



Alechina, Natasha and Logan, Brian (2016) Verifying systems of resource-bounded agents. Lecture Notes in Computer Science, 9709 . pp. 3-12. ISSN 0302-9743

Access from the University of Nottingham repository:

<http://eprints.nottingham.ac.uk/34430/1/cie16.pdf>

Copyright and reuse:

The Nottingham ePrints service makes this work by researchers of the University of Nottingham available open access under the following conditions.

This article is made available under the University of Nottingham End User licence and may be reused according to the conditions of the licence. For more details see:
http://eprints.nottingham.ac.uk/end_user_agreement.pdf

A note on versions:

The version presented here may differ from the published version or from the version of record. If you wish to cite this item you are advised to consult the publisher's version. Please see the repository url above for details on accessing the published version and note that access may require a subscription.

For more information, please contact eprints@nottingham.ac.uk

Verifying Systems of Resource-Bounded Agents

Natasha Alechina¹ and Brian Logan²

¹ University of Nottingham
Nottingham, UK

nza@cs.nott.ac.uk

² University of Nottingham
Nottingham, UK

bsl@cs.nott.ac.uk

Abstract. Approaches to the verification of multi-agent systems are typically based on games or transition systems defined in terms of states and actions. However such approaches often ignore a key aspect of multi-agent systems, namely that the agents' actions require (and sometimes produce) resources. We briefly survey previous work on the verification of multi-agent systems that takes resources into account, and outline some key challenges for future work.

1 Verifying Autonomous Systems

A multi-agent system (MAS) is a system that is composed of multiple interacting agents. An agent is an *autonomous* entity that has the ability to collect information, reason about it, and perform actions based on it in pursuit of its own goals or on behalf of others. Examples of agents are controllers for satellites, non-driver transport systems such as UAVs, health care systems, and even nodes in sensor networks.

Multi-agent systems are ubiquitous. Many distributed software and hardware systems can be naturally modelled as multi-agent systems. Such systems are by the nature of their components extremely complex, and the interaction between components and their environment can lead to undesired behaviours that are difficult to predict in advance. With the increasing use of autonomous agents in safety critical systems, there is a growing need to *verify* that their behaviour conforms to the desired system specification, and over the last decade verification of multi-agent systems has become a thriving research area [24].

A key approach to the verification of MAS is *model checking*. Model checking involves checking whether a model of the system satisfies a temporal logic formula corresponding to some aspect of the system specification. Model checking has the advantage that it is a fully automated technique, which facilitates its use in the MAS development process.³ A wide range of approaches to model-checking MAS have been proposed in the literature, ranging from the adaptation of standard model-checking tools, e.g., [12, 13] to the development of special-purpose model checkers for multi-agent systems, e.g., [27, 22].

³ Another strand of work focusses on theorem proving, e.g., [28], but such approaches typically require user interaction to guide the search for a proof.

2 Resource-Bounded Agents

In many multi-agent systems, agents are *resource-bounded*, in the sense that they require resources in order to act. Actions require time to complete and typically require additional resources depending on the application domain, for example energy or money. For many applications, the availability or otherwise of resources is critical to the properties we want to verify: a multi-agent system will have very different behaviours depending on the resource endowment of the agents that comprise it. For example, an agent with insufficient energy may be unable to complete a task in the time assumed by a team plan, if it has to recharge its battery before performing the task.

However, with a few exceptions which we discuss below, previous work on verification of MAS abstracts away from the fact that many multi-agent systems consist of agents that need resources to operate and that those resources are limited. In particular, current state-of-the-art verification techniques and tools for MAS are unable to verify system properties that depend on the resource production and consumption of the agents comprising the MAS.

In this paper we survey recent work in the emerging field of verification of resource-bounded agents, and highlight a number of challenges that must be overcome to allow practical verification of resource-bounded MAS. We argue that recent work on the complexity of model-checking for logics of strategic ability with resources offers the possibility of significant progress in the field, new verification approaches and tools, and the ability to verify the properties of a large, important class of autonomous system that were previously out of reach.

3 Model-Checking with Resources

In this section we give a brief introduction to model-checking multi-agent systems and explain how standard model checking approaches can be extended with resources.

