Natural Strategic Ability under Imperfect Information

Wojciech Jamroga
Institute of Computer Science,
Polish Academy of Sciences
Warsaw, Poland
wjamroga@in.tu-clausthal.de

Vadim Malvone
Université d'Evry
Evry, France
vadim.malvone@univ-evry.fr

Aniello Murano
Universitá degli studi di Napoli
"Federico II"
Naples, Italy
murano@na.infn.it

ABSTRACT

Strategies in game theory and multi-agent logics are mathematical objects of remarkable combinatorial complexity. Recently, the concept of *natural strategies* has been proposed to model more human-like reasoning about simple plans and their outcomes. So far, the theory of such simple strategic play was only considered in scenarios where all the agents have perfect information about the state of the game.

In this paper, we extend the notion of natural strategies to games with imperfect information. We also show that almost all the complexity results for model checking carry over from the perfect to imperfect information setting. That is, verification of natural strategies is usually no more complex for agents with uncertainty. This tells games of natural strategic ability clearly apart from most results in game theory and multi-agent logics.

KEYWORDS

[Agent Theories and Models] Logic and Game Theory; Logics for agents and multi-agents systems; [Verification and Validation of Agent-based Systems] Verification techniques for multi-agents systems, including model checking;

ACM Reference Format:
Wojciech Jamroga, Vadim Malvone, and Aniello Murano. 2019. Natural Strategic Ability under Imperfect Information. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019,* IFAAMAS, 9 pages.

1 INTRODUCTION

Game theory provides a powerful mathematical framework to reason about the interaction of autonomous, purposeful agents. It has seen numerous applications in robotics, computer science, and AI. An application domain of particular interest for the multi-agent systems community concerns formal verification, synthesis, and planning for reactive and embedded systems. As many relevant properties of multi-agent systems refer to *strategic abilities* of agents and their groups, the need for formal specification and verification of such properties is essential if we want to make sure that the systems operate in the way we want them to.

Logics for strategic reasoning. A fundamental contribution in this field is *alternating-time temporal logic* ATL and its syntactic extension ATL* [8]. Formulas of ATL are usually interpreted over *concurrent game structures (CGS)* which are labeled state-transition systems that model synchronous interaction among agents. For example, in a ticket vending machine, the ATL formula $\langle\langle c \rangle\rangle \mathsf{F}\mathrm{ticket}$ may be used to express that the customer c can ensure that he will eventually obtain a ticket, regardless of the actions of the other agents. The specification holds if c has a strategy whose every execution path satisfies $\mathsf{F}\mathrm{ticket}$. Clearly, it captures an important functionality requirement for any ticket vending machine. Similarly, in a bullet voting system, $\langle\langle v, crc \rangle\rangle \mathsf{G}(\mathsf{AF}\mathrm{voted}_{\mathrm{v,i}} \leftrightarrow \mathsf{AF}\mathrm{paid}_{\mathrm{crc,v}})$ says that the voter and the coercer have a collective strategy to ensure that the coercer will pay out the prearranged bribe whenever v has voted for the indicated candidate i. This is obviously an undesirable property for most voting systems, as it allows to establish a vote buying scheme between the coercer and the voter.

Natural strategies. Following the game-theoretic model of multi-step play, strategies in MAS are understood as conditional plans, and play a central role in reasoning about purposeful agents. Formally, strategies in ATL, as well as in other logics of strategic reasoning such as Strategy Logic [25, 47, 48], are defined as functions from sequences of system states (i.e., possible histories of the game) to actions. A simpler notion of *positional* a.k.a. *memoryless* strategies is formally defined by functions from states to actions. The approach makes sense from the mathematical point of view, and might be appropriate to reason about strategic abilities of a computer program with extensive computational power at hand. However, it is completely unrealistic for reasoning about human behavior. This is because humans are very bad at handling combinatorially complex objects. A human strategy should be relatively simple and intuitive (or "natural") in order for the person to understand it, memorize it, and execute it. This applies even more if the human agent is to come up with the strategy on his own. A similar concern can be raised for of artificial agents with limited memory and/or computational power, such as simple robots, sensors in an autonomous sensor network, and components of Internet of Things.

To address this problem, NatATL was introduced in [39]. The logic updates ATL by replacing the strategic operator $\langle\langle A \rangle\rangle \varphi$ with a bounded version $\langle\langle A \rangle\rangle^{\leq k} \varphi$, where $k \in \mathbb{N}$ denotes the complexity bound. To measure the complexity of strategies, it was assumed that they are represented by lists of guarded actions. For memoryless strategies, guards are boolean propositional formulas. For strategies with recall, guards are given as regular expressions over boolean propositional formulas.

Natural ability for imperfect information. A crucial issue in reasoning about MAS is the observational and epistemic limitations of the agents. The most important distinction is between *perfect* and *imperfect* information scenarios. In the former case, all the players have always full knowledge of the current state of the game. In the latter the players may have to make their decisions with limited information at hand. Both settings have been intensively investigated

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada.

in the literature, and applied in real-life domains. A classic approach to modeling imperfect information is to use indistinguishability relations over game states [9, 11–16, 18, 22–24, 26, 32, 38, 40, 42, 44–46, 50, 52]. Then, the agent is supposed to specify the same action for all indistinguishable states. This in turn has a huge impact on the related decision problems, such as game solving and model checking. Indeed, it is well known that solving multi-player games of imperfect information are computationally hard (non-elementary or even undecidable) [50, 52].

In this paper, we present a variant of NatATL for agents with imperfect information. To measure the complexity of strategies, we assume that they are represented by lists of guarded actions. For memoryless strategies, guards are epistemic boolean propositional formulas. For strategies with recall, guards are given as regular expressions over epistemic boolean propositional formulas. As technical results, we study the problem of model checking NatATL for both memoryless and memoryfull strategies. The complexity ranges from $\Delta_2^{\mathbf{P}}$ to **PSPACE** in the general case, and from **P** to $\Delta_2^{\mathbf{P}}$ for small complexity bounds. Thus, verification of natural strategies is usually no more complex for agents with uncertainty. This is an interesting result, and one that clearly separates games of natural strategic ability from most results in game theory and multi-agent logics.

Outline of the paper. The rest of the paper is organized as follows. In Sec.2 we introduce natural strategies with uncertainty. In Sec.3 we present NatATL for agents with imperfect information and in Sec.4 we study the model checking problem under imperfect recall strategies. In Sec.5 we introduce natural strategies with perfect recall and in Sec.6 we study the related model checking problem (under imperfect information). Finally, we conclude in Sec.7.

1.1 Related Work

Imperfect information games have been largely considered in the literature [21, 29, 36, 43, 52]. A seminal work is [52] in which a number of variants of 2-player games with imperfect information have been investigated. Generally, having imperfect information immediately reflects on worsening the complexity of solving the game. In multi-player games one can easily jump to non-elementary [50] or even to undecidability [29]. As an evidence we mention [51] where a pipeline of processes architecture is considered and each output communication of process i is taken as the input communication of process $i+1$. The reachability problem under imperfect information in this specific setting is decidable but complete for non-elementary time[1].

In many cases, solving the related decision problem for ATL* becomes undecidable [8, 29]. In particular it is undecidable in the case of three agents having perfect recall of the past, while it is elementary decidable in case the agents play positional (a.k.a. memoryless) strategies.

ATL* has been the subject of intensive research within multi-agent systems and AI. Works that are closest in spirit to our proposal concern modeling, specification, and reasoning about strategies of bounded agents. [3] investigates strategic properties of agents with bounded memory, while [11] considers bounded memory as an approximation of perfect recall. [6, 7, 19, 20] extend temporal and strategic logics to handle agents with bounded resources. Issues related to bounded rationality are also investigated in [10, 31, 35].

