Journal of Artificial Intelligence Research 73 (2022) 897-932 Submitted 01/2022; published 03/2022

# Approximating Perfect Recall when Model Checking Strategic Abilities: Theory and Applications

**Francesco Belardinelli** FRANCESCO.BELARDINELLI@IMPERIAL.AC.UK
*Imperial College London, United Kingdom*
and *Laboratoire IBISC, Université d'Evry, France*

**Alessio Lomuscio** A.LOMUSCIO@IMPERIAL.AC.UK
*Imperial College London*
*United Kingdom*

**Vadim Malvone** VADIM.MALVONE@TELECOM-PARIS.FR
*Télécom Paris*
*France*

**Emily Yu** ZHENGQI.YU@JKU.AT
*Imperial College London, United Kingdom*
and *Johannes Kepler University Linz, Austria*

## Abstract

The model checking problem for multi-agent systems against specifications in the alternating-time temporal logic $ATL$, hence $ATL^*$, under perfect recall and imperfect information is known to be undecidable. To tackle this problem, in this paper we investigate a notion of bounded recall under incomplete information. We present a novel three-valued semantics for $ATL^*$ in this setting and analyse the corresponding model checking problem. We show that the three-valued semantics here introduced is an approximation of the classic two-valued semantics, then give a sound, albeit partial, algorithm for model checking two-valued perfect recall via its approximation as three-valued bounded recall. Finally, we extend MCMAS, an open-source model checker for $ATL$ and other agent specifications, to incorporate bounded recall; we illustrate its use and present experimental results.

## 1. Introduction

Alternating-time Temporal Logic ($ATL$) and its extension $ATL^*$ are widely used formalisms to reason about strategic abilities of autonomous agents in multi-agent systems (Alur et al., 2002). Central to $ATL$ and related formalisms is the notion of the sequence of events a coalition of agents can jointly bring about, or avoid, in a system, irrespective of the actions of the other agents outside the coalition. $ATL$ has been extended in various directions giving rise to even more expressive formalisms, for example, by taking into account continuous time (Knapik et al., 2019), bounded resources (Alechina et al., 2015, 2018), epistemic concepts (Hoek & Wooldridge, 2003; Jamroga, 2004; Lomuscio & Raimondi, 2006; Ågotnes et al., 2015), and beyond.

A key consideration when using expressive specification languages, including $ATL$, is the computational complexity of the resulting model checking problem. In the case of $ATL$, this was shown to be *PTIME*-complete under perfect information (Alur et al., 2002). Agents in a multi-agent system (MAS), however, typically operate under imperfect information about the other agents and the environment. Once imperfect information is assumed, the resulting model checking problem becomes $\Delta_2^P$-complete under memoryless semantics (Jamroga & Dix, 2006), and it is undecidable under perfect recall (Dima & Tiplea, 2011). The latter case is particularly problematic since it hinders the development of any verification toolkit.

Recent approaches have attempted to overcome these difficulties. For instance, if agents can only communicate via broadcasting, decidability can be retained (Belardinelli et al., 2020a). Further, hierarchical systems, where information is shared in a strictly predetermined manner, have also been shown to provide decidable fragments (Berthon et al., 2021). These contributions analyse the verification problem under perfect recall and imperfect information, but they restrict the class of MAS considered. Here we take a different approach: we consider the whole class of MAS, but define an approximation of perfect recall that we call *bounded recall*. Informally, an agent's recall is bounded, if in her deliberations she disregards explicit information acquired more than a certain number of timestamps before. Therefore, under $n$-bounded recall, an agent's strategy does not depend on her whole history, but only on her last $n$ visited states. This is a natural assumption when reasoning about the abilities of agents in a concrete setting, as opposed to a purely theoretical one. Indeed, similar notions of resource-bounded strategies have been previously investigated in the literature as we discuss in detail below.

**Contributions.** In this paper we make three main contributions. Firstly, in Section 3 we develop a novel three-valued semantics for $ATL^*$ under bounded recall, which covers perfect recall as well as a limit case. We study the corresponding model checking problem, and analyse the formal properties of three-valued $ATL^*$ against the classic, two-valued, semantics. The main finding of this section is that – in terms of verification – bounded recall provides a provably sound approximation of perfect recall. This is shown in Corollary 2 below, which states that MAS properties under perfect recall can be decided by analysing their bounded recall approximations. Secondly, these theoretical results lay the foundations for a verification procedure for model checking MAS under imperfect information and perfect recall, by iteratively checking bounded recall versions of the same MAS in the three-valued semantics, with increasing amounts of memory. While the algorithm is incomplete in general, we show that if a bound on recall is assumed, it terminates in *EXPTIME*. Section 5 reports on an implementation of the algorithm, realised by extending MCMAS (Lomuscio et al., 2017), an open-source model checker for MAS, to bounded recall. Thirdly, we define the three-valued model checking procedure for $ATL^*$ in terms of a reduction to two-valued model checking. We deem the translations provided in Section 3 of general interest to reduce the model checking problem for multi-valued logic in general to classic two-valued model checking.

**Related Work.** We now discuss our work in the context of recent contributions on logic-based languages for the specification and verification of strategic abilities of agents in multi-agent systems (Alur et al., 2002).

**Three-valued $ATL$.** Three-valued temporal logics have been extensively explored in the literature on system verification, for example (Bruns & Godefroid, 1999; Godefroid & Jagadeesan, 2003; Ball & Kupferman, 2006; Shoham & Grumberg, 2004; Huth et al., 2004; Huth & Pradhan, 2004), including run-time verification (Bauer et al., 2006, 2007). Our approach differs from that of Bruns & Godefroid (1999); Godefroid & Jagadeesan (2003) in that it is not based on the definition of under- and over-approximations of transition systems. Ball & Kupferman (2006); Shoham & Grumberg (2004) put forward 3-valued abstraction techniques for $CTL$ and the alternating $\mu$-calculus (A$\mu$C), which assumes that agents have perfect information about their environment; instead, we consider the more complex case of imperfect information and perfect recall.

A related line of work, closely related to the present approach, is the one on three-valued semantics for $ATL$. Lomuscio & Michaliszyn (2014, 2015) introduce three-valued abstractions for interpreted systems to address the complexity of MAS verification. These investigations were developed further by Belardinelli et al. (2016); Lomuscio & Michaliszyn (2016) by means of predicate abstraction. While we take our inspiration from this line, our present contribution differs significantly. Firstly, the semantics and the underlying classes of systems we study here are different from those by Lomuscio & Michaliszyn (2014, 2015, 2016). Specifically, these works assume *non-uniform* strategies (Lomuscio & Raimondi, 2006), with significant implications on the decidability and complexity of the corresponding model checking problem. In particular, under non-uniform strategies model checking $ATL$ *with imperfect information* on interpreted systems is decidable in $PTIME$ both for the memoryfull and memoryless case (which is the whole point of considering non-uniform strategies). Hence, approximating perfect recall is not an issue in the setting of Lomuscio & Michaliszyn (2014, 2015, 2016). On the contrary, we here consider uniform strategies, as this is the framework commonly used when analysing strategic abilities of agents in MAS and game-theoretical contexts (Jamroga & van der Hoek, 2004). Secondly, the aims of the respective lines are different as we here seek an approximation of perfect recall via bounded recall.

A three-valued semantics for strategic abilities is also used by Belardinelli & Lomuscio (2017). However, similarly to the above, the authors focus on imperfect recall and their $ATL$ operators are interpreted differently from what we do here. More formally, according to Belardinelli & Lomuscio (2017), the falsehood of a formula of type $\langle\!\langle \Gamma \rangle\!\rangle \psi$ is given in terms of *may-strategies* of coalition $\Gamma$, whereas we here define it in terms of the strategic abilities of the complement coalition $\bar{\Gamma}$. This is a key feature of our semantics, as it allows us to preserve defined truth values when adding recall (Lemma 2). Furthermore, Belardinelli et al. (2019); Belardinelli & Malvone (2020) present a three-valued semantics for $ATL$ in the context of imperfect information and perfect recall strategies. In particular, the authors present an approximation of imperfect information to recover decidability. On the other hand, in this work we adopt the symmetric point of view by approximating perfect recall.

**Multi-valued Logics for Verification.** Multi-valued semantics have long been explored in the modal logic literature (Fitting, 1991, 1992). Since the early 2000s, multi-valued temporal logics have been used in the verification of distributed and multi-agent systems. Multi-valued semantics for the verification of specifications expressed in the temporal logic $CTL^*$ was first proposed by Vijzelaar & Fokkink (2017), and then extended to the modal $\mu$-calculus (Gurfinkel & Chechik, 2003; Bruns & Godefroid, 2003; Shoham & Grumberg, 2012; Pan et al., 2016). In this line formulas are interpreted on a possibly infinite algebraic structure, and modal operators correspond to operations on the values in the structure.

A similar approach has also been applied to temporal-epistemic logics for multi-agent systems (Konikowska & Penczek, 2002, 2004, 2006), including $ATL^*$ under imperfect information (Jamroga et al., 2020). However, a key difference w.r.t. our contribution is that none of the works above mentioned concerns the approximation of perfect recall by using bounded recall.

**Bounded Recall.** Classic, two-valued bounded recall and bounded strategies have been studied quite extensively in the literature. Ågotnes & Walther (2009) consider strategies according to two different notions of bound: over the set of histories and over the length of histories. In this framework, they show that $ATL$ with bounded memory is strictly more expressive than standard $ATL$. (Brihaye et al., 2009) extend $ATL$ in two directions: strategy contexts and bounded memory. Then, the model checking problem is proved to be in *EXPSPACE*. Further, Jamroga et al. (2019c,d) define strategies as a list of condition-action rules. Then, the authors present a variant of $ATL$ that makes use of strategy operators with a bound on the size of this list. In these contributions boundedness is studied from an expressiveness and complexity perspective, not as an approximation of perfect recall, which is the main focus of the present work. In some cases, the semantics are incomparable to ours even in a two-valued setting (Ågotnes & Walther, 2009; Brihaye et al., 2009). Finally, approximations to model check $ATL$ under imperfect information ($i$) have also appeared in the work of Jamroga et al. (2019b) with some significant differences. Jamroga et al. (2019b) consider *syntactic* approximation, rather than semantical, under the assumption of imperfect recall ($r$). So, their aim is to improve the performance of model checking $ATL_{ir}$, rather than approximating an undecidable problem.

Related to the line above, Vester (2013) presents an account of bounded strategies via finite-memory transducers. It is instructive to compare his treatment to ours. We explore this in Section 2.4 where we show that some finite-memory transducers cannot be translated polynomially into our bounded recall strategies and some bounded recall strategies cannot be polynomially recast as transducers. The two accounts are therefore incomparable in general. A further key point of departure is that our notion of bounded recall is intended to provide a basis for an iterative verification procedure for MAS based on a novel three-valued semantics, whereas Vester (2013) focus specifically on the theoretical properties of bounded recall.

Lastly, and unrelated to the above, Deuser & Naumov (2020) study how bounded recall affects the agents' abilities to execute plans composition. While their logic has the flavour of $ATL$ and strategic concepts, the machinery employed is different from ours and so are the overall goals of the investigation: axiomatisations in their case, verification in ours.

**Previous work**. This paper builds upon and extends previous contributions by the authors. Belardinelli et al. (2020b) consider bounded recall on interpreted systems but for a temporal epistemic logic, whose temporal part $CTL$ is strictly less expressive than $ATL$. In that work no notion of bounded recall on strategies is present and the verification algorithms are therefore different. More closely related to this contribution is the work by Belardinelli et al. (2018), where a three-valued semantics for $ATL^*$ was introduced. This article substantially extends the work of Belardinelli et al. (2018) by providing the complexity analysis of the various verification problems studied, full proofs for all main results, and additional details. Moreover, this article contains a reduction from three-valued model checking to the two-valued instance, which is original of this work and a stand-alone contribution in itself. Finally, no implementation was provided by Belardinelli et al. (2018), while here we are able to extend an open-source model checker and evaluate experimentally the performance of the proposed approach.

**Structure of the paper.** The rest of the paper is organised as follows. In Section 2 we introduce the notion of bounded recall in the context of interpreted systems and $ATL^*$, and compare bounded and perfect recall from the perspective of verification. In Section 3 we present our novel three-valued semantics for bounded and perfect recall, study the corresponding model checking problems, and analyse its formal properties against its classic formulation. Section 5 reports an implementation of the algorithm, realised by extending MCMAS to bounded recall. We conclude in Section 6.

## 2. Classic Bounded Recall

In this section we introduce a new two-valued semantics for $ATL^*$ under imperfect information and bounded recall, based on the standard interpretation of $ATL^*$ Alur et al. (2002). Then, we study the complexity of the corresponding model checking problem, and compare it with the case of perfect recall. Hereafter we assume sets $Ag = \{1, \ldots, m\}$ of indices for agents and $AP$ of atomic propositions. Given a set $U$, $\overline{U}$ denotes its complement. We denote the length of a tuple $v$ of elements as $|v|$, and its $i$th element either as $v_i$ or $v.i$. Then, let $last(v) = v_{|v|}$ be the last element in $v$. For $i \leq |v|$, let $v_{\geq i}$ be the suffix $v_i, \ldots, v_{|v|}$ of $v$ starting at $v_i$ and $v_{\leq i}$ the (finite) prefix $v_1, \ldots, v_i$ of $v$ starting at $v_1$. Finally, $\mathbb{N}^+ = \mathbb{N} \setminus \{0\}$ is the set of positive naturals.

### 2.1 Interpreted Systems

We follow the presentation of interpreted systems as given by Fagin et al. (1995). We will use them as a semantics for $ATL^*$ as originally put forward by Lomuscio & Raimondi (2006), rather than concurrent game structures. Nonetheless, the two accounts are closely related (Goranko & Jamroga, 2004).

**Definition 1** (Agent)**.** Given a set $Ag$ of indices for agents, an *agent* is a tuple $i = \langle L_i, Act_i, P_i, t_i \rangle$ such that

- $L_i$ is the finite set of *local states*;
- $Act_i$ is the finite set of *individual actions*;
- $P_i : L_i \to (2^{Act_i} \setminus \emptyset)$ is the *protocol function*;
- $t_i : L_i \times ACT \to L_i$ is the *local transition function*, where $ACT = Act_1 \times \cdots \times Act_{|Ag|}$ is the set of *joint actions*, such that for every $l \in L_i$, $a \in ACT$, $t_i(l, a)$ is defined iff $a_i \in P_i(l)$.

By Def. 1 an agent $i$ is situated in some local state $l \in L_i$, which represents the information she has about the current state of the system. At any state she can perform the actions in $Act_i$ according to protocol $P_i$. A joint action brings about a change in the state of the agent, according to the local transition function $t_i$. Hereafter, with an abuse of notation, we identify an agent index $i$ with the corresponding agent.

Given set $Ag$ of agents, a *global state* $s \in \mathcal{G}$ is a tuple $\langle l_1, \ldots, l_{|Ag|} \rangle$ of local states, one for each agent in $Ag$. Notice that an agent's protocol and transition function depend only on her local state, which might contain strictly less information than the global state. In this sense agents have *imperfect information* about the system. A history $h \in \mathcal{G}^+$ is a finite (non-empty) sequence of global states. For $n \geq 1$, $\mathcal{G}^n$ denotes the set of histories of length $n$, and $\mathcal{G}^{<1+n} = \bigcup_{1 \leq m \leq n} \mathcal{G}^m$ is the set of histories of length *at most* $n$; whereas $\mathcal{G}^{<\omega}$ denotes the set of all finite histories, that is, $\mathcal{G}^{<\omega} = \mathcal{G}^+$.

For every agent $i \in Ag$, we define an *indistinguishability relation* $\sim_i$ between global states based on the identity of local states, that is, $s \sim_i s'$ iff $s_i = s'_i$ (Fagin et al., 1995). This indistinguishability relation is extended to histories in a synchronous, pointwise way, that is, histories $h, h' \in \mathcal{G}^+$ are *indistinguishable* for agent $i \in Ag$, or $h \sim_i h'$, iff (i) $|h| = |h'|$ and (ii) for every $j \leq |h|$, $h_j \sim_i h'_j$.

**Definition 2** (IS)**.** An *interpreted system* is a tuple $M = \langle Ag, s_0, T, \Pi \rangle$, where

- $Ag$ is the set of *agents*;
- $s_0 \in \mathcal{G}$ is the *(global) initial state*;
- $T : \mathcal{G} \times ACT \to \mathcal{G}$ is the *global transition function* such that $s' = T(s, a)$ iff for every $i \in Ag$, $s'_i = t_i(s_i, a)$;
- $\Pi : \mathcal{G} \times AP \to \{\text{tt}, \text{ff}\}$ is the *(two-valued) labelling function*.

