.

Proposition 6.4. *There is an AMAS S, a strategy σ_A in S, and a deadlock path π in $IIS^\epsilon(S)$ such that π is concurrency-fair for σ_A.*

Proof. To demonstrate the property, it suffices to take the AMAS and the strategy of $\{gc, oc\}$ depicted in Figure 1, and the path $\pi = (000\, proceed\, 101\, \epsilon\, 101 \dots)$. □

Theorem 6.5. *Opponent-reactiveness and strategic **CF** are incomparable. Formally, there exists an AMAS S, a state g in $IIS^\epsilon(S)$, and a strategy σ_A such that $out^{\boldsymbol{SCF}}_{IIS^\epsilon(S)}(g, \sigma_A) \not\subseteq out^{\mathrm{React}}_{IIS^\epsilon(S)}(g, \sigma_A)$, and vice versa.*

Proof. Consider the undeadlocked model M^ϵ_{conf} in Example 4.5, and the strategy discussed in Example 5.2: $\sigma_{gc}(0) = proceed$, $\sigma_{gc}(1) = onsite$, $\sigma_{sc}(0) = proceed$. Let $\pi_1 = (000\, proceed\, 101\, \epsilon\, 101\, onsite\, 211\, rest\, 211\, handle\, 211\, rest\, 211 \dots)$. We have $\pi_1 \in out^{\mathbf{SCF}}_{M^\epsilon_{conf}}(g, \sigma_A)$, but $\pi_1 \notin out^{\mathrm{React}}_{M^\epsilon_{conf}}(g, \sigma_A)$. On the other hand, for path $\pi_2 = (000\, proceed\, 101\, onsite\, 211\, rest\, 211\, rest \dots)$, we have that $\pi_2 \notin out^{\mathbf{SCF}}_{M^\epsilon_{conf}}(g, \sigma_A)$, but $\pi_2 \in out^{\mathrm{React}}_{M^\epsilon_{conf}}(g, \sigma_A)$. □

Discussion. Theorem 6.5 suggests that reactiveness and fairness conditions arise from orthogonal concerns. The two concepts refer to different factors that influence which sequences of events can occur. Opponent-reactiveness constrains the choices that (a subset of) the agents can select. Concurrency-fairness and its strategic variant restrict the way in which the "scheduler" (Nature, Chance, God...) can choose from the events selected by the agents.

7 Strategies in Asymmetric Interaction

Now, we point out that AMAS are too restricted to model the strategic aspects of asymmetric synchronization in a natural way (e.g., a sender sending a message to a receiver).

7.1 Simple Choices are Not Enough

We demonstrate the problem on an example.

Example 7.1 (Voting). *As already pointed out, we have $IIS^\epsilon(S_{vote}), 00 \not\models \langle\langle v, ebm\rangle\rangle \mathrm{F}\, \mathsf{voted_a}$ in the model of Example 4.2. This is because receiving a vote for* a*, a vote for* b*, and the signal to send the vote, belong to* different choices *in the repertoire of the EBM, and the agent can only select one of them in a memoryless strategy. Moreover, formula $\langle\langle ebm\rangle\rangle \mathrm{F}\, \mathsf{voted_a}$ holds under the condition of opponent-reactiveness, i.e., the EBM can force a reactive voter to vote for a selected candidate. Clearly, it was not the intention behind the AMAS: the EBM is supposed to* listen *to the choice of the voter. No matter whose strategies are considered, and who reacts to whose actions, the EBM should have no influence on what the voter votes for.*

The problem arises because the repertoire functions in AMAS are based on the assumption that the agent can choose any single event in $R_i(l_i)$. This does not allow for natural specification of situations when the exact transition is determined by another agent. For the AMAS in Example 4.2, the decision to vote for candidate a or b (or to press $send$) should belong solely to the voter. Thus, setting the EBM repertoire as $R_{ebm}(0) = \{vote_a, vote_b, send\}$ does not produce a good model of strategic play in the scenario.

7.2 AMAS with Explicit Control

As a remedy, we extend the representations so that one can indicate which agent(s) control the choice between events.