The papers that studied explicit representation of strategies are also relevant. This category is much richer and includes extensions of ATL* with explicit reasoning about actions and strategies [1, 34, 54, 56], and logics that combine features of temporal and dynamic logic [33, 49]. A variant of STIT logic, that enables reasoning about strategies and their performance in the object language, can be found in [30]. Also, plans in agent-oriented programming are in fact rule-based descriptions of strategies. In particular, reasoning about agent programs using strategic logics was investigated in [4, 5, 17, 28, 57].

2 NATURAL STRATEGIES UNDER UNCERTAINTY

We begin by showing that the notion of *natural strategies*, introduced in [39], can be adapted to imperfect information scenarios in a very simple way.

2.1 Models of Multi-Agent Interaction

The semantics of NatATL is defined over concurrent game structures with imperfect information, a variant of synchronous multi-agent transition systems.

Definition 2.1 (iCGS). A *concurrent game structure with imperfect information* (*iCGS* for short) is a tuple $M = \langle \mathbb{A}\mathrm{gt}, St, Act, d, t, Prop, V, \{\sim_i\}_{i\in\mathbb{A}\mathrm{gt}}\rangle$ which includes:

- nonempty finite set of agents (or players) $\mathbb{A}\mathrm{gt} = \{a_1, \ldots, a_{|\mathbb{A}\mathrm{gt}|}\}$,
- nonempty finite set of states St,
- nonempty finite set of actions Act,
- nonempty finite set of atomic propositions $Prop$,
- propositional valuation $V : St \to 2^{Prop}$,
- for every $a_i \in \mathbb{A}\mathrm{gt}$, $\sim_i$ is a relation of *indistinguishability* between states, that is, given two states $s, s' \in St$, $s \sim_i s'$ iff s and s' are indistinguishable for agent a_i,
- the function $d : \mathbb{A}\mathrm{gt} \times St \to 2^{Act}$ defines availability of actions,
- the (deterministic) transition function t assigns a successor state $q' = t(q, \alpha_1, \ldots, \alpha_{|\mathbb{A}\mathrm{gt}|})$ to each state $q \in St$ and any tuple of actions $\alpha_i \in d(a_i, q)$ that can be executed by $\mathbb{A}\mathrm{gt}$ in q.

In the rest of the paper, we will write $d_a(q)$ instead of $d(a, q)$, and we will denote the set of collective choices of group A in state q by $d_A(q) = \prod_{a_i \in A} d_{a_i}(q)$. Moreover, we require that the iCGS is *uniform*, i.e., $q \sim_i q'$ implies $d_i(q) = d_i(q')$.

A *pointed iCGS* is a pair (M, q_0) where M is a concurrent game structure and q_0 a state in M.

A *path* $\lambda = q_0 q_1 q_2 \ldots$ in an iCGS is an infinite sequence of states such that there is a transition between each q_i, q_{i+1}. $\lambda[i]$ denotes the ith position on λ (starting from $i = 0$), $\lambda[i, j]$ the part of λ between positions i and j, and $\lambda[i, \infty]$ the suffix of λ starting with i. By $\Lambda_M \subseteq St^\omega$, we denote the set of all the paths in M, and by $\Lambda_M(q)$ all the paths in M starting in q. Similarly, a *history* $h = q_0 q_1 q_2 \ldots q_n$ is a finite sequence of states that can be effected by subsequent transitions. By $last(h) = q_n$ we denote the last element of the sequence. We denote by $H_M \subseteq St^+$ the set of all the histories in

[1] Other settings have been also taken in consideration and leading to an undecidable problem.

model M. We will omit the subscripts whenever they are clear from the context.

Example 2.2 (Logistic Robots). Two robots (agents 1 and 2), serve a machine that produces electronic units for car control systems (agent 3). The machine has two output belts, marked A and B. The task of the robots is to pick up the units from the belts, so that they can be later delivered to the assembly room. The initial state, labeled by the atomic proposition init, corresponds to the situation when both robots serve the first belt (q_{AA}). At each moment, the robots can stay where they are (action st) or move to the other belt (action mv). At the same time, the machine can send a unit to one of the belts (actions A and B) or stay idle (action i). If the belt where the unit arrives is not served by any of the robots, the unit falls down and gets damaged. This is modeled by a system transition to the "failure" state q_f, labeled by proposition damage.

Robot 1 can recognize its own position, but does not see the other robot. Moreover, robot 2 can see if it is by the same belt as the other robot, but has no sensor to identify the exact position. The observational capabilities of the machine are irrelevant.

An iCGS M_{robots} for the scenario is depicted in Figure 1. States, transitions (represented by solid arrows), indistinguishability relations (represented by dotted lines), and valuation of atomic propositions can be easily derived from the picture.

2.2 Simple Strategies for Uncertain Agents

We start by defining the notion of *uniform natural memoryless strategy* (or *nir*-strategy[2]) s_a for agent a. The idea is to use a rule-based representation, with a list of *condition-action* rules. The first rule whose condition holds in the current state is selected, and the corresponding action is executed.

Formally, we define the set of *epistemic conditions* EP for the agent a as follows:

$$\psi ::= \top \mid K_a\varphi \mid \neg\psi \mid \psi \wedge \psi$$
$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_b\varphi$$

where p is an atomic proposition and b an agent.

So, we are talking about formulas that are prefixed by K_a and then possibly combined by Boolean operators. In other words, formulas ψ are always Boolean conditions on a's knowledge.

Given an iCGS M, a state $q \in St$, and an epistemic condition φ, we inductively define whether q satisfies φ ($q \models \varphi$) and consequently as follows:

$q \models p$ iff $p \in V(q)$;
$q \models \neg\varphi$ iff $q \models \varphi$ does not hold;
$q \models \varphi_1 \wedge \varphi_2$ iff $q \models \varphi_1$ and $q \models \varphi_2$;
$q \models K_a\varphi$ iff for all $q' \sim_a q$, it holds that $q' \models \varphi$;

We represent uniform natural strategies by *lists of guarded actions*, i.e., sequences of n pairs $(\varphi_i,\ \alpha_i)$ such that: (1) φ_i is an epistemic condition, and (2) $\alpha_i \in d_a(q)$ for every $q \in St$ such that $q \models \varphi_i$. That is, φ_i is an epistemic condition on states of the iCGS, and α_i is an action available to agent a in every state where φ_i holds. Moreover, we assume that the last pair on the list is $(\top,\ \alpha)$, with $\alpha \in d_{Ag}(q)$, for all $q \in St$ and some $\alpha \in Act$, i.e., the last rule is guarded by a condition that will always be satisfied and contains an action that is executable in every state.

[2] As usual in literature, we use i for imperfect information and r for imperfect recall (memoryless). Additionally, we use n for natural strategies, thus the acronym.

The set of all uniform natural memoryless strategies of agent a is denoted by Σ_a^{nir}. By $size(s_a)$ we denote the number of guarded actions in s_a. Moreover, $cond_i(s_a)$ will denote the ith guard (condition) on the list, and $act_i(s_a)$ the corresponding action. Finally, $match(q, s_a)$ is the smallest $i \leq size(s_a)$ such that $q \models cond_i(s_a)$ and $act_i(s_a) \in d_a(q)$. That is, $match(q, s_a)$ matches state q with the first condition in s_a that holds in q, and action available in q.

It is easy to see that the strategies are uniform in the sense of [41, 53], i.e., they specify the same actions in indistinguishable states.