Intuitively, an interpreted system describes the interactions of a group $Ag$ of agents, starting from the initial state $s_0$, according to the transition function $T$. Notice that $T$ is defined on state $s$ for joint action $a$ iff $a_i \in P_i(s_i)$ for every $i \in Ag$.

## 2.2 ATL with Bounded Recall

We make use of the Alternating-time Temporal Logic $ATL^*$ (Alur et al., 2002) to reason about the strategic abilities of agents in interpreted systems.

**Definition 3** ($ATL^*$)**.** State ($\varphi$) and path ($\psi$) formulas in $ATL^*$ are defined as follows, for $q \in AP$ and $\Gamma \subseteq Ag$:

$$\begin{array}{rcl} \varphi & ::= & q \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\langle \Gamma \rangle\rangle \psi \\ \psi & ::= & \varphi \mid \neg\psi \mid \psi \wedge \psi \mid X\psi \mid (\psi U \psi) \end{array}$$

Formulas in $ATL^*$ are all and only the state formulas.

As customary, a formula $\langle\langle \Gamma \rangle\rangle \psi$ is read as 'the agents in coalition $\Gamma$ have a strategy to achieve goal $\psi$'. The meaning of $LTL$ operators 'next' $X$ and 'until' $U$ is standard (Baier & Katoen, 2008). Operators 'unavoidable' $[\![\Gamma]\!]$, 'eventually' $F$, and 'always' $G$ can be introduced as usual.

Formulas in the $ATL$ fragment of $ATL^*$ are obtained from Def. 3 by restricting path formulas $\psi$ as follows, where $\varphi$ is a state formula and $R$ is the release operator[1]:

$$\psi \quad ::= \quad X\varphi \mid (\varphi U \varphi) \mid (\varphi R \varphi)$$

In the rest of the paper we consider two other relevant fragments of $ATL^*$: the existential and universal fragments.

[1] Notice that the release operator $R$ can be defined in $ATL^*$ as the dual of until $U$ (indeed, it does not appear in the syntax of Def. 3), while it must be assumed as a primitive operator in $ATL$. We refer to Laroussinie et al. (2008) for more details on this point.

**Definition 4.** Let $q \in AP$ and $\Gamma \subseteq Ag$. State ($\varphi$) and path ($\psi$) formulas in the *existential fragment* $\exists ATL^*$ of $ATL^*$ are defined as follows:

$$\begin{array}{rcl} \varphi & ::= & q \mid \neg q \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \langle\!\langle \Gamma \rangle\!\rangle \psi \\ \psi & ::= & \varphi \mid \psi \vee \psi \mid \psi \wedge \psi \mid X\psi \mid (\psi U \psi) \mid (\psi R \psi) \end{array}$$

Path formulas ($\psi$) in the *universal fragment* $\forall ATL^*$ of $ATL^*$ are defined as for $\exists ATL^*$; whereas state formulas ($\varphi$) are defined as follows:

$$\varphi \quad ::= \quad q \mid \neg q \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid [\![\Gamma]\!] \psi$$

By Def. 4 in the existential (resp. universal) fragment, formulas are only of the form $\langle\!\langle \Gamma \rangle\!\rangle \psi$ (resp. $[\![\Gamma]\!] \psi$) or boolean combinations thereof. In particular, operator $[\![\Gamma]\!]$ (resp. $\langle\!\langle \Gamma \rangle\!\rangle$) is no longer definable in the existential (resp. universal) fragment.

Since the behaviour of agents in interpreted systems depends only on their local state, we assume agents employ *uniform strategies* (Jamroga & van der Hoek, 2004). That is, they perform the same action whenever they have the same information. Moreover, we assume that agents have some bounded recall of the local states visited during an execution. This is formalised as follows.

**Definition 5** (Uniform Strategy with Bounded Recall)**.** For $n \in \mathbb{N}^+ \cup \{\omega\}$, a *uniform strategy with n-bounded recall* for agent $i \in Ag$ is a function $f_i^n : \mathcal{G}^{<1+n} \to Act_i$ such that for all histories $h, h' \in \mathcal{G}^{<1+n}$, (i) $f_i^n(h) \in P_i(last(h).i)$; and (ii) $h \sim_i h'$ implies $f_i^n(h) = f_i^n(h')$.

By Def. 5 any strategy for agent $i$ has to return actions that are enabled for $i$. Also, whenever two histories are indistinguishable for agent $i$, then the same action is returned. Notice that for $n = 1$, we obtain memoryless (or imperfect recall) strategies; whereas for $n = \omega$, $1 + n = \omega$ and we have memoryful (or perfect recall) strategies.

Given an IS $M$, a *path* $p$ is an infinite sequence $s_1 s_2 \ldots$ of global states. For a set $F_\Gamma^n = \{f_i^n \mid i \in \Gamma\}$ of strategies, one for each agent in coalition $\Gamma$, a path $p$ is $F_\Gamma^n$*-compatible* iff for every $j > 0$, $p_{j+1} = T(p_j, a)$ for some joint action $a \in ACT$ such that for every $i \in \Gamma$, $a_i = f_i^n(p_1, \ldots, p_j)$ for $j \leq n$, $a_i = f_i^n(p_{j-n}, \ldots, p_j)$ otherwise. Hence, for $n \in \mathbb{N}^+$, $n$-bounded recall strategies take into account at most the $n$ previously visited states. This modelling choice is meant to account for agents with finite recall of past events (Ågotnes & Walther, 2009; Vester, 2013). In particular, any actual implementation of MAS with some sort of recall can only employ bounded recall, for some bound determined by the system's memory capacity. Finally, let $out(s, F_\Gamma^n)$ be the set of all $F_\Gamma^n$-compatible paths starting with some $s'$ such that $s' \sim_i s$ for some agent $i \in \Gamma$.

We can now assign a meaning to $ATL^*$ formulas on interpreted systems based on a semantics with two truth values: ff and tt.

**Definition 6** (Satisfaction)**.** Let $n \in \mathbb{N}^+ \cup \{\omega\}$. The two-valued satisfaction relation $\models_n^2$ for an IS $M$, state $s$, path $p$, and $ATL^*$ formula $\phi$ is defined as follows:

$$\begin{array}{lll} (M, s) \models_n^2 q & \text{iff} & \Pi(s, q) = \text{tt} \\ (M, s) \models_n^2 \neg\varphi & \text{iff} & (M, s) \not\models_n^2 \varphi \end{array}$$

$$
\begin{array}{lll}
(M,s) \models^2_n \varphi \wedge \varphi' & \text{iff} & (M,s) \models^2_n \varphi \text{ and } (M,s) \models^2_n \varphi' \\
(M,s) \models^2_n \langle\langle \Gamma \rangle\rangle \psi & \text{iff} & \text{for some joint strategy } F^n_\Gamma \text{, for all paths } p \in \mathit{out}(s, F^n_\Gamma) \text{, } (M,p) \models^2_n \psi \\
(M,p) \models^2_n \varphi & \text{iff} & (M,p_1) \models^2_n \varphi \\
(M,p) \models^2_n \neg\psi & \text{iff} & (M,p) \not\models^2_n \psi \\
(M,p) \models^2_n \psi \wedge \psi' & \text{iff} & (M,p) \models^2_n \psi \text{ and } (M,p) \models^2_n \psi' \\
(M,p) \models^2_n X\psi & \text{iff} & (M,p_{\geq 2}) \models^2_n \psi \\
(M,p) \models^2_n \psi U \psi' & \text{iff} & \text{for some } k \geq 1, (M,p_{\geq k}) \models^2_n \psi' \text{, and} \\
 & & \text{for all } j \text{, } 1 \leq j < k \text{ implies } (M,p_{\geq j}) \models^2_n \psi
\end{array}
$$

Def. 6 is parameterised by $n$ for bounded recall. It can be checked that for $n = 1$ (resp. $n = \omega$) we obtain the standard satisfaction clauses for $ATL^*_{ir}$ and $ATL^*_{iR}$ for imperfect information and imperfect (resp. perfect) recall[2]. We say that formula $\varphi$ is *true* in an IS $M$ (for $n$-bounded recall), or $M \models^2_n \varphi$, iff $(M, s_0) \models^2_n \varphi$. Furthermore, in Def. 6 we use $\not\models^2_n$ to represent that it is not the case that $\models^2_n$.

Finally, we observe that Def. 6 corresponds to the *subjective* interpretation of $ATL^*$, whereby formulas $\langle\langle \Gamma \rangle\rangle \psi$ are evaluated w.r.t. all paths $p \in \mathit{out}(s, F^n_\Gamma)$ compatible with some $s'$ indistinguishable from $s$ for some agent in $\Gamma$ (as well as the joint strategy $F^n_\Gamma$). This is a well-established semantical account in logics for strategies (Jamroga & van der Hoek, 2004), which has found applications in MAS verification (Busard et al., 2015). Intuitively, a formula $\langle\langle \Gamma \rangle\rangle \psi$ is true in a state $s$ according to the subjective interpretation if the strategy used by coalition $\Gamma$ is not merely successful in achieving goal $\psi$, but all the agents in $\Gamma$ know it to be successful as well. Moreover, the subjective interpretation allows us to introduce an epistemic operator as follows: $K_i \psi ::= \langle\langle \{i\} \rangle\rangle \psi U \psi$, whose semantics is derived as:

$$(M,s) \models^2_n K_i \psi \quad \text{iff} \quad \text{for every } s' \in S, \, s' \sim_i s \text{ implies } (M,s') \models^2_n \psi$$

The epistemic operator $K_i$ expresses an external notion of knowledge as described in the literature on epistemic logic (Fagin et al., 1995; Meyer & Hoek, 1995).

Now, we exemplify the formal machinery introduced so far with two examples.

**Example 1.** We consider a revised version of the Shell Game by Bulling et al. (2014) in which a Shuffler and a Guesser participate in a game with $N$ shells on the table. The Shuffler places a ball in one of the shells. The shells are initially placed in such a way that the Guesser can see the location of the ball. Then the Shuffler turns the shells over, so that the ball becomes hidden. From that point, the Guesser needs to wait for $m$ timestamps before submitting her guess. The Guesser wins the game if she successfully guesses the location of the ball. The atom $\mathit{shell}_i$ is assigned to states where the Shuffler places the ball in shell $i \leq N$, and atom $\mathit{guess}_i$ is true when the Guesser guesses shell $i$ (but this does not mean that the choice of the Guesser is correct).

More formally, this game can be represented as the IS $M = \langle Ag, s_0, T, \Pi \rangle$, such that $Ag = \{\mathit{Shuffler}, \mathit{Guesser}\}$, $Act_{\mathit{Shuffler}} = \{\mathit{place}_1, \ldots, \mathit{place}_N, H_1, \ldots, H_m, I\}$ where by action $\mathit{place}_i$ the shuffler places the ball in shell $i$, whereas by action $H_i$ he does the $i$-th step of hiding, and $Act_{\mathit{Guesser}} = \{\mathit{guess}_1, \ldots, \mathit{guess}_N, I\}$, where by action $\mathit{guess}_i$ the *Guesser* guesses the location of the ball in $i$. Finally, $I$ is the idle action. The global transition

[2] Note that: $i$ stands for imperfect information, $r$ stands for memoryless strategies, and $R$ for memoryful strategies, as introduced by Schobbens (2004).

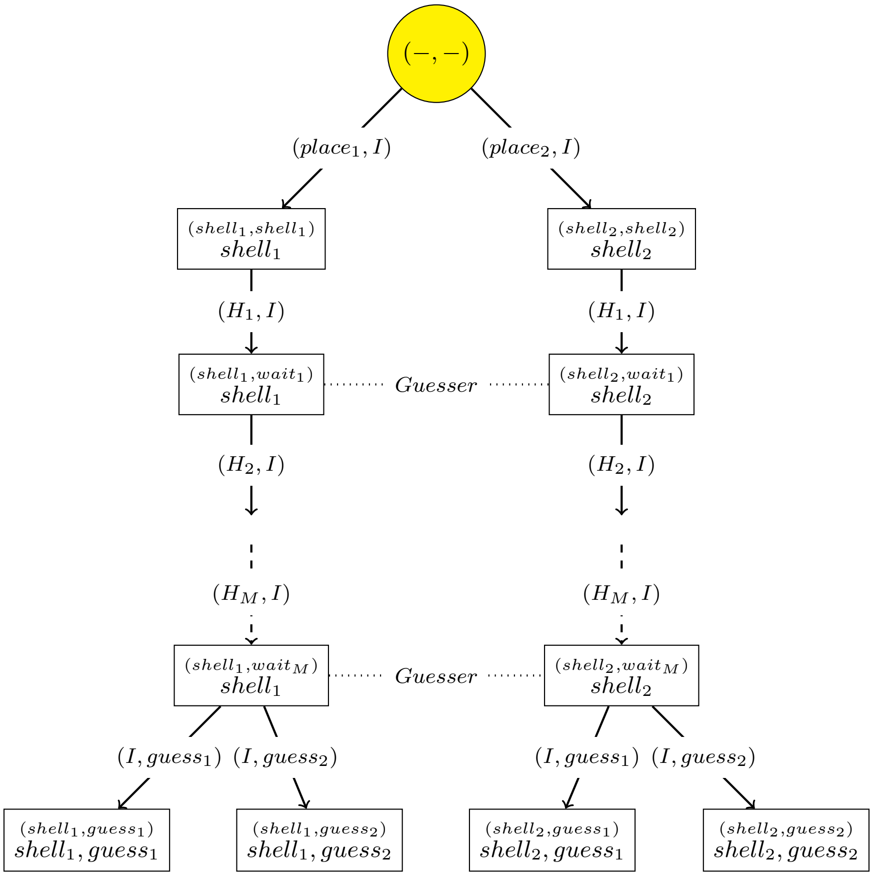
$(-,-)$
$(place_1, I)$
$(place_2, I)$
$(shell_1, shell_1)$ $shell_1$
$(shell_2, shell_2)$ $shell_2$
$(H_1, I)$
$(H_1, I)$
$(shell_1, wait_1)$ $shell_1$
Guesser
$(shell_2, wait_1)$ $shell_2$
$(H_2, I)$
$(H_2, I)$
$(H_M, I)$
$(H_M, I)$
$(shell_1, wait_M)$ $shell_1$
Guesser
$(shell_2, wait_M)$ $shell_2$
$(I, guess_1)$
$(I, guess_2)$
$(I, guess_1)$
$(I, guess_2)$
$(shell_1, guess_1)$ $shell_1, guess_1$
$(shell_1, guess_2)$ $shell_1, guess_2$
$(shell_2, guess_1)$ $shell_2, guess_1$
$(shell_2, guess_2)$ $shell_2, guess_2$

Figure 1: The IS $M$ for a revisited version of the Shell Game. Here, we consider the general setting in which there are $n$ steps of hiding before the choice of the Guesser.

function and the labelling function are given in Figure 1 for the case of $N = 2$. In particular, each global state is represented as a rectangle where the pair $(l_s, l_g)$ includes the Shuffler's local state $(l_s)$ and the Guesser's local state $(l_g)$. Further, in each rectangle, below the pair of local states, we have the true atoms in accordance with the labelling function.

The property "the Guesser has a winning strategy to guess the correct location of the ball" can be represented as follows:

$$\varphi_1 \quad = \quad \langle\!\langle Guesser \rangle\!\rangle F \varphi_{g\_win}$$

where $\varphi_{g\_win} = \bigvee_{i=1}^{N}(guess_i \wedge shell_i)$.

We observe that $\varphi_1$ is false w.r.t. memoryless strategies since to make the property true the Guesser is supposed to perform different actions in indistinguishable states. However, the Guesser has a $m+1$-bounded recall strategy to win the game. More formally, we have that $(M, s_0) \models_n^2 \varphi_1$ holds iff $n > m$.