Definition 7.2 (AMAS with explicit control)**.** *Everything is exactly as in Definition 2.1, except for the repertoires of choices, which are now functions $R_i : L_i \to 2^{2^{Evt_i}\setminus\{\emptyset\}}\setminus\{\emptyset\}$. That is, $R_i(l)$ lists nonempty subsets of events $X_1, X_2, \dots \subseteq Evt_i$, each capturing an available choice of i at the local state l. If the agent chooses $X_j = \{\alpha_1, \alpha_2, \dots\}$, then only an event in that set can be executed within the agent's module; however, the agent has no firmer control over which one will be fired. Accordingly, we assume that $T_i(l, \alpha)$ is defined iff $\alpha \in \bigcup R_i(l)$.*[1]

Notice that the AMAS of Definition 2.1 can be seen as a special case where $R_i(l)$ is always a list of singletons. The definitions of IIS and undeadlocked IIS stay the same, as agents' repertoires of choices are not actually used to generate the state-transition structure for the model of S. Moreover, undeadlocked AMAS with explicit control can be obtained analogously to Definition 5.4 by adding the auxiliary "epsilon"-agent with $R_\epsilon(q_0^\epsilon) = \{\{\epsilon\}\}$ in its sole local state.

Strategies still assign choices to local states; hence, the type of agent i's strategies is now $\sigma_i : L_i \to 2^{Evt_i} \setminus \{\emptyset\}$ s.t. $\sigma_i(l) \in R_i(l)$. The definition of the outcome set is updated accordingly, see below.

Definition 7.3 (Outcome sets for AMAS with explicit control)**.** *First, we lift the set of events enabled by $\overrightarrow{\alpha}_A = (\alpha_1, \dots, \alpha_m)$ at g to match the new type of repertoires and strategies. Formally, $\beta \in enabled(g, \overrightarrow{\alpha}_A)$ iff: (1) for every $i \in Agent(\beta) \cap A$, we have $\beta \in \alpha_i$, and (2) for every $i \in Agent(\beta) \setminus A$, it holds that $\beta \in \bigcup R_i(g^i)$.*

The outcome, React*-outcome, and* ***SCF****-outcome of σ_A in M, g are given as in Definitions 3.5, 5.1, and 6.3.*

Example 7.4 (Voting)**.** *We improve our voting model by assuming repertoires of choices for the voter and the EBM as follows: $R_{ebm}(0) = \{\{vote_a, vote_b, send\}\}$, $R_v(0) = \{\{vote_a\}, \{vote_b\}\}$, $R_v(1) = R_v(2) = \{\{send\}\}$, etc. That is, the voter's choices are as before, but the EBM only listens to what the voter selects.*

Clearly, $\langle\langle v, ebm\rangle\rangle \mathrm{F}\, \mathsf{voted_a}$ holds in the new AMAS. Moreover, $\langle\langle ebm\rangle\rangle \mathrm{F}\, \mathsf{voted_a}$ does not hold anymore, even assuming opponent-reactiveness.

It is easy to see that Propositions 4.4, 5.3, 5.5, and 6.4, as well as Theorems 5.6 and 6.5 still hold in AMAS with explicit control.

Discussion. When reasoning about strategic play of asynchronous agents, two kinds of asymmetry come into the picture. On the one hand, the processes (agents) being modeled often synchronize in an asymmetric way. For example, the sender chooses which message to send to the receiver. On the other hand, the agents A in formula $\langle\langle A\rangle\rangle\varphi$ choose the strategy and thus push the other agents to respond accordingly. The variant of AMAS introduced in (Jamroga et al. 2018) does not allow to capture the former kind of asymmetry. In consequence, the choice between the available synchronization branches belongs solely to the agents indicated by the formula. Unfortunately, there is no natural way to model the converse situation, i.e., when the agents in $\langle\langle A\rangle\rangle$ are forced by the choices of their opponents. With the new variant of AMAS, we extend the representations so that the modeler can explicitly specify the degree of autonomy of each participating agent. Without that, the degree of autonomy is implicit and comes from the formula being evaluated.