Proposition 2.3. *Given a uniform natural memoryless strategy s_a and two states such that $q \sim_a q'$, we have that $act_{match(q,s_a)}(s_a) = act_{match(q',s_a)}(s_a)$.*

Proof. Take $s_a = \big((\varphi_1, \alpha_1), \ldots, (\varphi_n, \alpha_n)\big)$ and any pair of states q, q' such that $q \sim_a q'$. Let $match(q, s_a) = i$. That is, φ_i holds in q, but every φ_j for $j < i$ does not. Since all of φ_i, φ_j are either equal to $\top$ or begin with K_a, they must either hold in both q, q', or in none of them. Thus, we get that $match(q', s_a) = i$, too. □

A *collective uniform natural strategy* for a group of agents $A = \{a_1, \ldots, a_{|A|}\}$ is a tuple of individual uniform natural strategies $s_A = (s_{a_1}, \ldots, s_{a_{|A|}})$. The set of such strategies is denoted by Σ_A^{nir}. The "outcome" function $out(q, s_A)$ returns the set of all paths that occur when agents A execute strategy s_A from state q onward. Formally, given a state $q \in St$, a subset of agents A and a collective uniform memoryless strategy s_A, we define:

$$out(q, s_A) = \{\lambda \in \Lambda \mid (\lambda[0] = q) \ \wedge\ \forall_{i\geq 0}\exists_{\alpha_1,\ldots,\alpha_{|\mathbb{A}gt|}} \cdot$$
$$(a \in A \Rightarrow \alpha_a = act_{match(\lambda[i],s_a)}(s_a)) \wedge$$
$$(a \notin A \Rightarrow \alpha_a \in d_a(\lambda[i])) \wedge (\lambda[i+1] = t(\lambda[i], \alpha_1, \ldots, \alpha_{|\mathbb{A}gt|}))\}.$$

Example 2.4 (Logistic Robots, ctd.). The following collective natural strategy can be used by the robots to ensure that the goods never get damaged in the iCGS of Example 2.2:

$$s_1: \quad (\ \ (\top,\ st)\ \ );$$
$$s_2: \quad (\ \ (\neg K_2\neg \mathrm{init},\ mv),\ (\top,\ st)\ \ ).$$

By looking at Figure 2.2 it is evident that s_1 will force robot 1 to always perform action st, and robot 2 to play action mv only in q_{AA} and q_{BB} (where he does not know whether init holds or not) and st elsewhere.

2.3 How to Measure Natural Strategies

By $compl(s_a)$, we denote the complexity of the strategy s_a. Intuitively, the complexity of a strategy is understood as the level of sophistication of its representation. Several natural metrics can be used to measure the complexity of a strategy, given its representation from $(\mathrm{EP}(Prop) \times Act)^+$, e.g.:

Used vocabulary: $compl_{\#}(s_a) = |dom(s_a)|$ where $dom(s_a)$ is the set of atomic propositions and epistemic operators used in the conditions of s_a;
Largest condition: $compl_{\max}(s_a) = \max\{|\varphi| \mid (\varphi, \alpha) \in s_a\}$;
Total size of the representation: $compl_{\Sigma}(s_a) = \sum_{(\varphi,\alpha)\in s_a} |\varphi|$

with $|\varphi|$ being the number of symbols in φ, without parentheses. From now on, we will focus on the last metric for complexity of

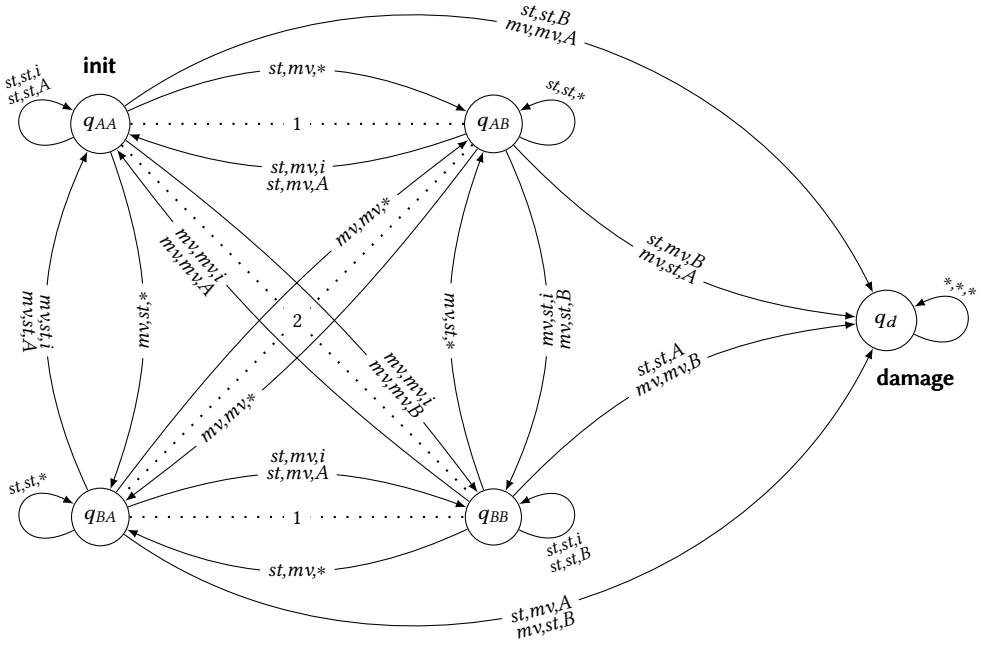

Figure 1: Logistic robots serving the output of a machine. The wildcard (∗) matches any action of the respective agent

strategies, which takes into account the total size of all the conditions used in the representation. That is, unless explicitly specified, we will assume $compl(s_a) = compl_\Sigma(s_a)$.

For a collective uniform natural strategy s_A, we define its complexity as the total complexity of the individual strategies in s_A. Formally, given $s_A = (s_{a_1}, \ldots, s_{a_{|A|}})$, we have that $compl(s_A) = \sum_{1 \le i \le |A|} compl(s_{a_i})$.

Example 2.5 (Logistic Robots, ctd.). For the natural strategy $s_{\{1,2\}}$ in Example 2.4, we get $compl(s_{\{1,2\}}) = 6$. Note that an even sparser representation of the same plan can be obtained by replacing robot 2's strategy with:

$$s_2' : \quad \big(\quad (K_2 \neg \mathsf{init},\ st), \quad (\top,\ mv) \quad \big),$$

which reduces the complexity of the joint strategy to 5.

3 REASONING ABOUT NATURAL ABILITY

Now we can propose a logic for reasoning about natural strategic ability under imperfect information. To achieve that, we use exactly the same syntax as for natural strategies with perfect information [39]. The semantics updates the one from [39] with the requirement that only uniform natural strategies are allowed.

3.1 A Logic for Natural Strategies

Natural ATL (NatATL, for short) is obtained by replacing in ATL the modality $\langle\langle A \rangle\rangle$ with the bounded strategic modality $\langle\langle A \rangle\rangle^{\le k}$. Intuitively, $\langle\langle A \rangle\rangle^{\le k} \gamma$ reads as "coalition A has a collective uniform strategy of *size less or equal than k* to enforce the property γ." As in ATL, the formulas of NatATL make use of classical temporal operators: X ("in the next state"), U (strong "until"), and W (weak "until"). Thus, the language of NatATL can be defined by the following grammar:

$$\varphi ::= \mathsf{p} \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\langle A \rangle\rangle^{\le k} \mathsf{X}\, \varphi \mid \langle\langle A \rangle\rangle^{\le k} \varphi\, \mathsf{U}\, \varphi \mid \langle\langle A \rangle\rangle^{\le k} \varphi\, \mathsf{W}\, \varphi.$$

where $\mathsf{p} \in Prop$, $A \subseteq \mathbb{A}\mathrm{gt}$, and $k \in \mathbb{N}$, given in a unary encoding.

Additionally, operators F ("now or sometime in the future") and G ("always from now on") are defined as follows: $\mathsf{F}\varphi \equiv \top\, \mathsf{U}\, \varphi$, $\mathsf{G}\varphi \equiv \varphi\, \mathsf{W}\, \bot$.

We will use 1NatATL to denote the fragment of NatATL that admits only formulas consisting of a single strategic modality, followed by a temporal formula over boolean connectives and atomic propositions. It would be interesting to consider the broader specification language of NatATL*, and extend our results accordingly. We leave this angle for future work.