In a model-checking approach to the verification of multi-agent systems, a MAS is represented by a finite state transition system.⁴ A state transition system consists of a set of states and transitions between them. Intuitively, each state of a MAS corresponds to a tuple of states of the agents and of the environment, and each transition corresponds to actions performed by the agents. Each state is labelled with atomic propositions that are true in that state. A standard assumption is that each state in the system has at least one outgoing transition (if a state is a deadlock state in the original MAS, we can model this by adding a transition to itself by some null action and labelling it with a ‘deadlock’ proposition). Properties of the system to be verified are expressed in an appropriate temporal logic L . The *model-checking problem for L* is, given a state transition system M (and possibly a state s) and an L formula ϕ , check whether ϕ is true in M (at state s).

For multi-agent systems, a temporal logic of particular interest is Alternating Time Temporal Logic (ATL) [9]. ATL generalises other temporal logics such as Computation Tree Logic (CTL) [19] (which can be seen as a one-agent ATL) by introducing notions

⁴ There is work on model-checking infinite state transition systems, see, for example, [11], but in this paper we concentrate on the finite case.

of strategic ability. ATL is interpreted over concurrent game structures (transition systems where edges correspond to a tuple of actions performed simultaneously by all the agents, see the example below). The language of ATL contains atomic propositions, boolean connectives \neg, \wedge , etc. and modalities $\langle\langle A \rangle\rangle\bigcirc, \langle\langle A \rangle\rangle\Box$ and $\langle\langle A \rangle\rangle\mathcal{U}$ for each subset (or coalition in ATL terms) A of the set of all agents, which express the strategic ability of the coalition A . $\langle\langle A \rangle\rangle\bigcirc\phi$ means that the coalition of agents A has a choice of actions such that, regardless of what the other agents in the system do, ϕ will hold in the next state. $\langle\langle A \rangle\rangle\Box\phi$ means that coalition A has a strategy to keep ϕ true forever, regardless of what the other agents do. A strategy is a choice of actions which either only depends on the current state (memoryless strategy) or on the finite history of the current state (perfect recall strategy). Finally, $\langle\langle A \rangle\rangle\phi\mathcal{U}\psi$ means that A has a strategy to ensure that after finitely many steps ψ holds, and in all the states before that, ϕ holds. The model-checking problem for ATL can be solved in time polynomial in the size of the transition system and the property [9], and there exist model-checking tools for ATL, for example, MOCHA [10] and MCMAS [27].

Example Figure 1 illustrates a simple ATL model of a system with two agents, 1 and 2, and actions α, β, γ and *idle*. Action tuples on the edges show the actions of each agent, for example, in the transition from state s_I to s , agent 1 performs action α and agent 2 performs *idle*. In this system, in state s_I , agent 1 has a (memoryless) strategy to enforce that p holds eventually in the future no matter what agent 2 does, which can be expressed in ATL as $\langle\langle\{1\}\rangle\rangle\top\mathcal{U}p$. Similarly, in s_I agent 1 has a memoryless strategy to keep $\neg p$ true forever, so $\langle\langle\{1\}\rangle\rangle\Box\neg p$ holds in s_I .

3.1 Adding Resources

In order to model multi-agent systems where agents' actions produce and consume resources, it is necessary to modify the approach above in two ways. One is to add resource annotations to the actions in the transition system: for each individual action and each resource type, we need to specify how many units of this resource type the action produces or consumes. For example, suppose that there are two resource types, r_1 and r_2 (e.g., energy and money). Then we can specify that action α in Figure 1 produces two units of r_1 and consumes one unit of r_2 , action β consumes one unit of r_1 and produces one unit of r_2 , action γ consumes five units of r_1 , and action *idle* does not produce or consume any resources.

The second modification is to extend the temporal logic so that we can express properties related to resources. For example, we may want to express a property that a group of agents A can eventually reach a state satisfying ϕ or can maintain the truth of ψ forever, provided that they have available n_1 units of resource type r_1 and n_2 units of resource type r_2 . Such statements about coalitional ability under resource bounds can be expressed in an extension of ATL where coalitional modalities are annotated with a resource bound on the strategies available to the coalition. We call logics where every action is associated with produced and consumed resources and the syntax reflects resource requirements on agents, *resource logics*.

To illustrate the properties resource logics allow us to express, consider the model in Figure 1 with the production and consumption of resources by actions specified above.