Various forms of asymmetric synchronization are present in most process algebras. For example, π-calculus distinguishes between the action $\overline{c}\langle a\rangle$ of sending the value a on channel c, and action $c(x)$ of listening on channel c and storing whatever comes in variable x. CSP goes further, and allows for a similar degree of flexibility to ours through suitable combinations of deterministic choice, nondeterministic choice, and interface parallel operators. Other synchronization primitives are also possible, see e.g. (Bloem et al. 2015). Instead of allowing for multiple synchronization primitives, we come up with a single general primitive that can be instantiated to cover different kinds of interaction.

We note in passing the similarity of our new repertoire functions in Definition 7.2 to state effectivity functions (Pauly 2001b; Pauly 2002) and especially alternating transition systems (Alur, Henzinger, and Kupferman 1998).

8 Partial Order Reduction Still Works

Partial order reduction (POR) has been defined for temporal and temporal-epistemic logics without "next" (Peled 1993; Gerth et al. 1999; Lomuscio, Penczek, and Qu 2010b), and recently extended to strategic specifications (Jamroga et al. 2018). The idea is to take a network of automata (AMAS in our case), and use depth-first search through the space of global states to generate a reduced model that satisfies exactly the same formulas as the full model. Essentially, POR removes paths that change only the interleaving order of an "irrelevant" event with another event. Importantly, the method generates the reduced model directly from the representation, without generating the full model at all.

8.1 Correctness of POR in the New Semantics

POR is a powerful technique to contain state-space explosion and facilitate verification, cf. e.g. the experimental results in (Jamroga et al. 2020). In this paper, we extend the class of models, and modify their execution semantics. We need to show that the reduction algorithm in (Jamroga et al. 2018), defined for the flawed semantics of ability, is still correct after the modifications. Our main technical result in this respect is Theorem 7.1, presented below. The detailed definitions, algorithms and proofs are technical (and rather tedious) adaptations of those in (Jamroga et al. 2018). We omit them here for lack of space, and refer the inquisitive reader to the extended version of the paper (Jamroga, Penczek, and Sidoruk 2020).

Theorem 7.1. Let $M = IIS(S^\epsilon)$, $M^\epsilon = IIS^\epsilon(S)$ and let $A \subseteq \mathbb{A}\mathrm{gt}$ be a subset of agents. Moreover, let $M' \subseteq M$

[1] For a set of sets X, we use $\bigcup X$ to denote $\bigcup_{x\in X} x$.

and $M^{\epsilon\prime} \subseteq M^{\epsilon}$ be the reduced models generated by DFS with the choice of enabled events $E(g')$ given by conditions **C1, C2, C3** and the independence relation $I_{A,PV}$. For each $\mathbf{sATL}^*_{\mathrm{ir}}$ formula φ over PV, that refers only to coalitions $\hat{A} \subseteq A$, we have:

1. $M, \iota \models^{\mathrm{React}}_{\mathrm{ir}} \varphi$ iff $M', \iota' \models^{\mathrm{React}}_{\mathrm{ir}} \varphi$, and
2. $M^{\epsilon}, \iota \models_{\mathrm{ir}} \varphi$ iff $M^{\epsilon\prime}, \iota' \models_{\mathrm{ir}} \varphi$.

Thus, the reduced models can be used to model-check the $\mathbf{sATL}^*_{\mathrm{ir}}$ properties of the full models.

Proof idea. We aim at showing that the full model M and the reduced one M' satisfy the same formulas of $\mathbf{ATL}^*_{\mathrm{ir}}$ referring only to coalitions $\hat{A} \subseteq A$ and containing no nested strategic operators. Thanks to the restriction on the formulas, the proof can be reduced to showing that M' satisfies the condition $\mathbf{AE}_A$, which states that for each strategy and for each path of the outcome of this strategy in M there is an equivalent path in the outcome of the same strategy in M'. In order to show that $\mathbf{AE}_A$ holds, we use the conditions on the selection of events $E(g')$ to be enabled at state g' in M'. The conditions include the requirement that ϵ is always selected, together with the three conditions $\mathbf{C1}, \mathbf{C2}, \mathbf{C3}$ adapted from (Peled 1994; Clarke, Grumberg, and Peled 1999; Jamroga et al. 2018).