3.2 Semantics of NatATL under Imperfect Information

Given an iCGS M, a state $q \in St$, a path $\lambda \in \Lambda$, and $k \in \mathbb{N}$, the semantics of NatATL is defined as follows:

- $M, q \models_{nir} \mathsf{p}$ iff $\mathsf{p} \in V(q)$, for $\mathsf{p} \in Prop$;
- $M, q \models_{nir} \neg\varphi$ iff $M, q \models_{nir} \varphi$ does not hold;
- $M, q \models_{nir} \varphi_1 \wedge \varphi_2$ iff $M, q \models_{nir} \varphi_1$ and $M, q \models_{nir} \varphi_2$;
- $M, q \models_{nir} \langle\langle A \rangle\rangle^{\le k} \mathsf{X}\, \varphi$ iff there is a uniform natural strategy $s_A \in \Sigma_A^{nr}$ such that $compl(s_A) \le k$ and, for each path $\lambda \in out(q, s_A)$, we have $M, \lambda[1] \models_{nir} \varphi$;

$M, q \models_{nir} \langle\langle A\rangle\rangle^{\leq k} \varphi \mathrel{U} \psi$ iff there is a uniform natural strategy $s_A \in \Sigma_A^{nir}$ such that $compl(s_A) \leq k$ and, for each path $\lambda \in out(q, s_A)$, we have $M, \lambda[i] \models_{nir} \psi$ for some $i \geq 0$ and $M, \lambda[j] \models_{nir} \varphi$ for all $0 \leq j < i$.

$M, q \models_{nir} \langle\langle A\rangle\rangle^{\leq k} \varphi \mathrel{W} \psi$ iff there is a uniform natural strategy $s_A \in \Sigma_A^{nir}$ such that $compl(s_A) \leq k$ and, for each path $\lambda \in out(q, s_A)$, we have either that $M, \lambda[i] \models_{nir} \psi$ for some $i \geq 0$ and $M, \lambda[j] \models_{nir} \varphi$ for all $0 \leq j < i$, or that $M, \lambda[i] \models_{nir} \varphi$ for all $i \geq 0$.

We will refer to the logical system (NatATL, $\models_{nir}$) as NatATL$_{ir}$, and analogously for 1NatATL$_{ir}$.

Example 3.1 (Logistic Robots, ctd.). Following Example 2.5, we get $M_{robots}, q_{AA} \models \langle\langle 1, 2\rangle\rangle^{\leq 5}\mathrm{G}\neg\mathrm{damage}$. It is also easy to see that no simpler strategy works. Thus, $M_{robots}, q_{AA} \models \neg\langle\langle 1, 2\rangle\rangle^{\leq 4}\mathrm{G}\neg\mathrm{damage}$.

Remark 1 (Objective vs. subjective semantics of ability). *We note that the above semantics encodes the "objective" notion of strategic ability. That is, $\langle\langle A\rangle\rangle\gamma$ holds in q if the agents have a joint strategy to enforce γ from q. The alternative is to require the existence of a strategy that succeeds from all the states that are indistinguishable from q for A (the "subjective" semantics). We refer the interested reader to [2] for an in-depth discussion.*

The subjective semantics of NatATL$_{ir}$ *can be obtained by a simple modification of the semantic clauses for $\langle\langle A\rangle\rangle^{\leq k}$. We also note that the model checking results, presented in the subsequent sections, can be adapted to the subjective case in a straightforward way.*

4 MODEL CHECKING FOR NATATL

In this section we study algorithms and the complexity of the model checking problem for NatATL with *nir*-strategies, i.e. NatATL$_{ir}$. We consider two cases: one in which the bound k on the size of natural strategies is assumed to be constant, and the more general case where k is variable and a parameter of the problem. For the former case, we prove that the problem is polynomial in the size of the model. For the latter, model checking becomes $\mathbf{\Delta_2^P}$-complete. Moreover, it is **NP**-complete for simple formulas with only one strategic operator, i.e., formulas of 1NatATL$_{ir}$.

The results and the proofs proposed in this section have been inspired by [37, 39, 53].

4.1 Model Checking for Small Natural Strategies

We begin by looking at NatATL$_{ir}$ model checking under the assumption that the complexity bounds k used in formulas are constant or bounded. In other words, they are not a parameter of the model checking problem. Under this restriction, one can show a polynomial reduction to the model checking problem for CTL formulas. Thus, we obtain the following result.

Theorem 4.1. *The model checking problem for* NatATL$_{ir}$ *with fixed k is in* **P** *with respect to the size of the model and the length of the formula.*

Proof. First, consider the formula $\varphi = \langle\langle A\rangle\rangle^{\leq k}\gamma$, in which $A \subseteq \mathbb{A}\mathrm{gt}$ and γ is a formula over boolean connectives and atomic propositions. By assumption, the collective strategy that we can assign to coalition A, namely s_A, is bounded and precisely it holds that $compl_\Sigma(s_A) \leq k$. Recall that epistemic boolean formulas are combinations of atomic propositions from the set *Prop*, boolean connectives from *Bool*, and the epistemic operators *Kn*. Thus, there are $O((|Prop| + |Bool| + |Kn|)^k \cdot |Act|)$ possible guarded actions of length at most k, and hence $O(((|Prop| + |Bool| + |Kn|)^k \cdot |Act|)^k) = O((|Prop| + |Bool| + |Kn|)^{k^2}(|Act|)^k)$ possible strategies, which is ensured to be finite as k is fixed. Note that the strategies are uniform by construction, cf. Proposition 2.3. The idea is to check the available collective strategies one by one, in an arbitrary order.

Given a collective strategy s_A, we can prune the *CGS* by removing all the edges that disagree with s_A. This operation costs $O(|t|)$ in the worst case, where t is the transition relation of the input *iCGS*. So far we have dealt with the strategic operator in the input formula φ, and we are left with a structure S that can be seen as a Kripke structure. Now, we can reduce our problem to model checking the CTL formula $\mathrm{A}\gamma$ ("for all paths γ") over S by using the standard model checking algorithm for CTL [27], well-known to have complexity $O(|t| \cdot |\gamma|)$. The total complexity is thus $O((|Prop| + |Bool| + |Kn|)^{k^2}(|Act|)^k \cdot O(|A| \cdot max(\{|\sim_a| \mid a \in A\})) \cdot (|t| + (|t| \cdot |\gamma|))) = O((|Prop| + |Bool| + |Kn|)^{k^2}(|Act|)^k \cdot O(|A| \cdot max(\{|\sim_a| \mid a \in A\})) \cdot |t| \cdot |\gamma|)$, and hence polynomial in the size of the model and the length of the formula.

To conclude the proof, note that if we have a formula with more strategic operators then we can use a classic bottom-up procedure. I.e., we start by solving the innermost subformula with a strategic operator (as we have done above) and, once this is solved, we update the formula and the structure, and continue with the new innermost subformula. The procedure ends when we have dealt with the outermost strategic operator in the input formula. □

4.2 Model Checking: General Case

We now study the complexity for NatATL$_{ir}$ with the bounds in strategic modalities given as variables.

Theorem 4.2. *Model checking* 1NatATL$_{ir}$ *is* **NP***-complete with respect to the size of the model, the length of the formula, and the maximal bound k in the formula.*

Proof. For the lower bound, we recall that 1NatATL$_{Ir}$ is **NP**-hard [39]. For the upper bound, consider $\varphi = \langle\langle A\rangle\rangle^{\leq k}\gamma$, in which $A \subseteq \mathbb{A}\mathrm{gt}$ and γ is a formula over boolean connectives and atomic propositions. We cannot anymore enumerate all the suitable strategies and check them one by one. To overcome this, we nondeterministically guess a uniform collective strategy s_A, and proceed using the same reasoning as in the proof of Theorem 4.1. Since the size of s_A is polynomial in the size of the model, the complexity of the algorithm is **NP**. □

To establish the model checking complexity for all formulas of NatATL$_{ir}$, we adapt the above proofs in a way similar to [37, 39].