**Example 2.** We consider the simple voting scenario presented by Jamroga et al. (2019a) comprising of $\ell$ voters, $k$ candidates, and a single coercer. Every voter $i \leq \ell$ votes in turn for one candidate $j \leq k$ (action $vote_{ij}$), and after casting her ballot, voter $i$ can either give a proof of vote to the coercer (action $give_{ij}$), or refrain from doing so (action $n\_give_i$), assuming the proof is trustworthy. The coercer receives the proof, and decides whether to punish voter $i$ or not (actions $punish_i$ and $n\_punish_i$). The decision is made $t$ timestamps after the proof is submitted by the voter (the coercer delays decision by performing the $wait$ action). More formally, this scenario can be represented as the IS $M = \langle Ag, s_0, T, \Pi\rangle$, such that $Ag = \{Coercer, Voter_1, \ldots, Voter_\ell\}$, $Act_{Coercer} = \{receive_1, \ldots, receive_\ell, punish_1, \ldots, punish_\ell, n\_punish_1, \ldots, n\_punish_\ell, wait, I\}$, where by action $receive_i$ the coercer receives the response from voter $i$, and $Act_{Voter_i} = \{vote_{i1}, \ldots, vote_{ik}, give_{i1}, \ldots, give_{ik}, n\_give_i, I\}$, where by action $n\_give_i$ the voter $i$ gives no proof, whereas by action $give_{ij}$ voter $i$ gives proof of having voted for candidate $j$. Finally, $I$ is the idle action. For the sake of clarity, the global transition function and the labelling function are given in Figure 2 in the case with a single voter, two candidates, and one waiting step. In particular, each global state is represented as a rectangle where the pair $(l_c, l_v)$ includes the coercer's local state ($l_c$) and the voter's local state ($l_v$). Further, in each rectangle, below the pair of local states, we have the atoms in accordance with the labelling function. Note that, since we have a single voter, in Figure 2 we omit the index $i$ for the voter's actions.

This IS $M$ is useful to analyse the expressive power of bounded-recall strategies. In particular, the property "for each voter $i$, for all the strategies for voter $i$, at the next step the coercer has a strategy such that voter $i$ is not punished if she votes for candidate 1 and provides the proof, otherwise she is punished" can be represented as follows:

$$\varphi_3 \;=\; \bigwedge_{i=1}^{\ell} [\![Voter_i]\!] X \langle\!\langle Coercer\rangle\!\rangle F((vote_{i1} \wedge n\_punish_i) \vee \\ (\bigvee_{j=2}^{k} vote_{ij} \wedge punish_i) \vee (n\_give_i \wedge punish_i))$$

We observe that $\varphi_3$ is false w.r.t. memoryless strategies, since for this property to hold, the coercer is supposed to perform two different actions in indistinguishable states (the states connected with dotted lines in Figure 2 are indistinguishable for the coercer). However, the coercer has a $t+1$-bounded recall strategy to win the game, where $t$ is the number of waiting steps. More formally, we have that $(M, s_0) \models^2_n \varphi_3$ holds iff $n > t$.

### 2.3 Model Checking Bounded Recall

We now analyse the model checking problem for bounded recall within the two-valued semantics, defined as follows.

**Definition 7** (Model Checking)**.** The *model checking (MC) problem* concerns determining whether, given an IS $M$, $ATL^*$ formula $\phi$, bound $n \in \mathbb{N}^+ \cup \{\omega\}$, truth value $v \in \{\text{tt}, \text{ff}\}$, it is the case that $(M \models^2_n \phi) = v$.

Fix a constant $n \in \mathbb{N}^+ \cup \{\omega\}$, the *n-fixed-recall MC problem* concerns determining whether, given an IS $M$, $ATL^*$ formula $\phi$, truth value $v \in \{\text{tt}, \text{ff}\}$, it is the case that $(M \models^2_n \phi) = v$.

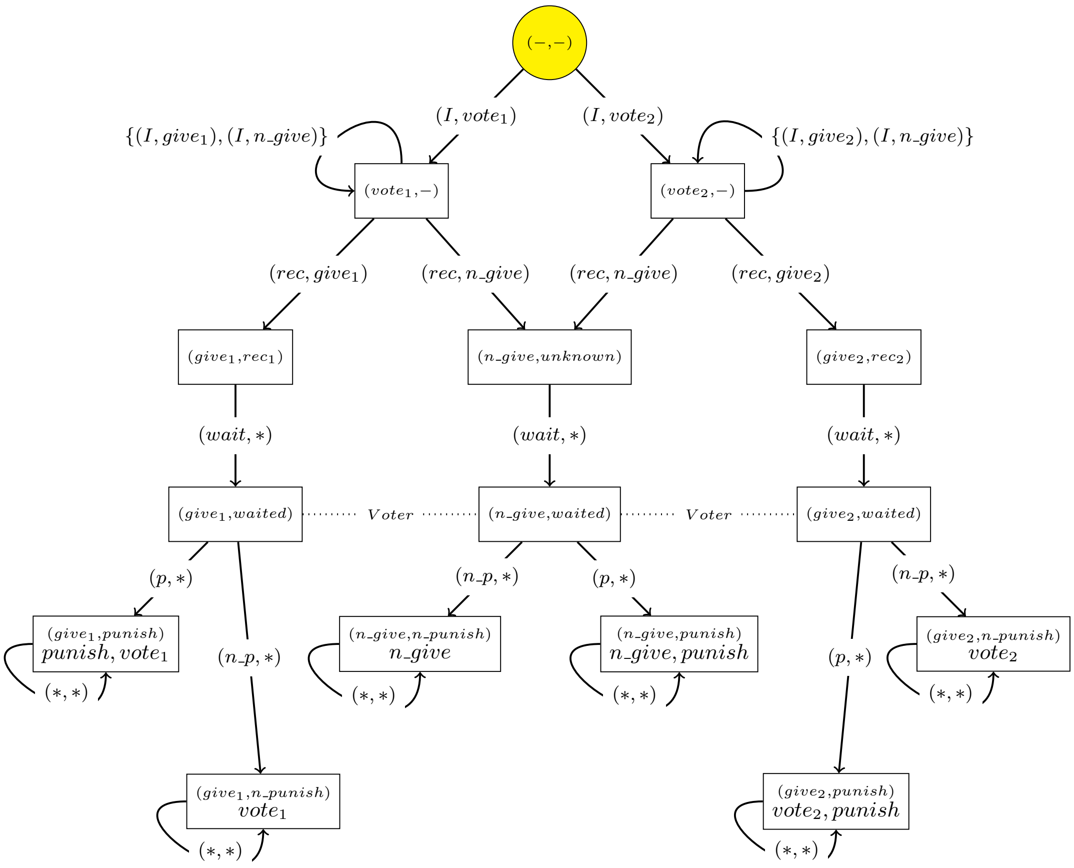

Figure 2: The IS $M$ for the simple voting scenario. We consider the setting of one waiting step before the Coercer makes a decision for punishment. Here, action $*$ represents any action available for the agent.

We show that the model checking $ATL$ with perfect recall (i.e., $n$-fixed-recall for $n = \omega$) and imperfect information is undecidable.

**Theorem 1.** The $\omega$-fixed-recall model checking problem for $ATL$ on the two-valued semantics with imperfect information is undecidable.

**Proof.** Dima & Tiplea (2011) prove that the model checking problem for $ATL$ with perfect recall (i.e., $n$-fixed-recall for $n = \omega$) over concurrent game structure with imperfect information (iCGS) is undecidable. Then, the result follows from the fact that IS and iCGS can be translated one into the other in polynomial time. Specifically, every IS $M$ induces an iCGS $G_M$ that satisfies exactly the same formulas in $ATL^*$. For the other direction, given a iCGS $G$ satisfying some non-restrictive condition, called *being square* by Belardinelli et al. (2020a) (which is fulfilled by the iCGS used in the undecidability proof by Dima & Tiplea

(2011)), we can extract an IS $M_G$ such that $G$ and $M_G$ satisfy the same formulas in $ATL^*$. The details of both translations can be found in Belardinelli et al. (2020a). □

As an immediate consequence of Theorem 1, model checking $ATL^*$ under the same conditions is also undecidable. We record these results in the following corollary.

**Corollary 1.** The $\omega$-fixed-recall model checking problem for $ATL^*$ on the two-valued semantics with imperfect information is undecidable.

In contrast we show that model checking $ATL^*$ with bounded recall and imperfect information is decidable.

**Theorem 2.** For $n \in \mathbb{N}^+$, the model checking problem for $ATL^*$ under $n$-bounded recall and imperfect information is in *EXPTIME*. Moreover, the corresponding $n$-fixed-recall MC problem is *PSPACE*-complete.

**Proof.** First, we provide the upper bound for the general case. In particular, given an IS $M = \langle Ag, s_0, T, \Pi\rangle$, a formula $\phi$, and a bound $n \in \mathbb{N}^+$, we describe a labelling algorithm to decide the corresponding model checking instance and we show it to be in *EXPTIME*. Given an agent $i = \langle L_i, Act_i, P_i, t_i\rangle$ in $Ag$ and $n \in \mathbb{N}^+$, we define a new agent $i' = \langle L_i^{\leq n}, Act_i, P'_i, t'_i\rangle$ such that $L_i^{\leq n}$ is the set of sequences $h$ of local states in $L_i$ of length at most $n$. Then, for every sequence $h \in L_i^{\leq n}$, $a \in P'_i(h)$ iff $a \in P_i(last(h))$. Finally, $h' = t'_i(h, a)$ iff $|h'| = |h| + 1$; for every $j \leq |h|$, $h'_j = h_j$; and $last(h') = t_i(last(h), a)$. Then, consider IS $M' = \mathtt{Inflate}(M, n) = \langle Ag', s_0, T', \Pi'\rangle$, where $Ag'$ is the set of all and only agents $i'$ defined as above from agents $i \in Ag$, and $\Pi'(h, q) = \Pi(last(h), q)$. That is, the states in $M'$ are the histories in $M$ of length at most $n$, and transitions and assignments in $M'$ mirrors those in $M$. Clearly, the size $|M'|$ of IS $M'$ defined as the number $|\mathcal{G}'|$ of states, is exponential in the size $|M| = |\mathcal{G}|$ of the original IS $M$, that is $M' = |G|^n$. Moreover, by induction on the structure of formulas in $ATL^*$ we can prove the following result:

**Lemma 1.** For every formula $\varphi$ in $ATL^*$, state $s$ in $M$, and history $h$ in $M'$ such that $last(h) = s$, we have

$$(M, s) \models^2_n \varphi \quad \text{iff} \quad (M', h) \models^2_1 \varphi$$

The base of induction is immediate as, for $\varphi = q$, $(M, s) \models^2_n \varphi$, iff $\Pi(s, q) = \text{tt} = \Pi'(h, q)$, iff $(M', h) \models^2_1 \varphi$. The inductive cases for Boolean connectives are also immediate. The case of interest is obviously for formulas of type $\varphi = \langle\!\langle \Gamma \rangle\!\rangle \psi'$. In this case, the result follows by the remark that a (uniform) strategy with $n$-bounded recall defined on states in $M$ is the same as a (uniform) memoryless strategy defined on histories in $M'$. This completes the proof of the lemma.

By Lemma 1, to determine whether $(M, s) \models^2_n \varphi$, it is sufficient to model check $\varphi$ on IS $M' = \mathtt{Inflate}(M, n)$ under the assumptions of imperfect information and imperfect recall, as shown in Figure 3. The latter problem is known to be in *PSPACE* (Schobbens, 2004). Hence, the whole procedure is in *EXPTIME*, as it is dominated by the construction of IS $M'$.

On the other hand, if we consider model checking $ATL^*$ for a fixed bound $n \in \mathbb{N}^+$, we obtain a *PSPACE* upper bound. To prove this, we consider the general procedure provided

```
Algorithm MC($M, \varphi, n$):
$M' =$ Inflate($M, n$);
return MC_ATL$^*_{\mathtt{ir}}$($M', \varphi$);
```

Figure 3: Algorithm to decide $ATL^*$ Model checking.

above applied for a given bound $n$. Fixing $n$, the size of $M' = |G|^n$ becomes polynomial in the size of the input. This removes the exponential blow-up in the construction of IS $M'$ from $M$, and therefore, all we need to consider is the complexity of model checking $ATL^*$ formulas under imperfect information and imperfect recall. We know this to be in *PSPACE*. As for the lower bound with $n$ fixed, it follows by the complexity of model checking formulas in linear-time temporal logic ($LTL$), which is known to be *PSPACE*-hard. □

As regards the $ATL$ fragment of $ATL^*$, we prove the following result.

**Theorem 3.** For $n \in \mathbb{N}^+$, the model checking problem for $ATL$ under $n$-bounded recall and imperfect information is in *EXPTIME*. Moreover, the corresponding $n$-fixed-recall MC problem is $\Delta_2^P$-complete.

**Proof.** The upper bound for the general case follows immediately from Theorem 2.

As regards the upper bound for $ATL$ with a fixed $n \in \mathbb{N}^+$, we adapt the proof for $ATL^*$ described above. Specifically, model checking $ATL$ under imperfect information and imperfect recall is known to be in $\Delta_2^P$ (Jamroga & Dix, 2006). This complexity dominates the procedure of inflating and model-checking, once the value $n \in \mathbb{N}^+$ has been fixed. As for the lower bound, we can use the same reduction to the problem *SNSAT*$_2$ of sequential satisfiability as in Jamroga & Dix (2006). □

We remark that for $n \in \mathbb{N}^+$, the complexity of $n$-fixed-recall model checking $ATL$ and $ATL^*$ with $n$-bounded recall (and imperfect information) is the same as for the imperfect recall case, that is, for $n = 1$ (Schobbens, 2004; Jamroga & Dix, 2006). Moreover, we provided tight complexity results only for a fixed $n$. Indeed, here we are mainly interested in the fact that, differently from the case of perfect recall, the model checking problem for bounded recall is decidable, irrespectively of its actual complexity (which we believe to be also *EXPTIME*-hard, but outside the scope of the present contribution).

The decidability results above can be the basis of a partial model checking procedure for perfect recall consisting in increasing the bound $n$ on the recall of agents. However, as the following demonstrates, increasing recall only preserves rather limited fragments of $ATL^*$ and may, therefore, only be of limited interest.

**Lemma 2.** Let $m, n \in \mathbb{N}^+ \cup \{\omega\}$ be such that $m \leq n$; let $\psi$ be an existential and $\phi$ an universal formula in $ATL^*$. Then,

$$(M, p) \models^2_m \psi \quad \Rightarrow \quad (M, p) \models^2_n \psi \qquad (1)$$

$$(M, p) \not\models^2_m \phi \quad \Rightarrow \quad (M, p) \not\models^2_n \phi \qquad (2)$$

**Proof.** The proofs for (1) and (2) are both by induction on the structure of the formula. We only consider the case where the main operator is the strategic modality. The other cases are immediate and thus omitted.

(1) By Def. 6 $(M, s) \models^2_m \langle\langle\Gamma\rangle\rangle\psi$ iff for some joint strategy $F^m_\Gamma$, for all paths $p \in out(s, F^m_\Gamma)$, $(M, p) \models^2_m \psi$. Given $F^m_\Gamma$ we construct a set $F^n_\Gamma$ of $n$-bounded recall strategies as follows: for every agent $i \in \Gamma$ and history $h \in \mathcal{G}^{<1+n}$, define $f^n_i(h) = f^m_i(h_{(|h|-m)}, \ldots, h_{|h|})$ for $m < |h|$, $f^n_i(h) = f^m_i(h)$ otherwise. Notice that each $f^n_i$ so defined is uniform, provided that $f^m_i$ is. Given such $F^n_\Gamma$, we obtain that $out(s, F^n_\Gamma) = out(s, F^m_\Gamma)$. In particular, for all paths $p \in out(s, F^n_\Gamma)$, $(M, p) \models^2_m \psi$ implies $(M, p) \models^2_n \psi$ by induction hypothesis, and therefore $(M, s) \models^2_n \langle\langle\Gamma\rangle\rangle\psi$.

(2) By Def. 6 $(M, s) \not\models^2_m [\![\Gamma]\!]\phi$ iff for some joint strategy $F^m_\Gamma$, for all paths $p \in out(s, F^m_\Gamma)$, $(M, p) \not\models^2_m \phi$. Given $F^m_\Gamma$ we can construct a set $F^n_\Gamma$ of strategies as in point (1). Again, each $f^n_i$ so defined is uniform, provided that $f^m_i$ is. Given $F^n_\Gamma$ thus defined, we obtain that $out(s, F^n_\Gamma) = out(s, F^m_\Gamma)$. In particular, for all paths $p \in out(s, F^n_\Gamma)$, $(M, p) \not\models^2_m \phi$ implies $(M, p) \not\models^2_n \phi$ by induction hypothesis, and therefore $(M, s) \not\models^2_n [\![\Gamma]\!]\phi$. □

By Lemma 2 adding memory preserves the truth of existential formulas as well as falsehood of universal formulas. However, it is not difficult to find counterexamples to the extensions of (1) and (2) even in $ATL$.