In this setting, we can verify if agent 1 can eventually enforce p provided that it has one unit of r_2 in state s_I , or whether the coalition of agents $\{1, 2\}$ can achieve p under this resource bound by working together. There are surprisingly many different ways of measuring costs of strategies and deciding which actions are executable by the agents given the resources available to them, but under at least one possible semantics, the answer to the first question is no and to the second one yes, but the latter requires a perfect recall strategy (the two agents should loop between states s_I and s until they produce a sufficient amount of resource r_1 , and then execute actions corresponding to the $\langle \gamma, idle \rangle$ transition from s to s').

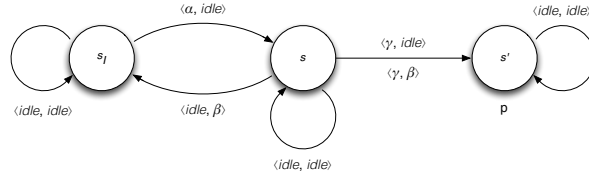


Fig. 1: State transition system.

Clearly, the model-checking problem for temporal logics is a special case of the model-checking problem for the corresponding resource logics. The question is, how much harder does the model-checking problem become when resources are added?

4 A Brief Survey of Resource Logics

In this section, we briefly review recent theoretical work on the development of resource logics. We focus on expressiveness and model-checking complexity, as these features determine the suitability of a particular logic for practical verification.

4.1 Consumption of Resources

Early work on resource logics considered only consumption of resources (no action produced resources), and initial results were encouraging.

One of the first logics capable of expressing resource requirements of agents was a version of Coalition Logic (CL)⁵, called Resource-Bounded Coalition Logic (RBCL), where actions only consume (and don't produce) resources. It was introduced in [1] with the primary motivation of modelling systems of resource-bounded reasoners, however the framework is sufficiently general to model any kind of action. The model-checking problem for this logic was shown to be decidable in [5] in polynomial time in the transition system and the property and exponential in the number of resource types.

⁵ CL is a fragment of ATL with only the next time $\langle\langle A \rangle\rangle \bigcirc$ modality.

A resource-bounded version of ATL, RB-ATL, where again actions only consume (and not produce) resources was introduced in [2]. It was also shown that the model-checking problem for this logic is decidable in time polynomial in the size of the transition system and exponential in the number of resource types. (For a single resource type, e.g., energy, the model-checking problem is no harder than for ATL.)

Practical work on model-checking standard computer science transition systems (not multi-agent systems) with resources also falls in the category of consumption-only systems, for example probabilistic model-checking of systems with numerical resources as done using PRISM model-checker [26] assumes costs monotonically increasing with time.

4.2 Adding Production

However, when resource production is considered in addition to consumption, the situation changes. In a separate strand of work, a range of different formalisms for reasoning about resources was introduced in [16, 14]. In those formalisms, both consumption and production of resources was considered. In [15] it was shown that the problem of halting on empty input for two-counter automata [25] can be reduced to the model-checking problem for several of their resource logics. Since the halting problem for two-counter automata is undecidable, the model-checking problem for a variety of resource logic with production of resources is undecidable. The reduction uses two resource types (to represent the values of the two counters) and either one or two agents depending on the version of the logic (whether the agents have perfect recall, whether the formula talking about coalition A can also specify resource availability for remaining agents, and whether nested operators ‘remember’ initial allocation of resources or can be evaluated independently of such initial allocation).

The only decidable cases considered in [14] are an extension of CTL with resources (essentially one-agent ATL) and a version where on every path only a fixed finite amount of resources can be produced. In [14], the models satisfying this property are called bounded, and it is pointed out that RBCL and RB-ATL are logics over a special kind of bounded models (where no resources are produced at all). Other decidability results for bounded resource logics have also been reported in the literature. For example, [20] define a decidable logic, PRB-ATL (Priced Resource-Bounded ATL), where the total amount of resources in the system has a fixed bound. The model-checking algorithm for PRB-ATL requires time polynomial in the size of the model and exponential in the number of resource types and the resource bound on the system. In [21] an EXP-TIME lower bound in the number of resource types for the PRB-ATL model-checking problem is shown.

A general logic over systems with numerical constraints called QATL* was introduced in [17]. In that paper, more undecidability results for the model-checking problem of QATL* and its fragments were shown. For example, QATL (Quantitative ATL) is undecidable even if no nestings of cooperation modalities is allowed. The main proposals for restoring decidability to the model-checking problem for QATL in [17] are removing negative payoffs (similar to removing resource production) and also introducing memoryless strategies. Shared resources were considered in [18]; most of the

cases considered there have undecidable model-checking (apart from the case of a single shared resource, which has decidable model-checking).