Theorem 4.3. *Model checking* NatATL$_{ir}$ *is $\mathbf{\Delta_2^P}$-complete with respect to the size of the model, the length of the formula, and the maximal bound k in the formula.*[3]

[3] $\mathbf{\Delta_2^P} = \mathbf{P^{NP}}$ is the class of problems solvable in polynomial time by a deterministic Turing machine making adaptive calls to an oracle for problems in **NP**.

Proof. For the lower bound, we recall that NatATL$_{lr}$ is $\Delta_2^{\mathbf{P}}$-hard [39]. For the upper bound, we make use of a bottom-up procedure based on the one introduced in the proof of Theorem 4.1. Precisely, take an arbitrary formula φ of NatATL$_{ir}$ and consider its inner part that is of the kind $\psi = \langle\langle A\rangle\rangle^{\leq k}\gamma$, with γ being a formula over boolean connectives and atomic propositions. Now, apply over ψ the procedure used in the proof of Theorem 4.2 that we know to be **NP**. Once ψ is solved, use the same **NP** procedure to solve ψ', a formula that contains ψ and a strategic operator, and so on for each strategic operator in φ. This means that we use an oracle over a polynomial procedure for each strategic operator in φ. Summing up, the total complexity to solve a formula in NatATL$_{ir}$ is $\mathbf{P}^{\mathbf{NP}} = \Delta_2^{\mathbf{P}}$. □

5 NATURAL ABILITIES OF AGENTS WITH MEMORY

Agents with memory can base their decisions on the history of the game, i.e., the sequence of states that has occurred so far. How can we represent conditions on such sequences? One possibility is to use states in some kind of automaton [55]. Here, we suggest that it is more intuitive for humans to represent conditions on histories by regular expressions over epistemic propositional formulas.

5.1 Natural Strategies with Recall

Let $Reg(L)$ be the set of regular expressions over the language L (with the standard constructors $\cdot, \cup, *$ representing concatenation, nondeterministic choice, and finite iteration). A *uniform natural strategy with recall* (or *niR*-strategy) s_a for agent a is a sequence of appropriate pairs from $Reg(\mathrm{EP}(Prop)) \times Act$. That is, it consists of pairs $(r,\ \alpha)$ where r is a regular expression over EP($Prop$), and α is an action available in $last(h)$, i.e. $\alpha \in d_a(last(h))$, for all histories $h \in H$ consistent with r. Formally, given a regular expression r and the language $L(r)$ on words generated by r, a history $h = q_0 \dots q_n$ is consistent with r iff $\exists b \in L(r)$ such that $|h| = |b|$ and $\forall_{0\leq i\leq n} h[i] \models b[i]$. Similarly to *nir*-strategies, the last pair on the list is assumed to be simply $(\top^*,\ idle)$. The set of such strategies is denoted by Σ_a^{niR}. Finally, $match(\lambda[0,i], s_a)$ is the smallest $n \leq size(s_a)$ such that $\forall_{0\leq j\leq i}\lambda[j] \models cond_n(s_a)[j]$ and $act_n(s_a) \in d_a(\lambda[i])$.

Again, we observe that the strategies are uniform in the sense of [41, 53], i.e., they specify the same actions in indistinguishable sequences of states.

Proposition 5.1. *Given a uniform natural strategy with recall s_a and two histories h, h' such that $|h| = |h'|$ and $\forall i\ .\ h[i] \sim_a h'[i]$, we have that $act_{match(h,s_a)}(s_a) = act_{match(h',s_a)}(s_a)$.*

Proof. The result follows directly by considering the proof given in Prop.2.3. In particular, take $s_a = \big((\varphi_1,\alpha_1),\dots,(\varphi_n,\alpha_n)\big)$ and any pair of histories h, h' such that $|h| = |h'|$ and $\forall i \leq |h|$, $h[i] \sim_a h'[i]$. Let $match(h, s_a) = m$. That is, $\forall i \leq |h|$, $\varphi_m[i]$ holds in $h[i]$, but every $\varphi_j[i]$ for $j < m$ does not. Since all of $\varphi_m[i], \varphi_j[i]$ are either equal to $\top$ or begin with K_a, they must either hold in both $h[i], h'[i]$, or in none of them. Thus, we get that $match(h', s_a) = m$, too. □

A *collective uniform natural strategy* for a group of agents $A = \{a_1,\dots,a_{|A|}\}$ is a tuple of individual uniform natural strategies $s_A = (s_{a_1},\dots,s_{a_{|A|}})$. The set of such strategies is denoted by Σ_A^{niR}.

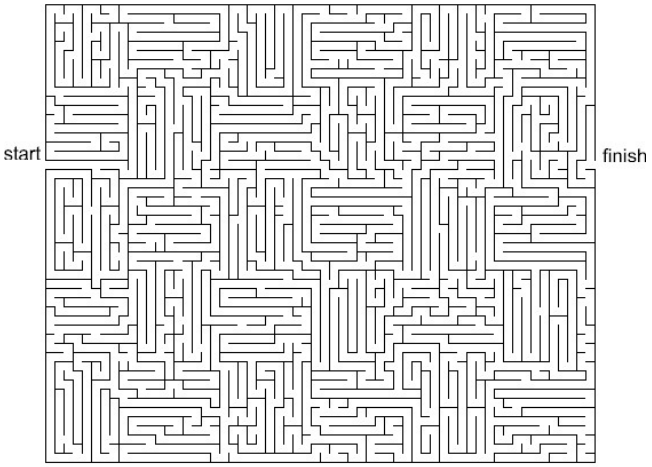

Figure 2: A maze with no loops

Again, $out(q, s_A)$ returns the set of all paths from q, consistent with strategy s_A. For strategies with recall, we simply replace "$match(\lambda[i], s_a)$" with "$match(\lambda[0,i], s_a)$" in the definition of $out(q, s_A)$ that we gave in Section 2.2 for memoryless strategies.

The metrics from Section 2.2 extend to strategies with recall and collective strategies with recall in the straightforward way. Additionally, we define a variant of the metric $compl_\Sigma(\cdot)$ that skips the initial $\top^*$ whenever it appears in a regular expression:

Size of the significant pattern: $compl_{\Sigma^*}(s_a) = \sum_{(r,\alpha)\in s_a} ||r||$, with $||\top^*|| = 1$, $||\top^* \cdot r|| = |r|$, and $||r|| = |r|$ otherwise. The latter is the size of a regular expression that is, as usual, the number of symbols it contains, including the syntactic symbols such as brackets, $+$, $\cdot$, and $*$.[4]

From now on, we will focus on the last metric for the complexity of strategies with recall. That is, unless explicitly specified, we will assume $compl(s_a) = compl_{\Sigma^*}(s_a)$.

Example 5.2 (Foggy maze). Consider agent r ("the rover") whose goal is to get through a maze, such as the one in Figure 2. We assume that the maze is *perfect*, i.e., it has no loops. Moreover, it is inhabited by a number of other, hostile agents. Each agent can, at any moment, decide to turn left (action $turn_L$), turn right (action $turn_R$), move forward ($step$) or do nothing ($wait$). Moving succeeds if there is neither a wall nor another agent in front. Agent r can also execute action $destroy$ that annihilates the agent standing in front of him, if there is one. The maze is sometimes overtaken by fog, in which case the agents see nothing for 1 or 2 moments.

Assume an iCGS M_{maze} modeling the scenario as follows. The states register the positions and the orientation of all the agents. Two states are indistinguishable to an agent i iff they agree on the position and the orientation of i, and:

- either in both states the maze is foggy,
- or in both states the fog is absent, and they agree on the content of the cell in front of i.

[4] Sometimes we will also use as an abbreviation $(r)^i$ to denote the concatenation of r i-times. In this case the size of $|(r)^i| = i|r| + (i-1)$.

The atomic propositions start and finish label the states where the rover is respectively at the maze entry and exit. Propositions wall and creature label the states where the rover faces respectively a wall or another agent.