**Lemma 3.** Let $m, n \in \mathbb{N}^+ \cup \{\omega\}$ be such that $m < n$. There exists formulas $\varphi$ and $\varphi' = \neg\varphi$ in $ATL$ such that

$$(M, p) \not\models^2_m \varphi \quad \text{and} \quad (M, p) \models^2_n \varphi \tag{3}$$

$$(M, p) \models^2_m \varphi' \quad \text{and} \quad (M, p) \not\models^2_n \varphi' \tag{4}$$

**Proof.** We only provide a proof for (3). Then, (4) follows immediately by considering $\varphi' = \neg\varphi$. Consider the revisited version of the Shell Game, as described in Example 1. Let $\varphi = \langle\langle Guesser\rangle\rangle F\varphi_{g\_win}$, where $\varphi_{g\_win} = \bigvee_{i=1}^{N}(guess_i \wedge shell_i)$ and $m, n \in \mathbb{N}^+ \cup \{\omega\}$ with $m < M + 1 \leq n$. Clearly, the Guesser has no $m$-bounded recall strategy to win the game, but she has a $n$-bounded recall strategy. □

By Lemmas 2 and 3 any naive attempt to approximate perfect recall by increasing bounded recall is severely restricted in two ways. Firstly, Lemma 2 holds only for the existential and universal fragments of $ATL^*$. Secondly, only the truth of existential formulas is preserved by adding memory, whereas negative results can only be lifted for the universal fragment. In Section 3, we present a three-valued semantics to overcome these difficulties.

### 2.4 A Comparison between Bounded Recall and Deterministic Finite-State Transducers

The treatment of strategies with finite memory was put forward by Vester (2013). We here compare that approach to the one here pursued. Vester (2013) represents finite-memory strategies as *deterministic finite-state transducers* (DFST) [3].

We show that bounded strategies and DFST cannot always be translated (polynomially) into the other; hence , the two formalisms are orthogonal. We begin by introducing the definition of DFST, but refer to Vester (2013) for more details.

[3] We remark that the structures defined as DFST by Vester (2013) are actually Mealy machines. For clarity, we keep the original terminology in this section, as Mealy machines are particular versions of DFSTs. This slight looseness of terminology does not affect the validity of the results presented.

**Definition 8** (DFST)**.** A *deterministic finite-state transducers* is a tuple $D = \langle V, v_0, In, Out, F_{in}, F_{out}\rangle$, where

- $V$ is a finite non-empty set of *states*, with *initial state* $v_0$;
- $In$ is the *input alphabet*;
- $Out$ is the *output alphabet*;
- $F_{in} : V \times In \to V$ is the *transition function*;
- $F_{out} : V \times In \to Out$ is the *output function.*

When strategies are represented as DFST, the set $V$ of states can be seen as the possible values of the internal memory of the strategy, and the initial state $v_0$ corresponds to the initial memory value. The input symbols in $In$ are the states of the interpreted system, and the output symbols in $Out$ are its actions. In each round of the strategy execution the DFST reads the current state. Then, it updates its memory based on the current memory value and the input state according to $F_{in}$. Finally, it outputs an action based on the current memory value and the input state according to $F_{out}$.

A function $\sigma : \mathcal{G}^+ \to Act$ is a *finite-memory strategy* if there exists a DFST such that for all histories $h \in \mathcal{G}^+$:

$$\sigma(h) = F_{out}(G(v_0, h_{\leq |h|-1}), last(h))$$

where for every state $v$ and history $h$, function $G$ is defined recursively as follows:

$$G(v, h) = \begin{cases} F_{in}(v, h) & \text{for } |h| = 1; \\ F_{in}(G(v, h_{\leq |h|-1}), last(h)) & \text{otherwise.} \end{cases}$$

That is, $G$ is the function that repeatedly applies the transition function $F_{in}$ on a sequence of inputs to calculate the state of the DFST after reading a given history $h$.

We now compare formally our definition of bounded strategy with finite-memory strategies given via DFST. Hereafter we say that two strategies (possibly given via DFST) are *equivalent* if they correspond to the same function $\sigma : \mathcal{G}^+ \to Act$. In the rest of this section we say that an IS $M$ is *non-trivial* if some agent has at least two states and two actions.

**Proposition 1.** Given a non-trivial IS $M$, for every bound $n \in \mathbb{N}^+$, there exists some DFST $D$ for which there is no equivalent strategy with $g(n)$-bounded memory, for any polynomial function $g$.

**Proof.** We construct a DFST $D$ such that for some history $h$ of length exponential in $n$, and different states $s$, $s'$ in $M$, it is the case that $F_{out}(G(v_0, s \cdot h_{\leq |h|-1}), last(h)) \neq F_{out}(G(v_0, s' \cdot h_{\leq |h|-1}), last(h))$. Specifically, consider the DFST $D = \langle \{v_0, v_1, v_2\}, v_0, S, Act, F_{in}, F_{out}\rangle$ such that:

1. for all $\bar{s} \in \mathcal{G} \setminus \{s, s'\}$, $F_{in}(v_0, \bar{s}) = v_0$;
2. $F_{in}(v_0, s) = v_1$ and $F_{in}(v_0, s') = v_2$;

3. for all $\bar{s} \in \mathcal{G}$, $F_{in}(v_1, \bar{s}) = v_1$ and $F_{in}(v_2, \bar{s}) = v_2$;

4. for all $\bar{s} \in \mathcal{G}$, $F_{out}(v_1, \bar{s}) = a$ and $F_{out}(v_2, \bar{s}) = b$, where $a, b \in Act$ and $a \neq b$ (these exists as $M$ is non-trivial by assumption).

By the construction above, we ensure that $F_{out}(G(v_0, s \cdot h_{\leq |h|-1}), last(h)) \neq F_{out}(G(v_0, s' \cdot h_{\leq |h|-1}), last(h))$. In fact, by (1) from the initial state $v_0$ of $D$ by reading $s$ (resp., $s'$), the DFST $D$ goes in $v_1$ (resp., $v_2$). By (2), from $v_1$ (resp., $v_2$) by reading any state of the IS, $D$ stays on $v_1$ (resp., $v_2$). By (3), we have $F_{out}(G(v_0, s \cdot h_{\leq |h|-1}), last(h)) = a$ and $F_{out}(G(v_0, s \cdot h_{\leq |h|-1}), last(h)) = b$, and therefore $F_{out}(G(v_0, s \cdot h_{\leq |h|-1}), last(h)) \neq F_{out}(G(v_0, s' \cdot h_{\leq |h|-1}), last(h))$ as required.

However, this memory-bounded strategy cannot be captured by any strategy whose recall is bounded by some polynomial $g(n)$. In fact, by hypothesis history $h$ is exponential in $n$, and since any bounded-recall strategy only considers the last $g(n)$ states of histories $s \cdot h$ and $s' \cdot h$ respectively at most, then it returns the same action for both of them. □

As regards translating bounded-recall strategies into DFST we have the following result.

**Proposition 2.** Given a non-trivial IS $M$, for every bound $n \in \mathbb{N}$, there exists some $n$-bounded recall strategy $f$ for which there is no equivalent DFST with $g(n)$ states, for any polynomial function $g$.

**Proof.** We provide a proof by contradiction. Given an $n$-bounded recall strategy $f$, suppose that we can always construct a DFST $D$ with $m$ states, where $m < |S|^{n-1}$. In particular, by considering all the possible $|S|^n$ histories of length $n$ in $M$ [4], by the pigeonhole principle there are at least two different histories $h$ and $h'$ in $\mathcal{G}^n$, in which at some points $k, j \leq n$, the function $G$ returns the same state of memory, that is, $G(v_0, h_{\leq j}) = G(v_0, h'_{\leq k})$.

Suppose further that $h_j = h'_k$, then we have that $F_{out}(G(v_0, h_{\leq j-1}), h_j) = F_{out}(G(v_0, h'_{\leq k-1}), h'_k)$, that is, the same action is returned when reading histories $h_{\leq j}$ and $h'_{\leq k}$. Since we supposed that our strategies have $n$-bounded recall, then w.l.o.g. we can assume that $f$ assigns different actions to $h_{\leq j}$ and $h'_{\leq k}$. But this contradicts the fact that $m < |S|^{n-1}$ states of memory in a DFST are sufficient to describe a $n$-bounded strategy. □

In other words, Proposition 2 states that an $n$-bounded strategy can in principle return a different action for every history of length $n$, that is, $|S|^n$ different actions. But to do this in a DFST $D$, we might need $|S|^n$ states.

Intuitively, Propositions 1 and 2 lead to the following observations. While DFST can be seen as representing strategies with finite memory, bounded strategies as here introduced express recall. Memory and recall are related, but orthogonal notions.

## 3. Three-Valued Bounded Recall

In Section 2 we remarked that model checking $ATL^*$ under imperfect information and perfect recall is undecidable in general (Dima & Tiplea, 2011). Moreover, any naive attempt to approximate perfect recall by increasing bounded recall is severely restricted by the results

[4] Note that, in a non-trivial IS we have $|S|^n$ different histories of length $n$, this is because we define histories as any sequence of states, without considering the transition function (see Subsection 2.1).

in Lemma 2 and 3 in two dimensions. Firstly, Lemma 2 holds only for the existential and universal fragments of $ATL^*$. Secondly, only the truth of existential formulas is preserved by adding memory, whereas negative results can only be lifted for the universal fragment.

To tackle these issues, in this section we lay the theoretical foundations of a partial model checking procedure based on a three-valued semantics. The procedure is partial as in some cases it returns "undefined" (uu) as truth value. On the other hand, differently from Lemma 2, the satisfaction of all $ATL^*$ formulas is preserved by adding memory.

### 3.1 Three-Valued ATL with Bounded Recall

We start by providing the three-valued satisfaction relation for $ATL^*$, where we consider a third truth value uu 'undefined', besides truth tt and falsehood ff.

First of all, in the rest of the paper we consider an *interpreted system* as a tuple $M = \langle Ag, s_0, T, \Pi \rangle$, where $Ag$, $s_0$, and $T$ are defined as in Def. 2, whereas a labelling is now a function $\Pi : \mathcal{G} \times AP \to \{\mathrm{tt}, \mathrm{ff}, \mathrm{uu}\}$. Notice that an IS $M$ according to Def. 2 is in particular an IS as here defined, simply all atoms are either true or false.

**Definition 9** (Three-valued Satisfaction)**.** Let $n \in \mathbb{N}^+ \cup \{\omega\}$. The three-valued satisfaction relation $\models^3_n$ for an IS $M$, state $s$, path $p$, $ATL^*$ formula $\phi$, and $v \in \{\mathrm{tt}, \mathrm{ff}\}$ is defined as follows, where $\neg\mathrm{tt} = \mathrm{ff}$ and $\neg\mathrm{ff} = \mathrm{tt}$:

$$
\begin{array}{lll}
((M,s) \models^3_n q) = v & \text{iff} & \Pi(s,q) = v \\
((M,s) \models^3_n \neg\varphi) = v & \text{iff} & ((M,s) \models^3_n \varphi) = \neg v \\
((M,s) \models^3_n \varphi \wedge \varphi') = \mathrm{tt} & \text{iff} & ((M,s) \models^3_n \varphi) = \mathrm{tt} \text{ and } ((M,s) \models^3_n \varphi') = \mathrm{tt} \\
((M,s) \models^3_n \varphi \wedge \varphi') = \mathrm{ff} & \text{iff} & ((M,s) \models^3_n \varphi) = \mathrm{ff} \text{ or } ((M,s) \models^3_n \varphi') = \mathrm{ff} \\
((M,s) \models^3_n \langle\langle\Gamma\rangle\rangle\psi) = \mathrm{tt} & \text{iff} & \text{for some } F^n_\Gamma, \text{ for all } p \in out(s, F^n_\Gamma),\ ((M,p) \models^3_n \psi) = \mathrm{tt} \\
((M,s) \models^3_n \langle\langle\Gamma\rangle\rangle\psi) = \mathrm{ff} & \text{iff} & \text{for some } F^n_{\overline{\Gamma}}, \text{ for all } p \in out(s, F^n_{\overline{\Gamma}}),\ ((M,p) \models^3_n \psi) = \mathrm{ff} \\
((M,p) \models^3_n \varphi) = v & \text{iff} & ((M,p_1) \models^3_n \varphi) = v \\
((M,p) \models^3_n \neg\psi) = v & \text{iff} & ((M,p) \models^3_n \psi) = \neg v \\
((M,p) \models^3_n \psi \wedge \psi') = \mathrm{tt} & \text{iff} & ((M,p) \models^3_n \psi) = \mathrm{tt} \text{ and } ((M,p) \models^3_n \psi') = \mathrm{tt} \\
((M,p) \models^3_n \psi \wedge \psi') = \mathrm{ff} & \text{iff} & ((M,p) \models^3_n \psi) = \mathrm{ff} \text{ or } ((M,p) \models^3_n \psi') = \mathrm{ff} \\
((M,p) \models^3_n X\psi) = v & \text{iff} & ((M,p_{\geq 2}) \models^3_n \psi) = v \\
((M,p) \models^3_n \psi U \psi') = \mathrm{tt} & \text{iff} & \text{for some } k \geq 1,\ ((M,p_{\geq k}) \models^3_n \psi') = \mathrm{tt}, \text{ and} \\
 & & \text{for all } j,\ 1 \leq j < k \text{ implies } ((M,p_{\geq j}) \models^3_n \psi) = \mathrm{tt} \\
((M,p) \models^3_n \psi U \psi') = \mathrm{ff} & \text{iff} & \text{for all } k \geq 1, \text{ either } ((M,p_{\geq k}) \models^3_n \psi') = \mathrm{ff} \\
 & & \text{or for some } j,\ 1 \leq j < k \text{ and } ((M,p_{\geq j}) \models^3_n \psi) = \mathrm{ff}.
\end{array}
$$

In all other cases the value of $\phi$ is undefined (uu).

For clarity, we also state the derived meaning of formulas $[\![\Gamma]\!]\psi ::= \neg\langle\langle\Gamma\rangle\rangle\neg\psi$:

$$
\begin{array}{lll}
((M,s) \models^3_n [\![\Gamma]\!]\psi) = \mathrm{tt} & \text{iff} & \text{for some } F^n_{\overline{\Gamma}}, \text{ for all } p \in out(s, F^n_{\overline{\Gamma}}),\ (M,s) \models^3_n \psi) = \mathrm{tt} \\
((M,s) \models^3_n [\![\Gamma]\!]\psi) = \mathrm{ff} & \text{iff} & \text{for some } F^n_{\Gamma}, \text{ for all } p \in out(s, F^n_{\Gamma}),\ ((M,s) \models^3_n \psi) = \mathrm{ff}
\end{array}
$$

Notice that all clauses for the three-valued semantics mirror the corresponding two-valued clauses, with a notable exception: for $\langle\langle\Gamma\rangle\rangle\psi$ to be false we require the existence of a joint strategy for the complement coalition $\overline{\Gamma} = Ag \setminus \Gamma$ that enforces $\psi$ to be false. Similar conditions have previously been proposed (Lomuscio & Michaliszyn, 2014). It is a stronger requirement than the usual clause on the coalition $\Gamma$ not being able to enforce $\psi$ (see Def. 6). However, it has the advantage of being preserved when adding memory, as it will become apparent in Lemma 7. Further, as for the two-valued semantics, we normally refer to the cases for $n = 1$ and $n = \omega$ as imperfect, resp. perfect, recall. Also notice that, as regards the Boolean operators, our semantics correspond to Kleene's three-valued logic.

We say that formula $\varphi$ is *true* (resp. *false*) in an IS $M$ (for $n$-bounded recall), or $(M \models^3_n \varphi) = \text{tt}$ (resp. ff), iff $((M, s_0) \models^3_n \varphi) = \text{tt}$ (resp. ff); otherwise $\varphi$ is *undefined*. Again, we observe that Def. 9 corresponds to the subjective, three-valued interpretation of $ATL^*$. The corresponding objective semantics can be obtained with minor modification, but it is beyond the scope of the present contribution.

We immediately prove that the three-valued notion of satisfaction in Def. 9 is an extension of the two-valued relation in Def. 6, in the sense that truth and falsehood in the three-valued semantics correspond respectively to truth and falsehood in the two-valued one.

**Lemma 4.** For every $n \in \mathbb{N}^+ \cup \{\omega\}$, formula $\phi$ in $ATL^*$,

$$((M, s) \models^3_n \phi) = \text{tt} \quad \Rightarrow \quad (M, s) \models^2_n \phi \tag{5}$$

$$((M, s) \models^3_n \phi) = \text{ff} \quad \Rightarrow \quad (M, s) \not\models^2_n \phi \tag{6}$$

**Proof.** The proofs for both (5) and (6) are by simultaneous induction on the structure of the formula. We present the case where the main operator is the strategic modality. The cases for the other operators are immediate.