Intuitively, $\mathbf{C1}$ states that, along each path π in M which starts at g', each event that is dependent on an event in $E(g')$ cannot be executed in M unless an event in $E(g')$ is executed first in M. $\mathbf{C2}$ says that $E(g')$ either contains all the events, or only events that do not change the values of relevant propositions. $\mathbf{C3}$ guarantees that for every cycle in M' containing no ϵ-transitions, there is at least one node g' in the cycle for which all the enabled events of g' are selected.

First, we show that M and M' are stuttering-equivalent, i.e., they have the same sets of paths modulo stuttering (that is, finite repetition of states on a path). The crucial observation here is that the reduction of M under the conditions **C1, C2, C3** is equivalent to the reduction of M without the ϵ-loops under the conditions **C1, C2, C3** of (Peled 1994), and then adding the ϵ-loops to all the states of the reduced model. Therefore, for the paths without ϵ-loops the stuttering equivalence can be shown similarly to (Clarke, Grumberg, and Peled 1999, Theorem 12) while for the paths with ϵ-loops we need more involved arguments in the proof. It turns out that in addition to the fact that M and M' are stuttering equivalent, we can show that stuttering equivalent paths of M and M' have the same maximal sequence of visible events. From that, we can prove that $\mathbf{AE}_A$ holds.

9 Conclusions

In this paper, we reconsider the asynchronous semantics of strategic ability for multi-agent systems, proposed in (Jamroga et al. 2018). We have already hinted at certain problems with the semantics in the extended abstract (Jamroga, Penczek, and Sidoruk 2021). Here, we demonstrate in detail how the straightforward combination of strategic reasoning and models of distributed systems leads to counterintuitive interpretation of formulas. We identify three main sources of problems. First, the execution semantics does not handle reasoning about deadlock-inducing strategies well. Secondly, fairness conditions need to be redefined for strategic play. Thirdly, the class of representations lacks constructions to resolve the tension between the asymmetry imposed by strategic operators on the one hand, and the asymmetry of interaction, e.g., between communicating parties.

We deal with the problems as follows. First, we change the execution semantics of strategies in asynchronous MAS by adding "silent" ϵ-transitions in states where no "real" event can be executed. We also propose and study the condition of *opponent-reactiveness* that assumes the agents outside the coalition to not obstruct the execution of the strategy forever. Note that, while the assumption may produce similar interpretation of formulas as in (Jamroga et al. 2018), it is now explicit – as opposed to (Jamroga et al. 2018), where it was "hardwired" in the semantics. The designer or verifier is free to adopt it or reject it, depending on their view of how the agents in the system behave and choose their actions.

Secondly, we propose a new notion of *strategic concurrency-fairness* that selects the fair executions of a strategy. Thirdly, we allow for nondeterministic choices in agents' repertoires. This way, we allow to explicitly specify that one agent has more control over the outcome of an event than the other participants of the event.

The main technical result consists in proving that partial order reduction for strategic abilities (Jamroga et al. 2018) is still correct after the semantic modifications. Thus, the new, more intuitive semantics admits efficient verification.

Beyond $\mathbf{ATL}_{\mathrm{ir}}$. In this study, we have concentrated on the logic $\mathbf{ATL}^*_{\mathrm{ir}}$, i.e., the variant of $\mathbf{ATL}^*$ based on memoryless imperfect information strategies. Clearly, the concerns raised here are not entirely (and not even not primarily) logical. $\mathbf{ATL}^*_{\mathrm{ir}}$ can be seen as a convenient way to specify the players and the winning conditions in a certain class of games (roughly speaking, 1.5-player games with imperfect information, positional strategies, and **LTL** objectives). The semantic problems, and our solutions, apply to all such games interpreted over arenas given by asynchronous MAS.

Moreover, most of the claims presented here are not specific to ir-strategies. In fact, we conjecture that our examples of semantic side effects carry over to the other types of strategies (except for the existence of coalitions whose all strategies have empty outcomes, which can happen for neither perfect information nor perfect recall). Similarly, our technical results should carry over to the other strategy types (except for the correctness of POR, which does not hold for agents with perfect information). We leave the formal analysis of those cases for future work.