The following natural strategy with recall guarantees that the rover gets through the maze (we use ψ_{fog} as a shorthand for $\neg K_r\text{creature} \wedge \neg K_r\neg\text{creature}$ to simplify the notation):

$$\begin{aligned} s_r : \quad (\quad & (\top^* \cdot \psi_{\text{fog}},\ \mathit{wait}),\\ & (\top^* \cdot K_r\text{creature},\ \mathit{destroy}),\\ & (\top^* \cdot K_r\neg\text{wall},\ \mathit{step}),\\ & (\top^* \cdot \neg K_r\text{wall} \cdot {\psi_{\text{fog}}}^* \cdot K_r\text{wall},\ \mathit{turn}_L),\\ & (\top^* \cdot \neg K_r\text{wall} \cdot ({\psi_{\text{fog}}}^* \cdot K_r\text{wall})^2,\ \mathit{turn}_R),\\ & (\top^* \cdot \neg K_r\text{wall} \cdot ({\psi_{\text{fog}}}^* \cdot K_r\text{wall})^3,\ \mathit{turn}_R),\\ & (\top^*,\ \mathit{turn}_R) \qquad). \end{aligned}$$

That is, if the fog is in the maze, the rover waits until it clears; if he sees an enemy, he destroys it. If he faces a wall, he tries first to turn left; if there is a wall as well, he keeps turning right until he finds a passage.

The complexity of the strategy is $\mathit{compl}(s_r) = \{8 + 2 + 3 + 16 + 29 + 42 + 1\} = 101$. Note also that, if we add to the model an atomic proposition fog that labels all the "foggy" states, we can use it instead of ψ_{fog} to specify the same behavior. This would reduce the complexity of the strategy to $\mathit{compl}(s_r') = \{1{+}2{+}3{+}9{+}15{+}21\} = 51$.

Finally, we observe that the strategy may result in a very ineffective traversal of the maze, i.e., the number of steps between the start and the exit can be large. Still, the natural strategy above has two important advantages. First, it is much simpler – and therefore much easier to store and use – than the combinatorial strategy that specifies the right choice for every position of the rover. Secondly, it is general in the sense that it does not depend on the actual shape of the labyrinth.

5.2 NatATL for Strategies with Recall

Now it is easy to define the semantics of natural strategic ability for agents with recall. Formally, we construct the semantic relation $\models_{niR}$ by replacing "$\models_{nir}$" with "$\models_{niR}$" and Σ_A^{nir} with Σ_A^{niR} in the clauses from Section 3.2, so that the clauses for strategic modalities become as follows:

- $M, q \models_{niR} \langle\langle A\rangle\rangle^{\leq k} \mathsf{X}\,\varphi$ iff there is a uniform natural strategy $s_A \in \Sigma_A^{niR}$ such that $\mathit{compl}(s_A) \leq k$ and, for each path $\lambda \in \mathit{out}(q, s_A)$, we have $M, \lambda[1] \models_{niR} \varphi$;
- $M, q \models_{niR} \langle\langle A\rangle\rangle^{\leq k} \varphi\,\mathsf{U}\,\psi$ iff there is a uniform natural strategy $s_A \in \Sigma_A^{niR}$ such that $\mathit{compl}(s_A) \leq k$ and, for each path $\lambda \in \mathit{out}(q, s_A)$, we have $M, \lambda[i] \models_{niR} \psi$ for some $i \geq 0$ and $M, \lambda[j] \models_{niR} \varphi$ for all $0 \leq j < i$.
- $M, q \models_{niR} \langle\langle A\rangle\rangle^{\leq k} \varphi\,\mathsf{W}\,\psi$ iff there is a uniform natural strategy $s_A \in \Sigma_A^{niR}$ such that $\mathit{compl}(s_A) \leq k$ and, for each path $\lambda \in \mathit{out}(q, s_A)$, we have either that $M, \lambda[i] \models_{niR} \psi$ for some $i \geq 0$ and $M, \lambda[j] \models_{niR} \varphi$ for all $0 \leq j < i$, or that $M, \lambda[i] \models_{niR} \varphi$ for all $i \geq 0$.

We will refer to the logical system (NatATL, $\models_{niR}$) as NatATL$_{iR}$.

Example 5.3 (Foggy maze, ctd.). For the maze model in Example 5.2, we have e.g. that $M_{maze}, q_{start} \models \langle\langle r\rangle\rangle^{\leq 93}\mathsf{F}\,\text{finish}$.

Algorithm $mCheck^{const}_{\text{NatATL}_{iR}}(M, q, \langle\langle A\rangle\rangle^{\leq k}\gamma)$:

```
for every s_A with compl(s_A) ≤ k do
  if not IsLosing(s_A, M, q, p1 U p2) then return(true);
return (false);
```

Figure 3: Model checking NatATL$_{iR}$ for simple goals, i.e., $\gamma \equiv \mathsf{p}_1\,\mathsf{U}\,\mathsf{p}_2$ or $\gamma \equiv \mathsf{p}_1\,\mathsf{W}\,\mathsf{p}_2$. The value of k is bounded by a constant

6 MODEL CHECKING FOR NATURAL STRATEGIES WITH RECALL

In this section we investigate the model checking problem for NatATL with *niR*-strategies, i.e., NatATL$_{iR}$. We consider the cases of both constant and variable bounds on strategies. We begin with the following lemma.

Lemma 6.1. *Given an iCGS M, a uniform natural strategy with recall $s_A = (s_{a_1}, \ldots, s_{a_n}) \in \Sigma_A^{niR}$ of size $k = \mathit{compl}(s_A)$, and a reachability objective $\gamma \equiv \mathsf{p}_1\,\mathsf{U}\,\mathsf{p}_2$. In order to check if s_A enforces γ from $q \in St_M$, it suffices to consider the prefixes of length $|St_M| \cdot 2^{2k^2}$ of paths in $\mathit{out}(q, s_A)$. The same applies to safety objectives ($\mathsf{p}_1\,\mathsf{W}\,\mathsf{p}_2$).*

Proof. Consider the tree of paths in M, starting from q and consistent with s_A. It can be obtained by an (infinite) process, based on the following notion of configuration: $C = (q_M, q_{reg_1}, \ldots, q_{reg_n})$ where q_M is the current state of M, and every q_{reg_i} is the current state of a deterministic finite automaton (DFA) accepting the ith regular expression in s_A. The initial configuration C_0 consists of q and the initial states of the DFA's.

Let C be the current configuration. The process takes, for each agent $a \in A$, the first DFA for a regular expression in s_a that is currently in an accepting state, and selects the corresponding action in s_a for execution by a. Then, for every possible tuple of responses from $\mathbb{A}\text{gt} \setminus A$, a transition is added, leading to the configuration C' consisting of the successor state q' in M and the states of the DFA's updated accordingly. Note that, whenever the process revisits a previously encountered configuration, exactly the same transitions as before are added. Thus, whatever reachability objective can be validated (resp. safety objective invalidated), it can be done on the initial, cycle-free segment of the tree.

Finally, observe that s_A contains at most k regular expressions, and each expression is of length at most k. For every regular expression of length ℓ, there exists an equivalent nondeterministic finite automaton (NFA) with at most 2ℓ states (Thompson's construction). Finally, for every NFA with n states, there exists an equivalent DFA with at most 2^n states (powerset construction). Thus, the number of configurations is at most $|St_M| \cdot (2^{2k})^k = |St_M| \cdot 2^{2k^2}$. □

6.1 Model Checking for Small Strategies

When the bound on strategies is fixed or bounded by some constant **K**, model checking can again proceed by checking the available strategies one by one. The algorithm for formulas with no nested strategic modalities is shown in Figure 3.