(5) By Def. 9 $((M, s) \models^3_n \langle\langle\Gamma\rangle\rangle\psi) = \text{tt}$ iff for some joint strategy $F^n_\Gamma$, for all paths $p \in out(s, F^n_\Gamma)$, $((M, p) \models^3_n \psi) = \text{tt}$. Fix such a joint strategy $F^n_\Gamma$. By induction hypothesis we obtain that for all paths $p \in out(s, F^n_\Gamma)$, $(M, p) \models^2_n \psi$. Then, $(M, s) \models^2_n \langle\langle\Gamma\rangle\rangle\psi$ as required.

(6) By Def. 9 $((M, s) \models^3_n \langle\langle\Gamma\rangle\rangle\psi) = \text{ff}$ iff for some joint strategy $F^n_{\overline{\Gamma}}$, for all paths $p \in out(s, F^n_{\overline{\Gamma}})$, $((M, p) \models^3_n \psi) = \text{ff}$. Fix such a joint strategy $F^n_{\overline{\Gamma}}$. By induction hypothesis we obtain that for all paths $p \in out(s, F^n_{\overline{\Gamma}})$, $(M, p) \not\models^2_n \psi$. In particular, for every joint strategy $F^n_\Gamma$ we can construct some path $p' \in out(s, F^n_\Gamma)$ (which is obtained when coalition $\overline{\Gamma}$ plays according to $F^n_{\overline{\Gamma}}$) such that $(M, p') \not\models^2_n \psi$ by hypothesis. As a result, $(M, s) \not\models^2_n \langle\langle\Gamma\rangle\rangle\psi$. □

On the other hand, the three-valued semantics is not a *conservative* extension of the two-valued one, in the sense that truth and falsehood in the two-valued semantics might sometimes correspond to undefined uu in the three-valued one. Specifically, the following lemma provides counterexamples to the converse of (5) and (6).

**Lemma 5.** For $n \in \mathbb{N}^+ \cup \{\omega\}$, there exists an IS $M$ with state $s$, and $ATL$ formulas $\varphi$ and $\varphi' = \neg\varphi$ such that

$$(M, s) \models^2_n \varphi \quad \text{and} \quad ((M, s) \models^3_n \varphi) = \text{uu} \tag{7}$$

$$(M, s) \not\models^2_n \varphi' \quad \text{and} \quad ((M, s) \models^3_n \varphi') = \text{uu} \tag{8}$$

**Proof.** As regards (7) consider again the Shell Game with $n$ hidden steps presented in Example 1. We remarked therein that $\langle\langle Guesser\rangle\rangle F\varphi_{g\_win}$, where $\varphi_{g\_win} = \bigvee_{i=1}^{N}(guess_i \wedge$

*shell*$_i$), is false in the $n$-bounded, two-valued semantics, and therefore $(M, s_1) \models_n^2 \varphi$, for $\varphi = \neg(\langle\!\langle Guesser \rangle\!\rangle F \varphi_{g\_win})$. However, in the same game the Shuffler has no $n$-bounded strategy to enforce the Guesser to lose, that is, $((M, s) \models_n^3 \langle\!\langle Guesser \rangle\!\rangle F \varphi_{g\_win}) \neq \mathrm{ff}$ (actually, the value is uu), and therefore $((M, s) \models_n^3 \varphi) \neq \mathrm{tt}$.

To check (8) it is sufficient to take $\varphi' = \neg\varphi$. Then, $(M, s_1) \not\models_n^2 \varphi'$ and $((M, s_1) \models_n^3 \varphi') =$ uu. □

By Lemma 4 and 5 the three-valued semantics can be thought of as an approximation of the two-valued one, as defined truth values in the former correspond to the same values in the latter, but not always viceversa.

### 3.2 The Complexity of Model Checking

We now analyse the model checking problem for the three-valued semantics.

**Definition 10** (Three-valued Model Checking)**.** The *model checking (MC) problem* concerns determining whether, given an IS $M$, $ATL^*$ formula $\phi$, bound $n \in \mathbb{N}^+ \cup \{\omega\}$, truth value $v \in \{\mathrm{tt}, \mathrm{ff}, \mathrm{uu}\}$, it is the case that $(M \models_n^3 \phi) = v$.

Fix a constant $n \in \mathbb{N}^+ \cup \{\omega\}$, the *n-fixed-recall MC problem* concerns determining whether, given an IS $M$, $ATL^*$ formula $\phi$, truth value $v \in \{\mathrm{tt}, \mathrm{ff}, \mathrm{uu}\}$, it is the case that $(M \models_n^3 \phi) = v$.

Similarly as in the two-valued semantics, we immediately obtain the following undecidability result.

**Theorem 4.** The $\omega$-fixed-recall model checking problem for $ATL$ on the three-valued semantics with imperfect information is undecidable.

**Proof.** The proof again follows by adapting the undecidability result by Dima & Tiplea (2011), which makes use of the $ATL$ formula $\varphi = \langle\!\langle \{1, 2\} \rangle\!\rangle G ok$ to express that a Turing machine does not halt on the empty word. Specifically, we observe that the two- and three-valued interpretations coincide for this particular formula $\varphi$ on the iCGS $M_T$ introduced by Dima & Tiplea (2011) to represent the execution of a Turing machine $T$. That is, we have that $(M_T \models_\omega^3 \varphi) = \mathrm{tt}$ iff $M_T \models_\omega^2 \varphi$. Indeed, the value of atom *ok* is always defined, and the structure of the clauses for operator $\langle\!\langle \{1, 2\} \rangle\!\rangle$ being true is the same in the two- and three-valued semantics. As a consequence, we obtain that a Turing machine $T$ does not halt on the empty word iff $(M_T \models_\omega^3 \varphi) = \mathrm{tt}$. □

Given the above, note that the MC problem is also undecidable.

By Theorem 4, model checking $ATL^*$ under the same assumptions is also undecidable. But again, by assuming bounded recall we retrieve decidability. To present this result, we make use of two auxiliary procedures to update the model and the formula, in order to handle three-valued atoms. In particular, given a model $M$ we use the procedure `Duplicate_atoms`$(M)$ to produce a new model $M'$ that differs from $M$ as for atoms and the labeling function as follows:

1. For each atom $q \in AP$, the procedure generates two new atoms $q_{\mathrm{tt}}$ and $q_{\mathrm{ff}}$ and add them to the new set of atoms $AP' = \{q_{\mathrm{tt}}, q_{\mathrm{ff}} \mid q \in AP\}$.

```
Algorithm Transl($\varphi, v$):

switch($\varphi$)
 case $\varphi = q$:
  switch($v$)
   case $v = \mathrm{tt}$: return $q_{\mathrm{tt}}$;
   case $v = \mathrm{ff}$: return $q_{\mathrm{ff}}$;
 case $\varphi = \neg\varphi'$:
  switch($v$)
   case $v = \mathrm{tt}$: return Transl($\varphi', \mathrm{ff}$);
   case $v = \mathrm{ff}$: return Transl($\varphi', \mathrm{tt}$);
 case $\varphi = \varphi' \wedge \varphi''$:
  switch($v$)
   case $v = \mathrm{tt}$: return Transl($\varphi', \mathrm{tt}$) $\wedge$ Transl($\varphi'', \mathrm{tt}$);
   case $v = \mathrm{ff}$: return Transl($\varphi', \mathrm{ff}$) $\vee$ Transl($\varphi'', \mathrm{ff}$);
 case $\varphi = \langle\langle\Gamma\rangle\rangle\psi$:
  switch($v$)
   case $v = \mathrm{tt}$: return $\langle\langle\Gamma\rangle\rangle$ Transl($\psi, \mathrm{tt}$);
   case $v = \mathrm{ff}$: return $\langle\langle\bar{\Gamma}\rangle\rangle$ Transl($\psi, \mathrm{ff}$);
 case $\varphi = X\psi$:
  switch($v$)
   case $v = \mathrm{tt}$: return $X$ Transl($\psi, \mathrm{tt}$);
   case $v = \mathrm{ff}$: return $X$ Transl($\psi, \mathrm{ff}$);
 case $\varphi = \psi U \psi'$:
  switch($v$)
   case $v = \mathrm{tt}$: return Transl($\psi, \mathrm{tt}$) $U$ Transl($\psi', \mathrm{tt}$);
   case $v = \mathrm{ff}$: return Transl($\psi, \mathrm{ff}$) $R$ Transl($\psi', \mathrm{ff}$);
 case $\varphi = \psi R \psi'$:
  switch($v$)
   case $v = \mathrm{tt}$: return Transl($\psi, \mathrm{tt}$) $R$ Transl($\psi', \mathrm{tt}$);
   case $v = \mathrm{ff}$: return Transl($\psi, \mathrm{ff}$) $U$ Transl($\psi', \mathrm{ff}$);
```

Figure 4: Translation of a formula $\varphi$ to verify the truth value $v$.

2. For each state $s \in \mathcal{G}$, the procedure defines the labeling function $\Pi'$ on $s$ as $\Pi'(s) = \{q_{\mathrm{tt}} \mid q \in AP \text{ and } q \in \Pi(s)\} \cup \{q_{\mathrm{ff}} \mid q \in AP \text{ and } q \notin \Pi(s)\}$.

Since the following results are not dependent on a particular bound $n \in \mathbb{N}^+ \cup \{\omega\}$ assumed, in the following we fix $n$ and omit it.

As regards the formula update, in Figure 4 we present an algorithm that, given an $ATL^*$-formula $\varphi$ on $AP$ and a truth value $v$, returns a new formula Transl($\varphi, v$) on $AP'$, which handles the new atoms generated by Duplicate_atoms(). Intuitively, the procedures above are meant to reduce model checking the three-valued semantics for $ATL^*$ to model checking two-valued semantics. To this end, we prove the following result.

**Lemma 6.** Given an IS $M$ and $ATL^*$ formula $\varphi$, let $M' = \texttt{Duplicate\_atoms}(M)$, $\varphi_{\mathrm{tt}} = \texttt{Transl}(\varphi, \mathrm{tt})$, and $\varphi_{\mathrm{ff}} = \texttt{Transl}(\varphi, \mathrm{ff})$. Then, we have that:

$$\begin{aligned}
(M', s) \models^2 \varphi_{\mathrm{tt}} &\quad\Leftrightarrow\quad ((M, s) \models^3 \varphi) = \mathrm{tt} && (9)\\
(M', s) \models^2 \varphi_{\mathrm{ff}} &\quad\Leftrightarrow\quad ((M, s) \models^3 \varphi) = \mathrm{ff} && (10)\\
(M', s) \models^2 \neg(\varphi_{\mathrm{tt}} \vee \varphi_{\mathrm{ff}}) &\quad\Leftrightarrow\quad ((M, s) \models^3 \varphi) = \mathrm{uu} && (11)
\end{aligned}$$

*Proof.* The proofs of (9) and (10) are by mutual induction on the structure of formula $\varphi$. For the inductive steps, we do not present the case where the main operator is a temporal modality. For the latter operators, the cases are immediate.

(9) *Base case.* For $\varphi = q$ and $\varphi_{\mathrm{tt}} = q_{\mathrm{tt}}$, $(M', s) \models^2 \varphi_{\mathrm{tt}}$ iff $\Pi'(s, q_{\mathrm{tt}}) = \mathrm{tt}$ by Def. 6. By point (2) in the definition of $\texttt{Duplicate\_atoms}(M)$, $\Pi'(s, q_{\mathrm{tt}}) = \mathrm{tt}$ iff $\Pi(s, q) = \mathrm{tt}$. By Def. 9, this is equivalent to $((M, s) \models^3 q) = \mathrm{tt}$.

*Inductive cases.*

For $\varphi_{\mathrm{tt}} = \texttt{Transl}(\neg\varphi', \mathrm{tt}) = \texttt{Transl}(\varphi', \mathrm{ff}) = \varphi'_{\mathrm{ff}}$, we have that $(M', s) \models^2 \varphi_{\mathrm{tt}}$ iff $(M', s) \models^2 \varphi'_{\mathrm{ff}}$, iff by induction hypothesis $((M, s) \models^3 \varphi') = \mathrm{ff}$. By Def. 9, this is the case iff $((M, s) \models^3 \varphi) = \mathrm{tt}$.

For $\varphi_{\mathrm{tt}} = \texttt{Transl}(\varphi' \wedge \varphi'', \mathrm{tt}) = \texttt{Transl}(\varphi', \mathrm{tt}) \wedge \texttt{Transl}(\varphi'', \mathrm{tt})$, we have that $(M', s) \models^2 \varphi_{\mathrm{tt}}$ iff $(M', s) \models^2 \varphi'_{\mathrm{tt}}$ and $(M', s) \models^2 \varphi''_{\mathrm{tt}}$, iff by induction hypothesis $((M, s) \models^3 \varphi') = \mathrm{tt}$ and $((M, s) \models^3 \varphi'') = \mathrm{tt}$. By Def. 9, this is the case iff $((M, s) \models^3 \varphi) = \mathrm{tt}$.

For $\varphi_{\mathrm{tt}} = \texttt{Transl}(\langle\!\langle\Gamma\rangle\!\rangle\psi, \mathrm{tt}) = \langle\!\langle\Gamma\rangle\!\rangle\texttt{Transl}(\psi, \mathrm{tt})$, by Def. 6, $(M', s) \models^2 \varphi_{\mathrm{tt}}$ iff there exists a joint strategy $F^n_\Gamma$, such that for all paths $p \in out(s, F^n_\Gamma)$, $(M', p) \models^2 \psi_{\mathrm{tt}}$. By induction hypothesis we have that $((M, p) \models^3 \psi) = \mathrm{tt}$. By Def. 9 this is the case iff $((M', s) \models^3 \langle\!\langle\Gamma\rangle\!\rangle\psi) = \mathrm{tt}$.

(10) *Base case.* For $\varphi = q$ and $\varphi_{\mathrm{ff}} = q_{\mathrm{ff}}$, $(M', s) \models^2 \varphi_{\mathrm{ff}}$ iff $\Pi'(s, q_{\mathrm{ff}}) = \mathrm{tt}$ by Def. 6. By point (2) in the definition of $\texttt{Duplicate\_atoms}(M)$, $\Pi'(s, q_{\mathrm{ff}}) = \mathrm{tt}$ iff $\Pi(s, q) = \mathrm{ff}$. By Def. 9, this is equivalent to $((M, s) \models^3 q) = \mathrm{ff}$.

*Inductive cases.*

For $\varphi_{\mathrm{ff}} = \texttt{Transl}(\neg\varphi', \mathrm{ff}) = \texttt{Transl}(\varphi', \mathrm{tt}) = \varphi'_{\mathrm{tt}}$, we have that $(M', s) \models^2 \varphi_{\mathrm{ff}}$ iff $(M', s) \models^2 \varphi'_{\mathrm{tt}}$, iff by induction hypothesis $((M, s) \models^3 \varphi') = \mathrm{tt}$. By Def. 9, this is the case iff $((M, s) \models^3 \varphi) = \mathrm{ff}$.

For $\varphi_{\mathrm{ff}} = \texttt{Transl}(\varphi' \wedge \varphi'', \mathrm{ff}) = \texttt{Transl}(\varphi', \mathrm{ff}) \vee \texttt{Transl}(\varphi'', \mathrm{ff})$, we have that $(M', s) \models^2 \varphi_{\mathrm{ff}}$ iff $(M', s) \models^2 \varphi'_{\mathrm{ff}}$ or $(M', s) \models^2 \varphi''_{\mathrm{ff}}$, iff by induction hypothesis $((M, s) \models^3 \varphi') = \mathrm{ff}$ or $((M, s) \models^3 \varphi'') = \mathrm{ff}$. By Def. 9, this is the case iff $((M, s) \models^3 \varphi) = \mathrm{ff}$.

For $\varphi_{\mathrm{ff}} = \texttt{Transl}(\langle\!\langle\Gamma\rangle\!\rangle\psi, \mathrm{ff}) = \langle\!\langle\bar{\Gamma}\rangle\!\rangle\texttt{Transl}(\psi, \mathrm{ff})$, by Def. 6, $(M', s) \models^2 \varphi_{\mathrm{ff}}$ iff there exists a joint strategy $F^n_{\bar{\Gamma}}$, such that for all paths $p \in out(s, F^n_{\bar{\Gamma}})$, $(M', p) \models^2 \psi_{\mathrm{ff}}$. By induction hypothesis we have that $((M, p) \models^3 \psi) = \mathrm{ff}$. By Def. 9 this is the case iff $((M', s) \models^3 \langle\!\langle\Gamma\rangle\!\rangle\psi) = \mathrm{ff}$.