Acknowledgements

We thank the anonymous reviewers for their insightful comments. The authors acknowledge the support of the National Centre for Research and Development, Poland (NCBR), and the Luxembourg National Research Fund (FNR), under the PolLux/FNR-CORE project STV (POLLUX-VII/1/2019). W. Penczek and T. Sidoruk also acknowledge the support from the CNRS/PAS project PARTIES.

References

Alur, R.; Henzinger, T.; Mang, F.; Qadeer, S.; Rajamani, S.; and Tasiran, S. 1998. MOCHA: Modularity in model checking. In *Proceedings of CAV*, volume 1427 of *Lecture Notes in Computer Science*, 521–525. Springer.

Alur, R.; de Alfaro, L.; Grossu, R.; Henzinger, T.; Kang, M.; Kirsch, C.; Majumdar, R.; Mang, F.; and Wang, B.-Y. 2001. jMocha: A model-checking tool that exploits design structure. In *Proceedings of ICSE*, 835–836. IEEE Computer Society Press.

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

Alur, R.; Henzinger, T. A.; and Kupferman, O. 1998. Alternating-time Temporal Logic. *Lecture Notes in Computer Science* 1536:23–60.

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

Belardinelli, F.; Lomuscio, A.; Murano, A.; and Rubin, S. 2017a. 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.

Belardinelli, F.; Lomuscio, A.; Murano, A.; and Rubin, S. 2017b. Verification of multi-agent 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.

Belardinelli, F.; Lomuscio, A.; Murano, A.; and Rubin, S. 2018. Alternating-time temporal logic on finite traces. In *Proceedings of IJCAI*, 77–83.

Bergstra, J. A., and Klop, J. W. 1985. Algebra of communicating processes with abstraction. *Theoretical Computer Science* 37:77–121.

Bloem, R.; Jacobs, S.; Khalimov, A.; Konnov, I.; Rubin, S.; Veith, H.; and Widder, J. 2015. *Decidability of Parameterized Verification*. Synthesis Lectures on Distributed Computing Theory. Morgan & Claypool Publishers.

Bratman, M. E. 1987. *Intentions, Plans, and Practical Reason*. Harvard University Press.

Busard, S.; Pecheur, C.; Qu, H.; and Raimondi, F. 2014. Improving the model checking of strategies under partial observability and fairness constraints. In *Formal Methods and Software Engineering*, volume 8829 of *Lecture Notes in Computer Science*. Springer. 27–42.

Cermák, P.; Lomuscio, A.; Mogavero, F.; and Murano, A. 2014. MCMAS-SLK: A model checker for the verification of strategy logic specifications. In *Proc. of CAV'14*, volume 8559 of *Lecture Notes in Computer Science*, 525–532. Springer.

Cermák, P.; Lomuscio, A.; and Murano, A. 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.

Chen, T.; Forejt, V.; Kwiatkowska, M.; Parker, D.; and Simaitis, A. 2013. PRISM-games: A model checker for stochastic multi-player games. In *Proceedings of TACAS*, volume 7795 of *Lecture Notes in Computer Science*, 185–191. Springer.

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

Gerth, R.; Kuiper, R.; Peled, D.; and Penczek, W. 1999. A partial order approach to branching time logic model checking. *Information and Computation* 150:132–152.

Godefroid, P., and Wolper, P. 1994. A partial approach to model checking. *Information and Computation* 110(2):305–326.

Goranko, V., and Jamroga, W. 2015. State and path coalition effectivity models of concurrent multi-player games. *Autonomous Agents and Multi-Agent Systems* 1–40.

Hoare, C. A. R. 1978. Communicating sequential processes. *Communications of the ACM* 21(8):666–677.

Holzmann, G. J. 1997. The model checker SPIN. *IEEE Transactions on Software Engineering* 23(5):279–295.

Huang, X., and van der Meyden, R. 2014. Symbolic model checking epistemic strategy logic. In *Proceedings of AAAI*, 1426–1432.

Jamroga, W.; Penczek, W.; Dembiński, P.; and Mazurkiewicz, A. 2018. Towards partial order reductions for strategic ability. In *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2018*, 156–165. IFAAMAS.