This brief survey of work to date suggest that the main approach until recently to dealing with both resource production and consumption was to bound the amount of produced resources globally in the model. For some systems of resource-bounded agents, this is a reasonable restriction. For example, agents that need energy to function and are able to charge their battery, can never ‘produce’ more energy than the capacity of their battery. This is a typical bounded system. However, in some cases, although every single application of the agent’s actions produces a fixed amount of some resource, repeating this action arbitrarily often will produce arbitrarily large amounts of the resource. This may apply to energy stored in unbounded storage, or to money, or many other natural situations. Recent work suggests that verification of such systems may still be possible.

5 Decidable Unbounded Production

In [6] a version of ATL, $\text{RB}\pm\text{ATL}$, was introduced where actions both produce and consume resources. The models of the logic do not impose bounds on the overall production of resources, and the agents have perfect recall. The syntax of $\text{RB}\pm\text{ATL}$ is very similar to that of ATL, but coalition modalities have superscripts which represent resource allocation to agents. Instead of stating the existence of *some* strategy, they state the existence of a strategy such that every computation generated by following this strategy consumes at most the given amount of resources. Coming back to the example, the property that agent 1 can eventually enforce p provided that it has one unit of r_2 can be expressed as $\langle\langle\{1\}^{(0,1)}\rangle\rangle\top\mathcal{U}p$ ⁶ The model-checking problem for $\text{RB}\pm\text{ATL}$ is decidable ($\text{RB}\pm\text{ATL}$ is very similar to one of the resource logics introduced in [14] for which the decidability of the model-checking problem was left open). The existence of a decidable resource logic with unbounded production was surprising, as it was the first indication that it is possible to automatically verify properties of this important class of resource-bounded multi-agent system.

However, although this result is encouraging, we are not yet at the point of practical verification of such systems. In [6] the lower bound on the complexity of the model-checking problem for $\text{RB}\pm\text{ATL}$ is shown to be EXPSPACE. The proof of EXPSPACE-hardness is by reduction of the reachability problem for Petri Nets to the model-checking problem for $\text{RB}\pm\text{ATL}$. Although the Petri Net reachability problem is decidable, the upper bound on its complexity is still unknown; similarly we do not know the upper bound on the $\text{RB}\pm\text{ATL}$ complexity. The complexity of the model-checking problem $\text{RB}\pm\text{ATL}$ is thus much higher than that for ATL without resources and the consumption-only resource logics surveyed above. This high complexity makes it difficult to develop practical verification approaches. The only exception is $1\text{-RB}\pm\text{ATL}$, $\text{RB}\pm\text{ATL}$ with a single resource type, where the complexity is PSPACE.

⁶ Here, $(0, 1)$ is the allocation of 0 units of r_1 and 1 unit of r_2 to agent 1. We only show resource bound for the proponent agents, $\{1\}$ in this case. Versions of resource logic where opponents are also resource-bounded all have an undecidable model-checking problem, see [14].

In [3], a new syntactic fragment FRAL of resource logic RAL with decidable model-checking has been identified. It restricts the occurrences of coalitional modalities on the left of the Until formulas; on the other hand, it allows nested modalities to refer to resource allocation at the time of evaluation, rather than always consider a fresh resource allocation, as in $\text{RB}\pm\text{ATL}$. More precisely, a formula $\langle\langle A^b \rangle\rangle \phi \mathcal{U} \langle\langle A^\downarrow \rangle\rangle \psi_1 \mathcal{U} \psi_2$ says that given resource allocation b , coalition A can always reach a state (maintaining ϕ) where with the remaining resources, it can reach ψ_2 while maintaining ψ_1 . The complexity of model-checking for this fragment is also open, and is also likely to be high.

Although model-checking of ATL with perfect recall and uniform strategies is undecidable, replacing uniformity with a weaker notion, for example defining it using distributed knowledge, is decidable [23]. Similar results hold for $\text{RB}\pm\text{ATL}$ with syntactic epistemic knowledge and weaker notions of uniformity, $\text{RB}\pm\text{ATSEL}$ [4].

Below is a summary of resource logics with decidable model-checking problem. In all of them, the semantics assume that in every state each agent has an available action to do nothing, which produces and consumes no resources.