Note that the verification of strategies is somewhat more involved than in the memoryless case. To this end, we use an oracle

Algorithm *IsLosing*$(s_A, M, q, \mathrm{p}_1 \mathrm{U}\, \mathrm{p}_2)$:

```
1  count := 0; state := q;
2  size := compl(s_A); limit := 2^{2·size^2};
3  for every regular expression r_i ∈ s_A do
4     initialize the NFA A_i for r_i;
5  repeat
6     if M, state ⊨ p_2 then return(false);
7     if M, state ⊭ p_1 then return(true);
8     for each a ∈ A do
9        match := minimal i such that r_i ∈ s_a
10          and A_i is in an accepting state;
11       α_a := act_match(s_a);
12    for each a ∉ A do
13       nondeterministically choose α_a ∈ d(a, state);
14    state := t(state, α_1, ..., α_|𝔸gt|);
15    for every NFA A_i do
16       nondeterministically update the state of A_i;
17    count := count + 1;
18 until count > limit;
19 return(true);
```

Figure 4: Oracle that tries to invalidate strategy s_A for a simple reachability goal $\mathrm{p}_1 \mathrm{U}\, \mathrm{p}_2$

Algorithm *mCheck*$_{\mathrm{NatATL}_{iR}}(M, q, \langle\langle A\rangle\rangle^{\leq k}\gamma)$:

```
1  guess a strategy s_A ∈ Σ_A^{niR} with compl(s_A) ≤ k;
2  return (not IsLosing(s_A, M, q, p_1 U p_2));
```

Figure 5: Model checking NatATL$_{iR}$ for simple goals; k is a parameter of the problem

IsLosing that returns "true" if it manages to guess a path invalidating the goal γ, and "false" otherwise. The case of simple reachability goals is presented in Figure 4; for simple safety goals, the oracle is defined analogously. It proceeds by nondeterministically unfolding a path consistent with strategy s_A from state q on until it either fulfills the goal, invalidates it, or exceeds the limit determined in Lemma 6.1. Notice that the oracle uses NFA implementations of the regular expressions in s_A, and *not* the DFA's that were employed in the proof of Lemma 6.1.

The complexity of the procedure is as follows. The oracle runs in $O(2^{2\mathbf{K}^2}) + O(\mathbf{K}) + O(|St_M| \cdot 2^{2\mathbf{K}^2} \cdot (\mathbf{K}|\mathbb{A}\mathrm{gt}| + |t| + |St|))$ steps. Since $\mathbf{K}$ is a constant, this reduces to $O(|St_M| \cdot (|\mathbb{A}\mathrm{gt}| + |t| + |St|))$. Thus, the oracle runs in nondeterministic polynomial time with respect to the size of the model. In consequence, the algorithm in Figure 3 runs in time $\mathbf{P}^{\mathbf{NP}} = \boldsymbol{\Delta}_2^{\mathbf{P}}$.

For nested strategic modalities, we proceed recursively, bottom-up, which yields the complexity of $\mathbf{P}^{\boldsymbol{\Delta}_2^{\mathbf{P}}} = \boldsymbol{\Delta}_2^{\mathbf{P}}$ for the whole problem.

Theorem 6.2. *Model checking for* NatATL$_{iR}$ *with fixed or bounded k is in $\boldsymbol{\Delta}_2^{\mathbf{P}}$ w.r.t. the size of the model and the length of the formula.*

6.2 Model Checking: General Case

We now study the model checking complexity for NatATL$_{iR}$ in case the bound on strategies is a parameter of the problem. For variable k, the algorithm in Figures 3 and 4 clearly runs in exponential time. It may also use an exponential amount of memory if we are not careful with how the space of strategies is explored. To avoid this, we slightly change the main procedure, see Figure 5.

The algorithm still runs in exponential time. Observe, however, that the oracle uses only polynomial space: the NFA's have at most $2k$ states altogether, and, by using binary representations of variables *count* and *limit*, we need at most $\log_2(2^{2k^2}) = 2k^2$ memory cells for each of them. Thus, the complexity of the algorithm in Figure 5 is $\mathbf{NP}^{\mathbf{NPSPACE}} = \mathbf{NP}^{\mathbf{PSPACE}} = \mathbf{PSPACE}$.

For nested strategic modalities, we again proceed recursively, which results in the $\mathbf{P}^{\mathbf{PSPACE}} = \mathbf{PSPACE}$ complexity for the whole problem.

Theorem 6.3. *Model checking* NatATL$_{iR}$ *is in* **PSPACE** *with respect to the size of the model, the length of the formula, and the maximal bound k in the formula.*

7 CONCLUSIONS

In this paper, we extend the alternative take on strategic reasoning, proposed in [39], that allows to reason about agents who can handle only relatively simple strategies. We show how to adapt the approach to the important (and nontrivial) case of imperfect information. To this end, we use a natural representation of strategies by lists of actions guarded by epistemic conditions, and assume that only strategies up to size k can be used. We show that such strategies are always executable. Furthermore, we formalize reasoning about the corresponding strategic play through two new variants of alternating-time temporal logic: NatATL$_{ir}$ and NatATL$_{iR}$. We argue that, similarly to perfect information games, this may be a more accurate view of ability than the one which admits any function from sequences of states to actions.

In terms of technical results, we concentrate on model checking for natural strategies under imperfect information. We show that, for memoryless agents, the problem is in **P** when k is fixed, and $\boldsymbol{\Delta}_2^{\mathbf{P}}$-complete when k is among the input parameters. Thus, synthesis and verification of natural memoryless strategies in the context of imperfect information is no more difficult than for perfect information. For agents with recall, the problem is in $\boldsymbol{\Delta}_2^{\mathbf{P}}$ when k is fixed, and in **PSPACE** in the general case, which is still close to the complexity results obtained in [39]. Thus, we ultimately identify a natural subclass of model checking for imperfect information strategies, where the verification is distinctly cheaper than in the general case.

This is certainly good news, and may prove beneficial in practical algorithms. Still, we emphasize that the main motivation for this work is conceptual rather than technical. We believe that the $\langle\langle A\rangle\rangle^{\leq k}$ operator captures an intuitive concept of strategic ability, and one that is useful in modeling of and reasoning about multi-agent interaction.

Acknowledgements. We thank the anonymous reviewers for their comments and suggestions. Wojciech Jamroga acknowledges the support of the National Centre for Research and Development (NCBR), Poland, under the PolLux project VoteVerif (POLLUX-IV/1/2016). Aniello Murano acknowledges the support from the Italian GNCS 2018 project "Metodi formali per la verifica e la sintesi di sistemi discreti e ibridi".

REFERENCES

[1] T. Ågotnes. 2006. Action and Knowledge in Alternating-time Temporal Logic. *Synthese* 149, 2 (2006), 377–409.

[2] T. Ågotnes, V. Goranko, W. Jamroga, and M. Wooldridge. 2015. Knowledge and Ability. In *Handbook of Epistemic Logic*, H.P. van Ditmarsch, J.Y. Halpern, W. van der Hoek, and B.P. Kooi (Eds.). College Publications, 543–589.

[3] T. Ågotnes and D. Walther. 2009. A Logic of Strategic Ability Under Bounded Memory. *Journal of Logic, Language and Information* 18, 1 (2009), 55–77.

[4] N. Alechina, M. Dastani, B. Logan, and J.-J. Ch. Meyer. 2007. A Logic of Agent Programs. In *Proceedings of AAAI*. 795–800.

[5] N. Alechina, B. Logan, M. Dastani, and J.-J. Ch. Meyer. 2008. Reasoning about agent execution strategies. In *Proceedings of International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 1455–1458.

[6] N. Alechina, B. Logan, N.H. Nga, and A. Rakib. 2009. A Logic for Coalitions with Bounded Resources. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*. 659–664.

[7] N. Alechina, B. Logan, H.N. Nguyen, and A. Rakib. 2010. Resource-Bounded Alternating-Time Temporal Logic. In *Proceedings of International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 481–488.

[8] R. Alur, T.A. Henzinger, and O. Kupferman. 2002. Alternating-Time Temporal Logic. *J. ACM* 49, 5 (2002), 672–713.

[9] R. Alur, S. Moarref, and U. Topcu. 2018. Compositional and symbolic synthesis of reactive controllers for multi-agent systems. *Inf. Comput.* 261, Part (2018), 616–633.