(11) We have that $(M', s) \models^2 \neg(\varphi_{\mathrm{tt}} \vee \varphi_{\mathrm{ff}})$ iff $(M', s) \not\models^2 \varphi_{\mathrm{tt}}$ and $(M', s) \not\models^2 \varphi_{\mathrm{ff}}$. By (9) and (10) this is the case iff $(M, s) \models^3 \varphi) \neq \mathrm{tt}$ and $(M, s) \models^3 \varphi) \neq \mathrm{ff}$, that is, $(M, s) \models^3 \varphi) = \mathrm{uu}$. □

Given Lemma 6, we can present the algorithm to decide the model checking problem for the three-valued semantics with bounded recall and analyse its complexity.

```
Algorithm MC3($M, \varphi, n$):
1 $M' =$ Inflate($M, n$);
2 $M'' =$ Duplicate_atoms($M'$);
3 $\varphi_{\text{tt}} =$ Transl($\varphi, \text{tt}$);
4 $\varphi_{\text{ff}} =$ Transl($\varphi, \text{ff}$);
5 if MC_ATL$^*_{\texttt{ir}}$($M'', \varphi_{\text{tt}}$)
6   then return tt;
7 else if MC_ATL$^*_{\texttt{ir}}$($M'', \varphi_{\text{ff}}$)
8   then return ff;
9 else return uu;
```

Figure 5: Algorithm to decide *ATL*$^*$ three-valued model checking.

**Theorem 5.** For $n \in \mathbb{N}^+$, the model checking problem for *ATL*$^*$ on the three-valued semantics with $n$-bounded recall and imperfect information is in *EXPTIME.* Moreover, the corresponding $n$-fixed-recall MC problem is *PSPACE*-complete.

**Proof.** As regards the general case for $n \in \mathbb{N}^+$, we extend the model checking procedure outlined in the proof of Theorem 2 by the procedure in Figure 5. As in the proof of Theorem 2, we "inflate" the original IS $M$ to create a new IS $M'$ whose states are histories of length at most $n$, with an exponential blow-up. In line 2, we add in $M'$ atoms $q_{\text{tt}}$ and $q_{\text{ff}}$, for each atom $q \in AP$, and update the labeling function as explained above: if $\Pi(s,q) = \text{tt}$ then $\Pi(s,q_{\text{tt}}) = \text{tt}$, otherwise $\Pi(s,q_{\text{ff}}) = \text{tt}$. The latter can be done in polynomial time in the size of $M$. Then, in lines 3-4, we call twice the translation procedure in Figure 4 to generate formulas $\varphi_{\text{tt}} = \texttt{Transl}(\varphi, \text{tt})$ and $\varphi_{\text{ff}} = \texttt{Transl}(\varphi, \text{ff})$ that will be used in the model checking procedure in two-valued semantics for imperfect information and imperfect recall. Note that, each translation can be done in polynomial time in the size of formula $\varphi$. In lines 5-7 we determine the truth value of both formulas. In particular, if the model checking procedure in two-valued semantics returns true when considering $\varphi_{\text{tt}}$ (line 5), our algorithm returns true since by Lemma 6.(9) formula $\varphi$ holds in $M$ under the three-valued semantics. Otherwise, in line 6 our algorithm checks whether the model checking procedure for $\varphi_{\text{ff}}$ returns true and then it returns false by Lemma 6.(10). Consequently, if both model checking calls return false, the algorithm returns undefined (line 7). Since, checking *ATL*$^*$ formulas on the IS $M''$ can be done in polynomial space Schobbens (2004), then the whole procedure is in *EXPTIME.*

For a fixed $n \in \mathbb{N}^+$, the procedure above is in *PSPACE.* As regards the lower bound, we make use of the same reduction as in Theorem 2. In particular, we can reduce model checking an *LTL* formula $\psi$ to the verification of the truth of the *ATL*$^*$ formula $\langle\!\langle \emptyset \rangle\!\rangle \psi$ in the three-valued semantics. □

By Theorems 2 and 5 model checking *ATL*$^*$ on the two- and three-valued semantics has the same complexity. This is also the case for *ATL*.

**Theorem 6.** For $n \in \mathbb{N}^+$, the model checking problem for *ATL* in the three-valued semantics with $n$-bounded recall and imperfect information is in *EXPTIME.* Moreover, the corresponding $n$-fixed-recall MC problem is $\Delta_2^P$-complete.

**Proof.** Clearly, the *EXPTIME* upper bound for the general case still holds.

As for a fixed $n \in \mathbb{N}^+$, we adapt the proof of Theorem 5. In particular, we modify lines 5-6 in Algorithm 5 by calling procedure $\mathtt{MC\_ATL_{ir}}$ instead, which is known to be in $\Delta_2^P$ (Jamroga & Dix, 2006). □

Again, for $n \in \mathbb{N}^+$, the complexity of the $n$-fixed-recall model checking problem for three-valued $ATL$ and $ATL^*$ with $n$-bounded recall (and imperfect information) is the same as for imperfect recall. Also, as in Section 2, in providing these results, we are primarily interested in the decidability of the model checking problem for bounded recall, irrespectively of tight complexity bounds for the general case.

We conclude by observing that translation `Transl`() is of interest in its own, as it allows to reduce three-valued model checking to the corresponding two-valued problem. We envisage to introduce similar translations for other multi-valued logics, by using the same procedure of adding new atomic propositions (one for each truth value), and then definining translations mirroring the truth conditions for each value. However, such general reduction of multi-valued model checking to the two-valued instance is beyond the scope of the present contribution. We leave it for future work.

## 4. Approximating Perfect Recall

In this section we lay the theoretical foundations of a partial model checking procedure to verify $ATL^*$ under the assumptions of imperfect information and perfect recall. In Section 4.1 we present a result on the preservation of defined truth values in $ATL^*$ when increasing recall. Then, in Section 4.2 we present the model checking procedure to approximate perfect recall.

### 4.1 Preservation of Three-Valued $ATL^*$

The main result of this section, which is akin to Lemma 2, details the preservation of $ATL^*$ formulas when increasing the amount of recall. However, differently from Lemma 2, Lemma 7 hereafter holds for all $ATL^*$ formulas.

**Lemma 7.** Let $m, n \in \mathbb{N}^+ \cup \{\omega\}$ be such that $m \leq n$; let $\psi$ be a formula in $ATL^*$. Then,

$$((M,s) \models_m^3 \psi) = \mathrm{tt} \quad \Rightarrow \quad ((M,s) \models_n^3 \psi) = \mathrm{tt} \tag{12}$$

$$((M,s) \models_m^3 \psi) = \mathrm{ff} \quad \Rightarrow \quad ((M,s) \models_n^3 \psi) = \mathrm{ff} \tag{13}$$

**Proof.** The proofs for both (12) and (13) are by simultaneous induction on the structure of formula $\psi$. We only present the case where the main operator is the strategic modality, the other cases being immediate.

(12) By Def. 9 $((M,s) \models_m^3 \langle\langle\Gamma\rangle\rangle\psi) = \mathrm{tt}$ if for some joint strategy $F_\Gamma^m$, for all paths $p \in out(s, F_\Gamma^m)$, $((M,p) \models_m^3 \psi) = \mathrm{tt}$. Given $F_\Gamma^m$ we can construct a joint strategy $F_\Gamma^n$ as for Lemma 2: for all $i \in \Gamma$ and all histories $h \in \mathcal{G}^{<1+n}$, we define $f_i^n(h) = f_i^m(h_{(|h|-m)}, \ldots, h_{|h|})$ for $m < |h|$, $f_i^n(h) = f_i^m(h)$ otherwise. Notice that each $f_i^n$ so defined is uniform, provided that $f_i^m$ is. Given $F_\Gamma^n$ thus defined, we obtain that $out(s, F_\Gamma^n) = out(s, F_\Gamma^m)$. In particular, for all paths $p \in out(s, F_\Gamma^n)$, $((M,s) \models_n^3 \psi) = \mathrm{tt}$ by induction hypothesis, and therefore $((M,s) \models_n^3 \langle\langle\Gamma\rangle\rangle\psi) = \mathrm{tt}$.

(13) By Def. 9 $((M,s) \models_m^3 \langle\langle\Gamma\rangle\rangle\psi) = \mathrm{ff}$ if for some joint strategy $F_\Gamma^m$, for all paths $p \in out(s, F_\Gamma^m)$, $((M,p) \models_m^3 \psi) = \mathrm{ff}$. Given $F_\Gamma^m$ we can construct a joint strategy $F_\Gamma^n$

```
Algorithm Iterative_MC($M, \psi, n$):
1 $j = 0$, $k = \mathrm{uu}$;
2 while $j < n$ and $k = \mathrm{uu}$
3         $j = j + 1$;
4         $k = \mathtt{MC3}(M, \psi, j)$;
5 end while;
6 if $k \neq \mathrm{uu}$ then return $(j, k)$;
7 else return $-1$;
```

Figure 6: The procedure `Iterative_MC` to decide $ATL^*$ iteratively.

as in point (12). Again, each $f_i^n$ so defined is uniform, provided that $f_i^m$ is. Given such $F_\Gamma^n$, we obtain that $out(s, F_\Gamma^n) = out(s, F_\Gamma^m)$. In particular, for all paths $p \in out(s, F_\Gamma^n)$, $((M,p) \models_n^3 \psi) = \mathrm{ff}$ by induction hypothesis, and therefore $((M,s) \models_n^3 \langle\!\langle\Gamma\rangle\!\rangle\psi) = \mathrm{ff}$. □

By Lemma 7 adding memory preserves defined truth values for all formulas in $ATL^*$. This is in contrast with Lemma 2. Indeed, even though in some cases the value of an $ATL^*$ formula may be undefined in the three-valued semantics, whenever it is defined, it does not change when memory is added.

By combining together Lemmas 4 and 7 we obtain our main result on the relationship between bounded recall and the two- and three-valued semantics.

**Corollary 2.** Let $m, n \in \mathbb{N}^+ \cup \{\omega\}$ be such that $m \leq n$; let $\psi$ be a formula in $ATL^*$. Then,

$$((M,p) \models_m^3 \psi) = \mathrm{tt} \quad \Rightarrow \quad ((M,p) \models_n^2 \psi) \tag{14}$$

$$((M,p) \models_m^3 \psi) = \mathrm{ff} \quad \Rightarrow \quad ((M,p) \not\models_n^2 \psi) \tag{15}$$

Of particular interest is the case for $m \in \mathbb{N}^+$ and $n = \omega$. By Corollary 2 we can outline a verification procedure for perfect recall, whereby $ATL^*$ formulas are checked in the three-valued semantics iteratively. If either the value true or false is returned, then by Corollary 2 this is also the truth value for the two-valued semantics under perfect recall. We explore this intuition in the verification procedure defined below.

### 4.2 A Partial Decision Procedure for $ATL^*_{iR}$

We now present a partial decision procedure for model checking $ATL^*$ under the assumptions of imperfect information and $n$-bounded recall. It is partial, as it is not guaranteed to terminate for the case of perfect recall, that is, for $n = \omega$. This procedure is described in algorithm `Iterative_MC`$(M, \psi, n)$ in Figure 6. It takes as input an IS $M$, an $ATL^*$ formula $\psi$, and a bound $n \in \mathbb{N}^+ \cup \{\omega\}$. It includes a `while`-loop (lines 2-6), whose guard checks whether the bound has not yet been attained ($j < n$) and $\psi$ has not yet been decided ($k = \mathrm{uu}$). Within the loop, formula $\psi$ is model-checked in $M$ according to the three-valued semantics by subroutine `MC3`(), and variable $k$ stores the result. On exiting the loop, variable $k$ is tested (line 6). If $k \neq \mathrm{uu}$, the loop was exited because of a defined answer for the three-valued model checking problem with $j$-bounded recall (and possibly bound $n$ was reached). By Corollary 2 we can then transfer the value returned to the corresponding model checking problem for the two-valued semantics. On the other hand, if $k = \mathrm{uu}$ then the bound has been attained in the loop and the default value $-1$ is returned to signal exit without a defined truth value. We now prove the termination of the algorithm in Figure 6 for $n \in \mathbb{N}^+$, as well as its soundness.

**Theorem 7.** For $n \in \mathbb{N}^+$, `Iterative_MC`() terminates in *EXPTIME*. Moreover, `Iterative_MC`() is sound: if the value returned is different from $-1$, then $M \models^2_n \phi$ iff $k = \text{tt}$ and $M \not\models^2_n \phi$ iff $k = \text{ff}$.

**Proof.** As regards termination in *EXPTIME*, notice that for $n \in \mathbb{N}^+$ the algorithm in Figure 6 calls procedure $MC3()$, which is in *EXPTIME* (Theorem 5), a bounded number of times. Then, the overall complexity is also in *EXPTIME*.

As for soundness, suppose that the value returned is different from $-1$. In particular, this means that either $k = \text{tt}$ or $k = \text{ff}$. If $k = \text{tt}$ then by the structure of `Iterative_MC`(), $((M, s) \models^3_j \psi) = \text{tt}$ for some $j \leq n$. By Corollary 2.(14) we obtain $M \models^2_n \phi$. On the other hand, suppose that $M \models^2_n \phi$ and assume $k = \text{ff}$ to derive a contradiction. Then, by the structure of `Iterative_MC`(), $((M, s) \models^3_j \psi) = \text{ff}$ for some $j \leq n$, and by Corollary 2.(15) we have $M \not\models^2_n \phi$, a contradiction. Hence, $k = \text{tt}$ as required. The cases for $M \not\models^2_n \phi$ iff $k = \text{ff}$ is similar. □

Incidentally, we observe that, for a fixed $n \in \mathbb{N}^+$, algorithm `Iterative_MC`() actually runs in *PSPACE*.

An important application of `Iterative_MC`() is for the case $n = \omega$, namely model checking perfect recall. In such a case, termination is no longer guaranteed, but soundness still is.

**Theorem 8.** For $n = \omega$, `Iterative_MC`() does not necessarily terminate. However, `Iterative_MC`() is sound: if the value returned is different from $-1$, then $M \models^2_n \phi$ iff $k = \text{tt}$ and $M \not\models^2_n \phi$ iff $k = \text{ff}$.

**Proof.** We have remarked that in several games, for example, the matching pennies game by Bulling et al. (2008), neither player has a strategy to win the game, no matter how much recall we assume on our players. So, algorithm `Iterative_MC`() will never return a defined truth value for any $j \in \mathbb{N}^+$, and therefore it will never exit the `while` loop.

Soundness follows again by Corollary 2. □

As a result, by Theorem 8 we have a sound, albeit incomplete, decision procedure for model checking $ATL^*$ with perfect recall and imperfect information. Observe that no complete procedure can be obtained as the problem is undecidable in general (Dima & Tiplea, 2011).

**Example 3.** In relation with the IS $M$ for the voting scenario in Example 2, consider again the specification $\varphi_3 = \bigwedge_{i=1}^{\ell} [\![Voter_i]\!] X \langle\!\langle Coercer \rangle\!\rangle F \phi_i$, where $\phi_i = ((vote_{i1} \wedge n\_punish_i) \vee (\bigvee_{j=2}^{k} vote_{ij} \wedge punish_i) \vee (n\_give_i \wedge punish_i))$, which intuitively states that no matter what voter $i$ does, at the next step the coercer has a strategy such that eventually either voter $i$ votes for candidate 1 or the coercer punishes her. This specification is neither existential nor universal, and therefore does not fall within the hypothesis of Lemma 2. Nevertheless, $\varphi_3$ is amenable to algorithm `Iterative_MC`() in Figure 6. Specifically, given the IS $M$ in Example 2, formula $\varphi_3$, and bound $n > t$, where $t$ is the number of waiting steps, the algorithm `Iterative_MC`$(M, \varphi_3, n)$ initializes the bound on recall to 0 and the value of $k$ to undefined uu. Then, in the `while` loop the subroutine `MC3`$(M, \varphi_3, 1)$ returns uu because, according to the three-valued semantics, the coercer does not have a memoryless strategy to enforce $F\phi_i$ at the next step, nor voter $i$ has a (memoryless) strategy to prevent $F\phi_i$ at the next step. On the other hand, in the $t+1$ iteration of the function call, `MC3`$(M, \varphi_3, t+1)$ returns true, as the coercer has a $t+1$-bounded recall strategy to enforce $F\phi_i$ at the next step, and therefore $\varphi_3$ holds. Thus, we conclude that the IS $M$ in Example 2 satisfies specification $\varphi_3$ under the assumptions of imperfect information and perfect recall.

## 5. Experimental Results

In this section we present the MCMAS$_{BR}$ model checker to verify ATL-specifications according to the bounded recall semantics, which can also be used to approximate perfect recall. Then, we evaluate it empirically on the two examples introduced in Section 2.