Logic	Resource production	Complexity of model-checking
RBCL	no	EXPTIME (PTIME in model)
RB-ATL	no	EXPTIME (PTIME in model)
PRB-ATL	bounded	EXPTIME
$\text{RB}\pm\text{ATL}$	yes	EXPSPACE-hard
1- $\text{RB}\pm\text{ATL}$	yes	PSPACE
FRAL	yes	?
$\text{RB}\pm\text{ATSEL}$	yes	EXPSPACE-hard

Table 1: Resource logics with decidable model-checking problem

6 Future Challenges

The $\text{RB}\pm\text{ATL}$ results offer the possibility of significant progress in the verification of resource-bounded multi-agent systems. However many challenges remain for future research. Below we list three of the most important.

Understanding sources of undecidability Developing a better understanding of the sources of decidability and undecidability (beyond boundedness) will be critical to future progress. As observed in [14], subtle differences in truth conditions for resource logics result in the difference between decidability and undecidability of the model-checking problem. Preliminary work in this direction is reported in [3].

Lower complexity It is useful to discover sources of undecidability and how to construct expressive logics for which the model-checking problem is decidable. However, it is even more important to be able to develop logics, or fragments of existing logics such as $RB\pm ATL$, that are sufficiently expressive for practical problems, but where the model-checking problem has tractable complexity (ideally polynomial in the size of the transition system, as in the case of bounded production logics). Only then would we be able to implement practical model-checking tools for systems of resource-bounded agents.

Practical tools Although model checking algorithms have been proposed for several of the logics surveyed, work on implementation is only beginning. We aim to develop practical model-checking tools for verifying resource-bounded MAS by extending the MCMAS model checker [27] to allow the modelling of multi-agent systems in which agents can both consume and produce resources. Work on symbolic encoding of $RB\pm ATL$ model-checking is reported in [8] and work on symbolic encoding of $RB\pm ATL$ model-checking are reported in [7].

Addressing these challenges will allow practical model-checking of resource logics and constitute a major break-through in multi-agent system verification.

References

1. Alechina, N., Logan, B., Nguyen, H.N., Rakib, A.: A logic for coalitions with bounded resources. In: Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI 2009). vol. 2, pp. 659–664. IJCAI/AAAI, AAAI Press (2009)
2. Alechina, N., Logan, B., Nguyen, H.N., Rakib, A.: Resource-bounded alternating-time temporal logic. In: Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010). pp. 481–488. IFAAMAS (2010)
3. Alechina, N., Bulling, N., Logan, B., Nguyen, H.N.: On the boundary of (un)decidability: Decidable model-checking for a fragment of resource agent logic. In: Yang, Q. (ed.) Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI 2015). IJCAI, AAAI Press, Buenos Aires, Argentina (July 2015)
4. Alechina, N., Dastani, M., Logan, B.: Verifying existence of resource-bounded coalition uniform strategies. In: Rossi, F. (ed.) IJCAI 2016, Proceedings of the 25th International Joint Conference on Artificial Intelligence. IJCAI/AAAI (2016)
5. Alechina, N., Logan, B., Nga, N.H., Rakib, A.: Logic for coalitions with bounded resources. *Journal of Logic and Computation* 21(6), 907–937 (December 2011)
6. Alechina, N., Logan, B., Nguyen, H.N., Raimondi, F.: Decidable model-checking for a resource logic with production of resources. In: Proceedings of the 21st European Conference on Artificial Intelligence (ECAI-2014). pp. 9–14. ECCAI, IOS Press, Prague, Czech Republic (August 2014)
7. Alechina, N., Logan, B., Nguyen, H.N., Raimondi, F.: Symbolic model-checking for one-resource $RB\pm ATL$. In: Yang, Q. (ed.) Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI 2015). IJCAI, AAAI Press, Buenos Aires, Argentina (July 2015)