[10] M. Barlo, G. Carmona, and H. Sabourian. 2008. Bounded memory with finite action spaces. *Sabanci University, Universidade Nova de Lisboa and University of Cambridge* (2008).

[11] F. Belardinelli, A. Lomuscio, and V. Malvone. 2018. Approximating Perfect Recall When Model Checking Strategic Abilitie. In *KR*. 435–444.

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

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

[14] R. Berthon, B. Maubert, and A. Murano. 2017. Decidability Results for ATL* with Imperfect Information and Perfect Recall. In *AAMAS*. ACM, 1250–1258.

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

[16] D. Berwanger and A. B. Mathew. 2017. Infinite games with finite knowledge gaps. *Inf. Comput.* 254 (2017), 217–237.

[17] R. Bordini, M. Fisher, W. Visser, and M. Wooldridge. 2006. Verifying Multi-Agent Programs by Model Checking. *Autonomous Agents and Multi-Agent Systems* 12, 2 (2006), 239–256.

[18] P. Bouyer, N. Markey, and S. Vester. 2017. Nash equilibria in symmetric graph games with partial observation. *Inf. Comput.* 254 (2017), 238–258.

[19] N. Bulling and B. Farwer. 2010. Expressing Properties of Resource-Bounded Systems: The Logics RTL* and RTL. In *Proceedings of Computational Logic in Multi-Agent Systems (CLIMA) (Lecture Notes in Computer Science)*, Vol. 6214. 22–45.

[20] N. Bulling and B. Farwer. 2010. On the (Un-)Decidability of Model Checking Resource-Bounded Agents. In *Proceedings of ECAI (Frontiers in Artificial Intelligence and Applications)*, Vol. 215. IOS Press, 567–572.

[21] N. Bulling and W. Jamroga. 2014. Comparing variants of strategic ability: how uncertainty and memory influence general properties of games. *Journal of Autonomous Agents and Multi-Agent Systems* 28, 3 (2014), 474–518.

[22] S. Busard, C. Pecheur, H. Qu, and F. Raimondi. 2015. Reasoning about memoryless strategies under partial observability and unconditional fairness constraints. *Inf. Comput.* 242 (2015), 128–156.

[23] P. Cermák, A. Lomuscio, F. Mogavero, and A. Murano. 2018. Practical verification of multi-agent systems against Slk specifications. *Inf. Comput.* 261, Part (2018), 588–614.

[24] K. Chatterjee, L. Doyen, E. Filiot, and J. F. Raskin. 2017. Doomsday equilibria for omega-regular games. *Inf. Comput.* 254 (2017), 296–315.

[25] K. Chatterjee, T.A. Henzinger, and N. Piterman. 2010. Strategy Logic. *Information and Computation* 208, 6 (2010), 677–693.

[26] T. Chen, F. Song, and Z. Wu. 2017. Model Checking Pushdown Epistemic Game Structures. In *Formal Methods and Software Engineering - 19th International Conference on Formal Engineering Methods (ICFEM17) (Lecture Notes in Computer Science)*, Vol. 10610. Springer, 36–53.

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

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

[29] C. Dima and F.L. Tiplea. 2011. *Model-checking ATL under Imperfect Information and Perfect Recall Semantics is Undecidable.* Technical Report. arXiv.

[30] H. Duijf and J.M. Broersen. 2016. Representing Strategies. In *Proceedings of SR*. 15–26. https://doi.org/10.4204/EPTCS.218.2

[31] A. Gupta, S. Schewe, and D. Wojtczak. 2014. Making the best of limited memory in multi-player discounted sum games. *arXiv preprint arXiv:1410.4154* (2014).

[32] J. Gutierrez, G. Perelli, and M. Wooldridge. 2018. Imperfect information in Reactive Modules games. *Inf. Comput.* 261, Part (2018), 650–675.

[33] D. Harel and D. Kozen. 1982. Process Logic: Expressiveness, Decidability, Completeness. *J. Comput. System Sci.* 25, 2 (1982), 144–170.

[34] A. Herzig, E. Lorini, F. Maffre, and D. Walther. 2014. Alternating-time Temporal Logic with Explicit Programs. In *Proceedings of LAMAS*.

[35] J. Hörner and W. Olszewski. 2009. How robust is the Folk Theorem? *The Quarterly Journal of Economics* (2009), 1773–1814.

[36] W. Jamroga and T. Ågotnes. 2007. Constructive knowledge: what agents can achieve under imperfect information. *J. Applied Non-Classical Logics* 17, 4 (2007), 423–475.

[37] W. Jamroga and J. Dix. 2006. Model Checking ATL$_{ir}$ is Indeed Δ_2^P-complete. In *Proceedings of EUMAS (CEUR Workshop Proceedings)*, Vol. 223.

[38] W. Jamroga, M. Knapik, and D. Kurpiewski. 2017. Fixpoint Approximation of Strategic Abilities under Imperfect Information. In *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. IFAAMAS, 1241–1249.

[39] W. Jamroga, V. Malvone, and A. Murano. 2017. Reasoning about Natural Strategic Ability. In *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. IFAAMAS, 714–722.

[40] W. Jamroga and A. Murano. 2015. Module Checking of Strategic Ability. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems AAMAS 2015*. IFAAMAS, 227–235.

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

[42] O. Kupferman and M. Y. Vardi. 1997. Module checking revisited. In *CAV'97*. Springer, 36–47.

[43] O. Kupferman and M. Y. Vardi. 2000. Synthesis with incomplete informatio. In *Advances in Temporal Logic*. Springer, 109–127.

[44] V. Malvone, A. Murano, and L. Sorrentino. 2017. Hiding Actions in Multi-Player Games. In *AAMAS*. 1205–1213.

[45] V. Malvone, A. Murano, and L. Sorrentino. 2018. Additional Winning Strategies in Reachability Games. *Fundam. Inform.* 159, 1-2 (2018), 175–195.

[46] B. Maubert and A. Murano. 2018. Reasoning about Knowledge and Strategies under Hierarchical Information. In *KR*. AAAI Press, 530–540.

[47] F. Mogavero, A. Murano, G. Perelli, and M.Y. Vardi. 2014. Reasoning About Strategies: On the Model-Checking Problem. *ACM Transactions on Computational Logic* 15, 4 (2014), 1–42.

[48] F. Mogavero, A. Murano, G. Perelli, and M.Y. Vardi. 2017. Reasoning about Strategies: on the Satisfiability Problem. *Logical Methods in Computer Science* 13, 1 (2017). https://doi.org/10.23638/LMCS-13(1:9)2017

[49] P. Novák and W. Jamroga. 2009. Code Patterns for Agent Oriented Programming. In *Proceedings of AAMAS'09*. 105–112.

[50] A. Pnueli and R. Rosner. 1989. On the Synthesis of a Reactive Module.. In *POPL'89*. Association for Computing Machinery, 179–190.

[51] A. Pnueli and R. Rosner. 1990. Distributed reactive systems are hard to synthesize. In *FOCS*. 746–757.

[52] J. H. Reif. 1984. The Complexity of Two-Player Games of Incomplete Information. *J. Comput. Syst. Sci.* 29, 2 (1984), 274–301.

[53] P. Y. Schobbens. 2004. Alternating-Time Logic with Imperfect Recall. *Electronic Notes in Theoretical Computer Science* 85, 2 (2004), 82–93.

[54] W. van der Hoek, W. Jamroga, and M. Wooldridge. 2005. A Logic for Strategic Reasoning. In *Proceedings of AAMAS'05*. 157–164.

[55] S. Vester. 2013. Alternating-time temporal logic with finite-memory strategies. In *GandALF 2013*. 194–207.

[56] D. Walther, W. van der Hoek, and M. Wooldridge. 2007. Alternating-time Temporal Logic with Explicit Strategies. In *Proceedings TARK XI*. Presses Universitaires de Louvain, 269–278.

[57] N. Yadav and S. Sardiña. 2012. Reasoning about Agent Programs Using ATL-Like Logics. In *Proceedings of JELIA*. 437–449.