### 5.1 The MCMAS$_{BR}$ Model Checker

We implemented the algorithms in Section 4 in MCMAS$_{BR}$ MCMAS$_{BR}$ (2021), an experimental model checker that extends the open-source verification tool MCMAS (Lomuscio et al., 2017) by supporting the bounded recall semantics introduced in Section 2, while maintaining full functionality for memoryless semantics. In summary, agents in MCMAS$_{BR}$ recall a bounded number of the latest states visited in the run, which is given as input by the user. Protocol functions are defined as for the memoryless semantics. Given the notion of recall here adopted, the agents' strategies are based on bounded local histories, rather than on their present state only, as it is the case under the memoryless semantics. MCMAS$_{BR}$ takes as input an ISPL file describing the multi-agent system under analysis and a set of formulas to be verified. The syntax of the ISPL file is the same as for standard MCMAS. The present version of the checker only supports *ATL* specifications, which is the case for MCMAS too.

Verification under three-valued bounded recall semantics is carried out by invoking the tool with the command:

```
python mcmas_br.py [k] [file.ispl]
```

where `k` is the bound on recall specified by user and `file.ispl` is the ISPL file containing the model and the specification. In the present version of MCMAS$_{BR}$ all agents have the same bound on recall; extending this feature would not be problematic, but it is beyond the scope of the current contribution.

Upon invocation, the tool parses the input ISPL file, and for each specification $\varphi$ appearing in the ISPL file, it generates two translated formulas $\varphi_{\mathrm{tt}}$ and $\varphi_{\mathrm{ff}}$, according to function `Transl`() described in Figure 4. The tool then makes model checking calls iteratively until it reaches the maximum recall bound `k`; after each check, the tool displays the verification result. At each iteration, the tool constructs the model, where the agents' recall has its dimension fixed by the bound, and new atomic propositions are also duplicated and added to the model according to algorithm `MC3`() in Figure 5. For each variable in the local state, an array of BDD variables of the length at most `k` is generated for encoding the local histories. Since we have a fixed memory window of `k`, at the initialisation stage where the history is less than k, an additional unused state is used as a place holder. The model construction phase generates the set of bounded histories which are of arbitrary lengths up to the bound on recall. For certain formulas with undefined values for smaller bounds, the tools allows early termination once a true or false value is obtained.

MCMAS$_{BR}$ implements several methods to minimise the memory and computational overheads generated by the bounded recall semantics. We adopt an efficient usage of BDD variables, which allows us to manipulate individual observations of each local history. For example, at each time step in a run, the symbolic encoding of each local history contains the composition of the previous history with the new variable assignments representing portions of the local history. At each execution step, only the oldest observation is discarded and the rest of the observations in the state only shifts by one position in the BDD variables of the next state. The new variable assignment is applied to the set of BDD variables encoding the latest observation. This optimises the BDD memory used for computing and storing large local histories, notably during the subset construction stage where Boolean variables are generated to encode the bounded history space. Agents' protocols are also optimised to account for the bounded recall semantics, and are used to generate history-based strategies. The model checking algorithm is adapted from Busard et al. (2015) which contains practical optimisations such as early termination and caching for speeding up the model checking process under uniformity conditions.

### 5.2 Evaluation

Intuitively, the increased expressivity of the bounded recall semantics comes at the computational cost of a larger number of Boolean variables required to encode histories when compared against the standard memoryless semantics. This is expected to cause a performance degradation in the verification step. Note that, however, the problem remains decidable, differently from the case of unbounded recall, which is undecidable in general. To evaluate experimentally the cost of bounded recall, we now report the experiments conducted on the scalability of MCMAS$_{BR}$ as we increase the value of the bound and the example size, starting with the Shell Game described in Example 1 in ISPL.

Here, we generalise the $ATL$ specification "the Guesser has a winning strategy to guess the correct location of the ball" shown in the proof of Lemma 3, to an arbitrary number $N$ of shells:

$$\varphi_1 = \langle\langle Guesser \rangle\rangle F \varphi_{g\_win}$$

where $\varphi_{g\_win} = \bigvee_{i=1}^{N}(guess_i \wedge shell_i)$.

As a further specification, we check whether "the Shuffler has a strategy to enforce that the Guesser will not guess the correct location of the ball and thus cannot win":

$$\varphi_2 = \langle\langle Shuffler \rangle\rangle G \neg \varphi_{g\_win}$$

Intuitively, the truth of $\varphi_1$ depends on whether the bound on recall is large enough for the agent to distinguish the states that contain information of the shell location, that is, when the bound is greater than the number of waiting steps. On the other hand, $\varphi_2$ should be false whenever the bound exceeds the number of waiting steps and undefined for smaller bounds.

| (shells, waiting) | bound | reachable histories | possible histories | BDD memory | $\varphi_1$ |  | $\varphi_2$ |  |
|---|---|---|---|---|---|---|---|---|
|  |  |  |  |  | time | value | time | value |
| (20,20) | 5 | 2421 | $4.7 \times 10^{15}$ | 17912816 | 1.473 | uu | 1.326 | uu |
|  | 9 | 4021 | $1.6 \times 10^{28}$ | 46451184 | 15.117 | uu | 12.163 | uu |
|  | 13 | 5621 | $5.7 \times 10^{40}$ | 63613616 | 66.635 | uu | 63.478 | uu |
|  | 17 | 7221 | $2.0 \times 10^{53}$ | 65742672 | 97.402 | uu | 94.056 | uu |
|  | 21 | 8821 | $6.8 \times 10^{65}$ | 63289776 | 72.788 | tt | 63.855 | ff |
|  | 25 | 10421 | $2.3 \times 10^{78}$ | 71859312 | 180.324 | tt | 170.324 | ff |
| (20,24) | 5 | 2501 | $6.5 \times 10^{15}$ | 29185392 | 4.018 | uu | 3.125 | uu |
|  | 9 | 4101 | $2.9 \times 10^{28}$ | 38615600 | 9.115 | uu | 9.463 | uu |
|  | 13 | 5701 | $1.3 \times 10^{41}$ | 45592752 | 62.780 | uu | 60.856 | uu |
|  | 17 | 7301 | $5.7 \times 10^{53}$ | 50125520 | 179.450 | uu | 132.888 | uu |
|  | 21 | 8901 | $2.5 \times 10^{66}$ | 62216144 | 276.825 | uu | 296.082 | uu |
|  | 25 | 10501 | $1.1 \times 10^{79}$ | 67699408 | 212.931 | tt | 173.577 | ff |
| (30,20) | 5 | 5131 | $1.2 \times 10^{17}$ | 30722752 | 3.455 | uu | 4.270 | uu |
|  | 9 | 8731 | $5.9 \times 10^{30}$ | 49790544 | 70.410 | uu | 88.751 | uu |
|  | 13 | 15931 | $2.8 \times 10^{44}$ | 68733232 | 214.854 | uu | 256.994 | uu |
|  | 17 | 15571 | $1.3 \times 10^{58}$ | 75444064 | 256.938 | uu | 322.090 | uu |
|  | 21 | 23131 | $6.3 \times 10^{71}$ | 65889200 | 94.542 | tt | 108.452 | ff |
|  | 25 | 22771 | $3.0 \times 10^{85}$ | 83397840 | 278.351 | tt | 355.215 | ff |
| (30,24) | 5 | 5251 | $1.6 \times 10^{17}$ | 31004400 | 5.978 | uu | 23.686 | uu |
|  | 9 | 8851 | $9.1 \times 10^{30}$ | 51694640 | 30.242 | uu | 35.217 | uu |
|  | 13 | 12451 | $5.2 \times 10^{44}$ | 60624528 | 218.773 | uu | 244.457 | uu |
|  | 17 | 16051 | $3.0 \times 10^{58}$ | 62232976 | 527.586 | uu | 638.194 | uu |
|  | 21 | - | $1.7 \times 10^{72}$ | - | * | - | * | - |
|  | 25 | - | $9.7 \times 10^{85}$ | - | * | - | * | - |

Table 1: Experimental results for the *Shell Game*.

Table 1 shows the experimental results obtained with MCMAS$_{BR}$ running on an Intel Core$^{\text{TM}}$ i7-2600 CPU 3.40GHz machine with 16GB RAM running Ubuntu v18.04.2 (Linux kernel v4.15). The table displays the following information in each column:

1. the number of shells and of waiting steps;
2. the bound on recall given as a user parameter;
3. the number of reachable histories of the particular game instance (Note our implementation treats histories as states, and therefore in practice on MCMAS$_{BR}$ this is the number of reachable states);
4. the total number of (possible) histories in the instance's state space;
5. the amount of BDD memory usage for the instance (in Mb);
6. for each formula $\varphi_1$ and $\varphi_2$, its verification time, inclusive of both the model construction step and the model checking algorithm running time (in seconds), as well as the verification result obtained.

As reported in Table 1, we ran experiments with a varying bound on recall between 5 and 25, and 20 to 30 shells. As expected, $\varphi_1$ was evaluated to undefined when the bound was smaller than the number of waiting steps, as Guesser was not able to remember the location of the ball and thus did not have a winning strategy in such scenarios. The experiments confirmed the correctness of the implementation. The verification performance degrades as the bound increases, leading to an increase in the associated state space. Note that

| (voters, waiting) | bound | reachable histories | possible histories | BDD memory | $\varphi_3$ |  | $\varphi_4$ |  |
|---|---|---|---|---|---|---|---|---|
|  |  |  |  |  | time | value | time | value |
| (2, 6) | 3 | 253 | $5.5 \times 10^{9}$ | 10527136 | 0.421 | uu | 0.077 | uu |
|  | 5 | 285 | $1.7 \times 10^{16}$ | 12675360 | 0.550 | uu | 0.205 | uu |
|  | 7 | 317 | $5.3 \times 10^{22}$ | 30666656 | 18.697 | tt | 0.433 | ff |
|  | 9 | 365 | $1.7 \times 10^{29}$ | 36925408 | 109.035 | tt | 0.638 | ff |
|  | 11 | 285 | $5.2 \times 10^{35}$ | 39708256 | 202.593 | tt | 1.425 | ff |
| (2, 8) | 3 | 301 | $8.7 \times 10^{9}$ | 10895424 | 0.427 | uu | 0.089 | uu |
|  | 5 | 333 | $3.7 \times 10^{16}$ | 13422240 | 1.052 | uu | 0.230 | uu |
|  | 7 | 365 | $1.6 \times 10^{23}$ | 32387328 | 10.084 | uu | 0.475 | uu |
|  | 9 | 397 | $6.6 \times 10^{29}$ | 50866528 | 463.351 | tt | 0.921 | ff |
|  | 11 | 445 | $2.8 \times 10^{36}$ | 35314128 | 3130.008 | tt | 1.720 | ff |
| (3, 6) | 3 | 1209 | $4.5 \times 10^{12}$ | 13346960 | 1.741 | uu | 0.260 | uu |
|  | 5 | 1401 | $1.2 \times 10^{21}$ | 27770800 | 622.87 | uu | 0.633 | uu |
|  | 7 | 7025 | $3.3 \times 10^{29}$ | 12781792 | $*$ | - | 4.179 | ff |
|  | 9 | 7616 | $8.9 \times 10^{37}$ | 18676640 | $*$ | - | 72.874 | ff |
|  | 11 | 8256 | $2.4 \times 10^{46}$ | 18148672 | $*$ | - | 121.720 | ff |
| (3, 8) | 3 | 1433 | $7.1 \times 10^{12}$ | 15651024 | 1.860 | uu | 0.273 | uu |
|  | 5 | 1625 | $2.6 \times 10^{21}$ | 33152144 | 796.670 | uu | 0.734 | uu |
|  | 7 | 1817 | $9.6 \times 10^{29}$ | 27577424 | $*$ | - | 5.652 | uu |
|  | 9 | 2009 | $3.6 \times 10^{38}$ | 28857328 | $*$ | - | 75.669 | ff |
|  | 11 | 2297 | $1.3 \times 10^{47}$ | 35314128 | $*$ | - | 192.040 | ff |

Table 2: Experimental results for the *Simple Voting Scenario*.

*undefined* formulas can sometimes require a longer computation time, with an exhaustive search in the state space. However, when the bound reaches a value where the formula becomes defined, the computation time is shorter, which is likely due to early termination when a winning strategy is found. In Table 1 we used a time-out of 60 minutes, which is represented as a star $*$.

To further evaluate the scalability of MCMAS$_{BR}$, we implemented and verified the Simple Voting Scenario in Example 2, where we consider multiple agents. We evaluate the voting protocol against specification $\varphi_3$ in Example 3. It expresses that no matter what voter $i$ does, at the next step, Coercer has a strategy whereby they can enforce each voter to vote for Candidate 1, otherwise the voter will be punished:

$$\varphi_3 = \bigwedge_{i=1}^{n_v} [\![Voter_i]\!] X \langle\!\langle Coercer \rangle\!\rangle F((vote_{i1} \wedge n\_punish_i) \vee (\bigvee_{j=2}^{k} vote_{ij} \wedge punish_i) \vee \\ (n\_give_i \wedge punish_i))$$

Further, we evaluate the $ATL$ specification $\varphi_4$ stating that the voters collectively have a strategy to avoid being punished:

$$\varphi_4 = \langle\!\langle all\_voters \rangle\!\rangle G \neg (\bigvee_{i=1}^{n_v} (n\_give_i \wedge n\_punish_i) \vee (give_i \wedge punish_i))$$

Table 2 reports the verification results obtained by evaluating the Simple Voting Scenario with different numbers of voters and waiting steps. Formula $\varphi_3$ is evaluated as undefined for bounds smaller than 9, corresponding to the fact that Coercer can no longer recall the voting proof he received from the Voter. By increasing the bound on recall, the formula is then evaluated to true. Formula $\varphi_4$ is also evaluated to undefined for bounds smaller than 9, but then is evaluated to false when the bound is increased to 9 or above, corresponding again to a situation where the Coercer has sufficient memory to recall the proof that has been received earlier. As for the Shell Game, the performance of the model checker degrades as we increase the bound on recall, hence the model size. The memory footprint of the tool increases at a slower rate compared with the increase in verification time, indicating an efficient usage of BDD variables.

## 6. Conclusions

Model checking multi-agent systems against alternating-time temporal logic is known to be undecidable under the assumptions of perfect recall and imperfect information. In this paper we put forward a sound, albeit incomplete, verification procedure for perfect recall based on a notion of bounded recall. To do so, we introduced bounded recall on interpreted systems by providing both a two- and a three-valued semantics. By using the three-valued semantics for bounded recall we were able to prove Lemma 4 on the preservation of defined truth values from the bounded to the perfect recall case for all $ATL^*$ specifications. As shown in Lemma 2, in the classic two-valued semantics, preservation holds only for the rather restricted universal and existential fragments of $ATL^*$. These results lay the foundation for the iterative procedure illustrated in Section 4, which can, in some cases, solve the model checking problem under perfect recall by considering a bounded amount of memory for the agents in the system. Since model checking perfect recall under incomplete information is undecidable in general, the procedure discussed is necessarily incomplete. Yet, to the best of our knowledge, this constitutes the first procedure available which can provide solutions in cases of practical interest. We illustrated our method by extending MCMAS to support bounded recall functionalities. The resulting tool, MCMAS$_{BR}$, which employs symbolic structures to encode recall histories, was evaluated experimentally. The analysis showed that, for some protocols of interest, recall bounds of approximately 20 steps, corresponding to over $10^{85}$ possible histories can be practically checked. The time effort required for the resulting checks appears to be exponential, in line with the *EXPTIME* bound provided at theoretical level. Finally, with translation function `Transl`() in Section 3 we provided a method to reduce three-valued model checking to the corresponding two-valued problem. We believe that such a reduction is of independent interest and might find applications beyond the scope of the present contribution.

In further work we would like to explore combinations of bounded recall with other notions of interest in specifications for multi-agent systems, including Strategy Logic and epistemic logic. In a further line we would like to explore the combination between recall bounds and bounded resources.

**Acknowledgements.** F. Belardinelli acknowledges the support of ANR JCJC Project SVeDaS (ANR-16-CE40-0021). A. Lomuscio is supported by a Royal Academy of Engineering Chair in Emerging Technology. E. Yu acknowledges the support of the Austrian FWF grant W1255-N23 and the LIT AI Lab funded by the State of Upper Austria.

## References

T. Ågotnes & D. Walther. A logic of strategic ability under bounded memory. *Journal of Logic, Language and Information*, 18(1):55–77, 2009. doi: 10.1007/s10849-008-9075-4. URL `https://doi.org/10.1007/s10849-008-9075-4`.

