Natural Strategic Ability

Wojciech Jamroga[1], Vadim Malvone[2], Aniello Murano[2]

[1] Institute of Computer Science, Polish Academy of Sciences
[2] Università degli Studi di Napoli Federico II, Italy

Abstract. In game theory, as well as in the semantics of game logics, a strategy can be represented by any function from states of the game to the agent's actions. That makes sense from the mathematical point of view, but not necessarily in the context of human behavior. This is because humans are quite bad at executing complex plans, and also rather unlikely to come up with such plans in the first place. In our work, we adopt the view of bounded rationality, and look only at "simple" strategies in specifications of agents' abilities. We formally define what "simple" means, and propose a variant of alternating-time temporal logic that takes only such strategies into account. We also study the model checking problem for the resulting semantics of ability.

1 Introduction

Logics for strategic reasoning provide powerful tools to reason about multi-agent systems [1, 14, 13, 9, 6, 11]. The logics allow to express properties of agents' behavior and its dynamics, driven by their individual and collective goals. An important factor here is interaction between the agents, which can be cooperative as well as adversarial. Specifications in agent logics can be then used as input to *model checking* [7, 12], which makes it possible to verify the correct behavior of a multi-agent system using recently developed practical automatic tools [10, 4, 5].

A fundamental contribution in this field is *Alternating-Time Temporal Logic (*ATL^**)* and its fragment ATL [1]. ATL^* formulas are usually interpreted over *concurrent game structures (CGS)* which are labeled state-transition systems that model synchronous interaction among agents. For example, given a CGS modeling a system with k agents and a shared resource, the ATL formula $\langle\langle A\rangle\rangle \mathsf{F}\mathsf{grant}$ expresses the fact that the set of agents A can ensure that, regardless of the actions of the other agents, an access to the resource will be eventually granted. The specification holds if agents in A have a collective strategy whose every execution path satisfies $\mathsf{F}\mathsf{grant}$. As in game theory, strategies 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 [6, 11]) 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. That makes sense from a mathematical point of view, and also in case we think of strategic ability of a machine (robot, computer program). We claim, however, that the approach is not very realistic 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 has to come up with the strategy on its own.

2 Our Contribution

In [8], we adopt the view of bounded rationality, and look only at strategies whose complexity does not exceed a given bound. In this way we put a limit on the resources needed to represent and use the strategy. More precisely, we introduce NatATL*, a logic that extends 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, we assume 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.

In terms of technical results, we show that model-checking for NatATL with memoryless strategies is in $\mathbf{P}$ when k is fixed, and $\mathbf{\Delta_2^P}$-complete when k is a parameter of the problem. For strategies with recall, the problem is in $\mathbf{\Delta_2^P}$ when k is fixed, and in $\mathbf{\Delta_3^P}$ in the general case, cf. the summary presented in Figure 1.

Clearly, reasoning about *simple* natural memoryless strategies is no more difficult than about arbitrary ATL strategies (and in practice we expect it to be actually easier). On the other hand, verification of natural strategies with recall seems distinctly harder. It would be interesting to look for conditions under which the latter kind of strategies can be synthesized in polynomial time.

We also prove an important property that sets NatATL apart from standard ATL: in NatATL, the memoryless and memoryfull semantics do not coincide. Moreover, we show that our representation of memoryfull strategies is equally expressive, and strictly more succinct, than deterministic finite-state transducers from [15]. As a consequence, model checking natural strategic ability in unbounded strategies with recall is undecidable.

3 Future work

In the future, we plan to extend the framework to natural strategies with imperfect information. We would also like to extend our results to the broader language of NatATL*, and refine them in terms of parameterized complexity. Another interesting path concerns a graded version of the logic with counting how many successful natural strategies are available. We also intend to look at other natural expressions of strategies, including a survey of psychological studies suggesting how people plan and execute their long-term behaviors. Finally, a more complete account of bounded rationality may be obtained by combining bounds on conceptual complexity of strategies (in the spirit of our work here) with their temporal complexity via timing constraints in the vein of [3, 2].

	memoryless	finite recall
ATL	**P**-complete	**P**-complete
1NatATL, fixed k	in **P**	in $\mathbf{\Delta_2^P}$
NatATL, fixed k	in **P**	in $\mathbf{\Delta_2^P}$
1NatATL, variable k	**NP**-complete	in $\mathbf{\Sigma_2^P}$
NatATL, variable k	$\mathbf{\Delta_2^P}$-complete	in $\mathbf{\Delta_3^P}$

Fig. 1. Summary of model checking complexity results

References

1. R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time Temporal Logic. *Journal of the ACM*, 49:672–713, 2002.
2. É. André, L. Petrucci, W. Jamroga, M. Knapik, and W. Penczek. Timed ATL: forget memory, just count. In *AAMAS 2017*, pages 1460–1462, 2017.
3. T. Brihaye, F. Laroussinie, N. Markey, and G. Oreiby. Timed concurrent game structures. In *CONCUR 2007*, pages 445–459, 2007.
4. P. Čermák, A. Lomuscio, F. Mogavero, and A. Murano. MCMAS-SLK: A Model Checker for the Verification of Strategy Logic Specifications. In *CAV'14*, LNCS 8559, pages 524–531. Springer, 2014.
5. P. Čermák, A. Lomuscio, and A. Murano. Verifying and Synthesising Multi-Agent Systems against One-Goal Strategy Logic Specifications. In *AAAI 2015*, pages 2038–2044. AAAI Press, 2015.
6. K. Chatterjee, T. Henzinger, and N. Piterman. Strategy Logic. *Information and Computation*, 208(6):677–693, 2010.
7. E. Clarke and E. Emerson. Design and Synthesis of Synchronization Skeletons Using Branching-Time Temporal Logic. In *LP'81*, LNCS 131, pages 52–71. Springer, 1981.
8. W. Jamroga, V. Malvone, and A. Murano. Reasoning about natural strategic ability. In *AAMAS 2017*, pages 714–722, 2017.
9. W. Jamroga and W. van der Hoek. Agents that know how to play. *Fundamenta Informaticae*, 63(2–3):185–219, 2004.
10. A. Lomuscio, H. Qu, and F. Raimondi. MCMAS: A Model Checker for the Verification of Multi-Agent Systems. In *CAV'09*, LNCS 5643, pages 682–688. Springer, 2009.
11. F. Mogavero, A. Murano, G. Perelli, and M. Vardi. Reasoning About Strategies: On the Model-Checking Problem. *ACM Transactions on Computational Logic*, 15(4):34:1–42, 2014.
12. J. Queille and J. Sifakis. Specification and Verification of Concurrent Programs in Cesar. In *International Symposium on Programming'81*, LNCS 137, pages 337–351. Springer, 1981.
13. P. Y. Schobbens. Alternating-time logic with imperfect recall. *Electronic Notes in Theoretical Computer Science*, 85(2):82–93, 2004.
14. W. van der Hoek and M. Wooldridge. Tractable Multiagent Planning for Epistemic Goals. In *AAMAS 02*, pages 1167–1174, 2002.
15. S. Vester. Alternating-time temporal logic with finite-memory strategies. In *GandALF 2013*, pages 194–207, 2013.