8. Alechina, N., Logan, B., Nguyen, H.N., Raimondi, F., Mostarda, L.: Symbolic model-checking for resource-bounded ATL. In: Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015. pp. 1809–1810. ACM (2015)
9. Alur, R., Henzinger, T., Kupferman, O.: Alternating-time temporal logic. *Journal of the ACM* 49(5), 672–713 (2002)
10. Alur, R., Henzinger, T.A., Mang, F.Y.C., Qadeer, S., Rajamani, S.K., Tasiran, S.: MOCHA: Modularity in model checking. In: Computer Aided Verification. pp. 521–525 (1998)
11. Belardinelli, F.: Verification of non-uniform and unbounded artifact-centric systems: decidability through abstraction. In: Bazzan, A.L.C., Huhns, M.N., Lomuscio, A., Scerri, P. (eds.) International conference on Autonomous Agents and Multi-Agent Systems, AAMAS '14. pp. 717–724. IFAAMAS/ACM (2014)
12. Bordini, R.H., Fisher, M., Visser, W., Wooldridge, M.: Model checking rational agents. *IEEE Intelligent Systems* 19(5), 46–52 (2004)
13. Bordini, R.H., Fisher, M., Visser, W., Wooldridge, M.: Verifying multi-agent programs by model checking. *Journal of Autonomous Agents and Multi-Agent Systems* 12(2), 239–256 (March 2006), <http://dro.dur.ac.uk/622/>
14. Bulling, N., Farwer, B.: On the (un-)decidability of model checking resource-bounded agents. In: Proceedings of the 19th European Conference on Artificial Intelligence (ECAI 2010). Frontiers in Artificial Intelligence and Applications, vol. 215, pp. 567–572. IOS Press (2010)
15. Bulling, N., Farwer, B.: On the (un-)decidability of model checking resource-bounded agents. Tech. Rep. IfI-10-05, Clausthal University of Technology (2010)
16. Bulling, N., Farwer, B.: Expressing properties of resource-bounded systems: The logics RBTL and RBTL*. In: Post-Proceedings of CLIMA '09. pp. 22–45. LNCS 6214 (2010)
17. Bulling, N., Goranko, V.: How to be both rich and happy: Combining quantitative and qualitative strategic reasoning about multi-player games (extended abstract). In: Mogavero, F., Murano, A., Vardi, M.Y. (eds.) Proceedings 1st International Workshop on Strategic Reasoning, SR 2013. EPTCS, vol. 112, pp. 33–41 (2013)
18. Bulling, N., Nguyen, H.N.: Model checking resource bounded systems with shared resources via alternating büchi pushdown systems. In: Chen, Q., Torroni, P., Villata, S., Hsu, J.Y., Omicini, A. (eds.) PRIMA 2015: Principles and Practice of Multi-Agent Systems - 18th International Conference. Lecture Notes in Computer Science, vol. 9387, pp. 640–649. Springer (2015), <http://dx.doi.org/10.1007/978-3-319-25524-8>
19. Clarke, E.M., Emerson, E.A., Sistla, A.P.: Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems* 8(2), 244–263 (1986)
20. Della Monica, D., Napoli, M., Parente, M.: On a logic for coalitional games with priced-resource agents. *Electr. Notes Theor. Comput. Sci.* 278, 215–228 (2011)
21. Della Monica, D., Napoli, M., Parente, M.: Model checking coalitional games in shortage resource scenarios. In: Proceedings of the 4th International Symposium on Games, Automata, Logics and Formal Verification (GandALF 2013. EPTCS, vol. 119, pp. 240–255 (2013)
22. Dennis, L.A., Fisher, M., Webster, M.P., Bordini, R.H.: Model checking agent programming languages. *Automated Software Engineering* 19(1), 5–63 (2012)
23. Dima, C., Tiplea, F.L.: Model-checking ATL under imperfect information and perfect recall semantics is undecidable. *CoRR abs/1102.4225* (2011), <http://arxiv.org/abs/1102.4225>
24. Fisher, M., Dennis, L.A., Webster, M.P.: Verifying autonomous systems. *Commun. ACM* 56(9), 84–93 (2013)
25. Hopcroft, J.E., Ullman, J.D.: Introduction to Automata Theory, Languages and Computation. Addison-Wesley (1979)

26. Kwiatkowska, M.Z., Norman, G., Parker, D.: Prism 4.0: Verification of probabilistic real-time systems. In: Proceedings of 23rd International Conference on Computer Aided Verification (CAV 2011). Lecture Notes in Computer Science, vol. 6806, pp. 585–591. Springer (2011)
27. Lomuscio, A., Qu, H., Raimondi, F.: MCMAS: A model checker for the verification of multi-agent systems. In: Bouajjani, A., Maler, O. (eds.) Proceedings of the 21st International Conference on Computer Aided Verification (CAV 2009). Lecture Notes in Computer Science, vol. 5643, pp. 682–688. Springer (2009)
28. Shapiro, S., Lespérance, Y., Levesque, H.J.: The cognitive agents specification language and verification environment for multiagent systems. In: Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'02). pp. 19–26. ACM Press, New York, NY, USA (2002)