Jamroga, W.; Knapik, M.; Kurpiewski, D.; and Mikulski, Ł. 2019. Approximate verification of strategic abilities under imperfect information. *Artificial Intelligence* 277.

Jamroga, W.; Penczek, W.; Sidoruk, T.; Dembiński, P.; and Mazurkiewicz, A. 2020. Towards partial order reductions for strategic ability. *Journal of Artificial Intelligence Research* 68:817–850.

Jamroga, W.; Penczek, W.; and Sidoruk, T. 2020. Strategic abilities of asynchronous agents: Semantic paradoxes and how to tame them. *CoRR* abs/2003.03867. Revised version, July 2021.

Jamroga, W.; Penczek, W.; and Sidoruk, T. 2021. Strategic abilities of asynchronous agents: Semantic side effects. In *Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2021*, 1545–1547. ACM.

Kacprzak, M., and Penczek, W. 2004. Unbounded model checking for alternating-time temporal logic. In *Proceedings of the 3rd International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2004*, 646–653. IEEE Computer Society.

Kurpiewski, D.; Pazderski, W.; Jamroga, W.; and Kim, Y. 2021. STV+Reductions: Towards practical verification of strategic ability using model reductions. In *Proceedings of* *the 20th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2021*, 1770–1772. IFAAMAS.

Lomuscio, A., and Raimondi, F. 2006. MCMAS : A model checker for multi-agent systems. In *Proceedings of TACAS*, volume 4314 of *Lecture Notes in Computer Science*, 450–454. Springer.

Lomuscio, A.; Penczek, W.; and Qu, H. 2010a. Partial order reductions for model checking temporal epistemic logics over interleaved multi-agent systems. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2010*, 659–666.

Lomuscio, A.; Penczek, W.; and Qu, H. 2010b. Partial order reductions for model checking temporal-epistemic logics over interleaved multi-agent systems. *Fundam. Inform.* 101(1-2):71–90.

Lomuscio, A.; Qu, H.; and Raimondi, F. 2017. MCMAS: An open-source model checker for the verification of multi-agent systems. *International Journal on Software Tools for Technology Transfer* 19(1):9–30.

Lomuscio, A.; van der Meyden, R.; and Ryan, M. 2000. Knowledge in multiagent systems: initial configurations and broadcast. *ACM Transactions on Computational Logic* 1(2):247–284.

Milner, R.; Parrow, J.; and Walker, D. 1992. A calculus of mobile processes, I. *Information and Computation* 100(1):1–40.

Milner, R. 1980. *A Calculus of Communicating Systems*, volume 92 of *Lecture Notes in Computer Science*. Springer.

Pauly, M. 2001a. *Logic for Social Software*. Ph.D. Dissertation, University of Amsterdam.

Pauly, M. 2001b. A logical framework for coalitional effectivity in dynamic procedures. *Bulletin of Economic Research* 53(4):305–324.

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

Peled, D. 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, 409–423. Springer-Verlag.

Peled, D. 1994. Combining partial order reductions with on-the-fly model-checking. In *Proceedings of the 6th International Conference on Computer Aided Verification*, LNCS 818, 377–390. Springer-Verlag.

Peled, D. 1996. Partial order reductions: Model checking using representatives. In *Proceedings of the 21st International Symposium on Mathematical Foundations of Computer Science (MFCS'96)*, volume 1113 of *LNCS*, 93–112. Springer-Verlag.

Penczek, W.; Szreter, M.; Gerth, R.; and Kuiper, R. 2000. Improving partial order reductions for universal branching time properties. *Fundamenta Informaticae* 43:245–267.

Pilecki, J.; Bednarczyk, M.; and Jamroga, W. 2014. Synthesis and verification of uniform strategies for multi-agent systems. In *Proceedings of CLIMA XV*, volume 8624 of *Lecture Notes in Computer Science*, 166–182. Springer.

Priese, L. 1983. Automata and concurrency. *Theoretical Computer Science* 25(3):221 – 265.

Schobbens, P. Y. 2004. Alternating-time logic with imperfect recall. *Electronic Notes in Theoretical Computer Science* 85(2):82–93.