T. Ågotnes, V. Goranko, W. Jamroga, & M. Wooldridge. Knowledge and ability. In *Handbook of Logics for Knowledge and Belief*, pages 543–589. Elsevier, 2015.

N. Alechina, B. Logan, H. N. Nguyen, F. Raimondi, & L. Mostarda. Symbolic model-checking for resource-bounded ATL. In G. Weiss, P. Yolum, R. H. Bordini, and E. Elkind, editors, *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015, Istanbul, Turkey, May 4-8, 2015*, pages 1809–1810. ACM, 2015. URL `http://dl.acm.org/citation.cfm?id=2773448`.

N. Alechina, N. Bulling, S. Demri, & B. Logan. On the complexity of resource-bounded logics. *Theoretical Computer Science*, 750:69–100, 2018. doi: 10.1016/j.tcs.2018.01.019. URL `https://doi.org/10.1016/j.tcs.2018.01.019`.

R. Alur, T. A. Henzinger, & O. Kupferman. Alternating-time temporal logic. *Journal of the ACM*, 49(5):672–713, 2002. doi: 10.1145/585265.585270. URL `https://doi.org/10.1145/585265.585270`.

C. Baier & J. Katoen. *Principles of model checking*. MIT Press, 2008. ISBN 978-0-262-02649-9.

T. Ball & O. Kupferman. An abstraction-refinement framework for multi-agent systems. In *21th IEEE Symposium on Logic in Computer Science (LICS 2006), 12-15 August 2006, Seattle, WA, USA, Proceedings*, pages 379–388. IEEE Computer Society, 2006. doi: 10.1109/LICS.2006.10. URL `https://doi.org/10.1109/LICS.2006.10`.

A. Bauer, M. Leucker, & C. Schallhart. Monitoring of real-time properties. In S. Arun-Kumar and N. Garg, editors, *FSTTCS 2006: Foundations of Software Technology and Theoretical Computer Science, 26th International Conference, Kolkata, India, December 13-15, 2006, Proceedings*, volume 4337 of *Lecture Notes in Computer Science*, pages 260–272. Springer, 2006. doi: 10.1007/11944836_25. URL `https://doi.org/10.1007/11944836\_25`.

A. Bauer, M. Leucker, & C. Schallhart. The good, the bad, and the ugly, but how ugly is ugly? In O. Sokolsky and S. Taşıran, editors, *Runtime Verification*, pages 126–138, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. ISBN 978-3-540-77395-5.

F. Belardinelli & A. Lomuscio. Agent-based abstractions for verifying alternating-time temporal logic with imperfect information. In K. Larson, M. Winikoff, S. Das, and E. H. Durfee, editors, *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2017, São Paulo, Brazil, May 8-12, 2017*, pages 1259–1267. ACM, 2017. URL `http://dl.acm.org/citation.cfm?id=3091300`.

F. Belardinelli & V. Malvone. A three-valued approach to strategic abilities under imperfect information. In D. Calvanese, E. Erdem, and M. Thielscher, editors, *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning, KR 2020, Rhodes, Greece, September 12-18, 2020*, pages 89–98, 2020. doi: 10.24963/kr.2020/10. URL `https://doi.org/10.24963/kr.2020/10`.

F. Belardinelli, A. Lomuscio, & J. Michaliszyn. Agent-based refinement for predicate abstraction of multi-agent systems. In G. A. Kaminka, M. Fox, P. Bouquet, E. Hüllermeier, V. Dignum, F. Dignum, and F. van Harmelen, editors, *ECAI 2016 - 22nd European*

*Conference on Artificial Intelligence, 29 August-2 September 2016, The Hague, The Netherlands - Including Prestigious Applications of Artificial Intelligence (PAIS 2016)*, volume 285 of *Frontiers in Artificial Intelligence and Applications*, pages 286–294. IOS Press, 2016. doi: 10.3233/978-1-61499-672-9-286. URL https://doi.org/10.3233/978-1-61499-672-9-286.

F. Belardinelli, A. Lomuscio, & V. Malvone. Approximating perfect recall when model checking strategic abilities. In M. Thielscher, F. Toni, and F. Wolter, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Sixteenth International Conference, KR 2018, Tempe, Arizona, 30 October - 2 November 2018*, pages 435–444. AAAI Press, 2018. URL https://aaai.org/ocs/index.php/KR/KR18/paper/view/18010.

F. Belardinelli, A. Lomuscio, & V. Malvone. An abstraction-based method for verifying strategic properties in multi-agent systems with imperfect information. In *Proceedings of AAAI*, 2019.

F. Belardinelli, A. Lomuscio, A. Murano, & S. Rubin. Verification of multi-agent systems with public actions against strategy logic. *Artificial Intelligence*, 285:103302, 2020a. doi: 10.1016/j.artint.2020.103302. URL https://doi.org/10.1016/j.artint.2020.103302.

F. Belardinelli, A. Lomuscio, & E. Yu. Model checking temporal epistemic logic under bounded recall. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7071–7078. AAAI Press, 2020b. URL https://aaai.org/ojs/index.php/AAAI/article/view/6193.

R. Berthon, B. Maubert, A. Murano, S. Rubin, & M. Y. Vardi. Strategy logic with imperfect information. *ACM Transactions in Computational Logic*, 22(1):5:1–5:51, 2021. doi: 10.1145/3427955. URL https://doi.org/10.1145/3427955.

T. Brihaye, A. D. C. Lopes, F. Laroussinie, & N. Markey. ATL with strategy contexts and bounded memory. In S. N. Artëmov and A. Nerode, editors, *Logical Foundations of Computer Science, International Symposium, LFCS 2009, Deerfield Beach, FL, USA, January 3-6, 2009. Proceedings*, volume 5407 of *Lecture Notes in Computer Science*, pages 92–106. Springer, 2009. doi: 10.1007/978-3-540-92687-0_7. URL https://doi.org/10.1007/978-3-540-92687-0_7.

G. Bruns & P. Godefroid. Model checking partial state spaces. In *Proceedings of the 11th International Conference on Computer Aided Verification (CAV99)*, volume 1633 of *LNCS*, pages 274–287. Springer-Verlag, 1999.

G. Bruns & P. Godefroid. Model checking with multi-valued logics. Technical Report ITD-03-44535H, Bell Labs, 2003.

N. Bulling, W. Jamroga, & J. Dix. Reasoning about temporal properties of rational play. *Annals of Mathematics and Artificial Intelligence*, 53(1-4):51–114, 2008. doi: 10.1007/s10472-009-9110-4. URL `https://doi.org/10.1007/s10472-009-9110-4`.

N. Bulling, W. Jamroga, & M. Popovici. Atl* with truly perfect recall: Expressivity and validities. In T. Schaub, G. Friedrich, and B. O'Sullivan, editors, *ECAI 2014 - 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic - Including Prestigious Applications of Intelligent Systems (PAIS 2014)*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pages 177–182. IOS Press, 2014. doi: 10.3233/978-1-61499-419-0-177. URL `https://doi.org/10.3233/978-1-61499-419-0-177`.

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

K. Deuser & P. Naumov. On composition of bounded-recall plans. *Artificial Intelligence*, 289:103399, 2020.

C. Dima & F. Tiplea. Model-checking ATL under imperfect information and perfect recall semantics is undecidable. *CoRR*, abs/1102.4225, 2011. URL `http://arxiv.org/abs/1102.4225`.

R. Fagin, J. Y. Halpern, Y. Moses, & M. Y. Vardi. *Reasoning about Knowledge.* MIT Press, 1995.

M. Fitting. Many-valued modal logics. *Fundamenta Informaticae*, 15(3-4):335–350, 1991.

M. Fitting. Many-valued modal logics II. *Fundamenta Informaticae*, 17:55–73, 1992.

P. Godefroid & R. Jagadeesan. On the expressiveness of 3-valued models. In *Proceedings of the 4th International Conference on Verification, Model Checkig, and Abstract Interpretation (VMCAI03)*, volume 2575 of *LNCS*, pages 206–222. Springer-Verlag, 2003.

V. Goranko & W. Jamroga. Comparing semantics for logics of multi-agent systems. *Synthese*, 139(2):241–280, 2004.

A. Gurfinkel & M. Chechik. Multi-valued model checking via classical model checking. In *Proceedings of the 14th International Conference on Concurrency Theory (CONCUR03)*, volume 2761 of *LNCS*, pages 266–280. Springer-Verlag, 2003.

W. v. Hoek & M. Wooldridge. Cooperation, knowledge, and time: Alternating-time temporal epistemic logic and its applications. *Studia Logica*, 75(1):125–157, 2003.

M. Huth & S. Pradhan. Consistent partial model checking. *Electronic Notes in Theoretical Computer Science*, 73:45–85, 2004. doi: 10.1016/j.entcs.2004.08.003. URL `https://doi.org/10.1016/j.entcs.2004.08.003`.

M. Huth, R. Jagadeesan, & D. A. Schmidt. A domain equation for refinement of partial systems. *Mathematical Structures in Computer Science*, 14(4):469–505, 2004. doi: 10.1017/S0960129504004268. URL `https://doi.org/10.1017/S0960129504004268`.

W. Jamroga. Some remarks on alternating temporal epistemic logic. In *Proceedings of Formal Approaches to Multi-Agent Systems (FAMAS 2003)*, pages 133–140, 2004.

W. Jamroga & J. Dix. Model checking abilities under incomplete information is indeed delta2-complete. In B. Dunin-Keplicz, A. Omicini, and J. A. Padget, editors, *Proceedings of the 4th European Workshop on Multi-Agent Systems EUMAS'06, Lisbon, Portugal, December 14-15, 2006*, volume 223 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2006. URL `http://ceur-ws.org/Vol-223/66.pdf`.

W. Jamroga & W. van der Hoek. Agents that know how to play. *Fundamenta Informaticae*, 62:1–35, 2004.

W. Jamroga, M. Knapik, D. Kurpiewski, & L. Mikulski. Approximate verification of strategic abilities under imperfect information. *Artificial Intelligence*, 277, 2019a.

W. Jamroga, M. Knapik, D. Kurpiewski, & L. Mikulski. Approximate verification of strategic abilities under imperfect information. *Artificial Intelligence*, 277, 2019b. doi: 10.1016/j.artint.2019.103172. URL `https://doi.org/10.1016/j.artint.2019.103172`.

W. Jamroga, V. Malvone, & A. Murano. Natural strategic ability. *Artif. Intell.*, 277, 2019c. doi: 10.1016/j.artint.2019.103170. URL `https://doi.org/10.1016/j.artint.2019.103170`.

W. Jamroga, V. Malvone, & A. Murano. Natural strategic ability under imperfect information. In E. Elkind, M. Veloso, N. Agmon, and M. E. Taylor, editors, *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '19, Montreal, QC, Canada, May 13-17, 2019*, pages 962–970. International Foundation for Autonomous Agents and Multiagent Systems, 2019d. URL `http://dl.acm.org/citation.cfm?id=3331791`.

W. Jamroga, B. Konikowska, D. Kurpiewski, & W. Penczek. Multi-valued verification of strategic ability. *Fundamenta Informaticae*, 175(1-4):207–251, 2020. doi: 10.3233/FI-2020-1955. URL `https://doi.org/10.3233/FI-2020-1955`.

M. Knapik, É. André, L. Petrucci, W. Jamroga, & W. Penczek. Timed ATL: forget memory, just count. *Journal of Artificial Intelligence Research*, 66:197–223, 2019. doi: 10.1613/jair.1.11612. URL `https://doi.org/10.1613/jair.1.11612`.

B. Konikowska & W. Penczek. Reducing model checking from multi-valued CTL$^*$ to CTL$^*$. In *Proceedings of the 13th International Conference on Concurrency Theory (CONCUR02)*, volume 2421 of *LNCS*, pages 226–239. Springer-Verlag, 2002.

B. Konikowska & W. Penczek. Model checking multi-valued modal $\mu$-calculus revisited. In *Proceedings of the International Workshop on Concurrency, Specification and Programming (CS&P04)*, volume 170 of *Informatik-Berichte*, pages 307–318. Humboldt University, 2004.

B. Konikowska & W. Penczek. Model checking for multivalued logic of knowledge and time. In *Proceedings of the 5th international joint conference on Autonomous Agents and Multiagent Systems (AAMAS06)*, pages 169–176. IFAAMAS, 2006.

F. Laroussinie, N. Markey, & G. Oreiby. On the expressiveness and complexity of ATL. *Logical Methods in Computer Science*, 4(2:7), May 2008. doi: 10.2168/LMCS-4(2:7)2008. URL `http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/LMO-lmcs08.pdf`.

A. Lomuscio & J. Michaliszyn. An abstraction technique for the verification of multi-agent systems against ATL specifications. In C. Baral, G. D. Giacomo, and T. Eiter, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Fourteenth International Conference, KR 2014, Vienna, Austria, July 20-24, 2014*. AAAI Press, 2014. URL `http://www.aaai.org/ocs/index.php/KR/KR14/paper/view/8003`.

A. Lomuscio & J. Michaliszyn. Verifying multi-agent systems by model checking three-valued abstractions. In G. Weiss, P. Yolum, R. H. Bordini, and E. Elkind, editors, *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015, Istanbul, Turkey, May 4-8, 2015*, pages 189–198. ACM, 2015. URL `http://dl.acm.org/citation.cfm?id=2772907`.

A. Lomuscio & J. Michaliszyn. Verification of multi-agent systems via predicate abstraction against ATLK specifications. In C. M. Jonker, S. Marsella, J. Thangarajah, and K. Tuyls, editors, *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems, Singapore, May 9-13, 2016*, pages 662–670. ACM, 2016. URL `http://dl.acm.org/citation.cfm?id=2937022`.

A. Lomuscio & F. Raimondi. Model checking knowledge, strategies, and games in multi-agent systems. In H. Nakashima, M. P. Wellman, G. Weiss, and P. Stone, editors, *5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006), Hakodate, Japan, May 8-12, 2006*, pages 161–168. ACM, 2006. doi: 10.1145/1160633.1160660. URL `https://doi.org/10.1145/1160633.1160660`.

A. Lomuscio, H. Qu, & F. Raimondi. 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, 2017. doi: 10.1007/s10009-015-0378-x. URL `https://doi.org/10.1007/s10009-015-0378-x`.

MCMAS$_{BR}$. MCMAS with bounded recall. 2021.
`http://vas.doc.ic.ac.uk/software`.

J.-J. C. Meyer & W. v. Hoek. *Epistemic Logic for AI and Computer Science*. Cambridge University Press, 1995. ISBN 052146014X.

H. Pan, Y. Li, Y. Cao, & Z. Ma. Model checking computation tree logic over finite lattices. *Theoretical Computer Science*, 612:45–62, 2016. doi: 10.1016/j.tcs.2015.10.014. URL `https://doi.org/10.1016/j.tcs.2015.10.014`.

P. Schobbens. Alternating-time logic with imperfect recall. *Electronic Notes in Theoretical Computer Science*, 85(2):82–93, 2004. doi: 10.1016/S1571-0661(05)82604-0. URL `https://doi.org/10.1016/S1571-0661(05)82604-0`.

S. Shoham & O. Grumberg. Monotonic abstraction-refinement for CTL. In K. Jensen and A. Podelski, editors, *Tools and Algorithms for the Construction and Analysis of*

*Systems, 10th International Conference, TACAS 2004, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2004, Barcelona, Spain, March 29 - April 2, 2004, Proceedings*, volume 2988 of *Lecture Notes in Computer Science*, pages 546–560. Springer, 2004. doi: 10.1007/978-3-540-24730-2_40. URL `https://doi.org/10.1007/978-3-540-24730-2\_40`.

S. Shoham & O. Grumberg. Multi-valued model checking games. *Journal of Computer and System Sciences*, 78(2):414–429, 2012. doi: 10.1016/j.jcss.2011.05.003. URL `https://doi.org/10.1016/j.jcss.2011.05.003`.

S. Vester. Alternating-time temporal logic with finite-memory strategies. In G. Puppis and T. Villa, editors, *Proceedings Fourth International Symposium on Games, Automata, Logics and Formal Verification, GandALF 2013, Borca di Cadore, Dolomites, Italy, 29-31th August 2013*, volume 119 of *EPTCS*, pages 194–207, 2013. doi: 10.4204/EPTCS.119.17. URL `https://doi.org/10.4204/EPTCS.119.17`.

S. Vijzelaar & W. J. Fokkink. Multi-valued simulation and abstraction using lattice operations. *ACM Transactions on Embedded Computing Systems*, 16(2):42:1–42:26, 2017. doi: 10.1145/3012282. URL `https://doi.org/10.1145/3012282`.