Logic-based Specification and Verification of Homogeneous Dynamic Multi-agent Systems

JAAMAS Track

Riccardo De Masellis
Uppsala University, Sweden
riccardo.demasellis@it.uu.se

Valentin Goranko
Stockholm University, Sweden
valentin.goranko@philisophy.su.se

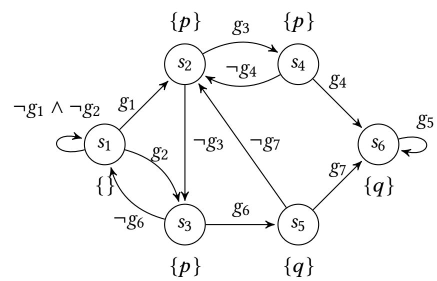
ACM Reference Format:
Riccardo De Masellis and Valentin Goranko. 2021. Logic-based Specification and Verification of Homogeneous Dynamic Multi-agent Systems: JAAMAS Track. In *Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), Online, May 3–7, 2021*, IFAAMAS, 3 pages.

1 INTRODUCTION

We consider discrete concurrent multi-agent transition systems (cf. [1] or [4]) which are *homogeneous* and *dynamic*. The *homogeneity* means that all agents are essentially indistinguishable from each other, as they have the same available actions at each state and the effect of these actions depends not on *which* agents perform them, but only on *how many* agents perform each action. Thus, the state transitions are determined only by the vector of numbers of agents performing each action and are specified symbolically, by means of conditions on these numbers definable in Presburger arithmetic [8]. The only distinction between the agents in such homogeneous systems is whether they are *controllable* (by the system supervisor/controller) or *uncontrollable*, representing the environment or adversary. The *dynamicity* of the systems that we consider means that the set (hence, the number) of agents being present (or, just active) in the system may vary throughout the system evolution, possibly at every transition from a state to a state. Typical examples of homogeneous and dynamic systems include: voting procedures [10, 13], sensor networks [15] and markets. We model the dynamicity by assuming that there is an unbounded (and possibly infinite) set of 'potentially existing' agents, but that only finitely many of them are 'actually existing/present' at each stage of the evolution of the system. We note, however, the difference between such dynamic systems and simply *parametric* systems, where the number of agents is taken as a parameter but remains fixed during the whole evolution of the system.

We develop abstract models of homogeneous dynamic multi-agent systems (HDMAS), illustrated on Fig. 1 (explained further), and a logic-based framework for formal specification and algorithmic verification for them. We assume that the agents are divided into *controllable* (by the system supervisor or controller) and *uncontrollable*, representing the environment or an adversary. Both numbers, of controllable and uncontrollable agents, may be fixed or varying throughout the system evolution, possibly at every transition. As a logical language for formal specification we introduce a suitably extended version, $\mathcal{L}_{\text{HDMAS}}$, of the Alternating Time temporal Logic ATL [1] where the strategic operator takes two parameters: the numbers (fixed or variable) of controllable and of uncontrollable agents. We then present an algorithm for model checking $\mathcal{L}_{\text{HDMAS}}$ on HDMAS and give worst-case complexity estimates.

Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), U. Endriss, A. Nowé, F. Dignum, A. Lomuscio (eds.), May 3–7, 2021, Online.



$$g_1 := (x_1 \geq 2x_2) \wedge (x_3 \leq 3)$$
$$g_2 := (x_1 + x_2 + x_3 \leq 10) \wedge (x_3 > 3)$$
$$g_3 := (x_1 > 5) \wedge (x_3 > x_1)$$
$$g_4 := x_1 > 5 \wedge (3x_2 < x_1 + 2x_3)$$
$$g_5 := x_1 = x_1;$$
$$g_6 := x_1 + 2x_2 \geq x_3$$
$$g_7 := x_2 = x_3;$$

Figure 1: An abstract example of a HDMAS.

Related work. Our framework shares with Open Multi-Agent Systems (OMAS) [12] the characteristic 'dynamic' feature of agents, which can leave and join the system at runtime, however the two frameworks differ both in models and specification languages.

Parametric systems [3, 7, 11], games for counting abstraction [14], Modular Interpreted Systems [9], Homogeneous Systems [13] and Population Protocols [2, 6] are also related, although they do not consider the same kind of dynamicity: in the work above, the number of agents is fixed along system executions, possibly as a parameter and the formal specification languages do not explicitly allow quantification over the number of agents. Also, the language $\mathcal{L}_{\text{HDMAS}}$ is original for our logical framework.

2 MODELLING FRAMEWORK

The formal definition of HDMAS models can be found in the full paper [5]. Here we illustrate them with the example in Figure 1, where the circles represent states of the system, labeled with sets of atomic propositions that are true in them, and arrows are transitions, labeled by Presburger arithmetic formulas called *guards*, which determine the transitions depending on the number of agents performing each possible action. For instance guard g_1 states that transition from s_1 to s_2 is performed when the number of agents performing action 1 (represented by the variable x_1) is greater or equal twice the number of agents performing action 2 (variable x_2) and the number of agents performing action 3 (variable x_3) is less than or equal 3. The set of agents is possibly infinite and the *action availability function* (not shown on the figure) assigns to each state the set of actions that are available to all agents at that state. A special "empty" action ε, which has no impact and is not mentioned in the guards, is always available. Also, the guards are defined so that for every state and every tuple of numbers of agents performing the actions available in that specific state, there is exactly one guard that is satisfied by that tuple, so that the system can always progress, in a deterministic way.

3 SPECIFICATION LANGUAGE AND LOGIC

The logical language $\mathcal{L}_{\text{HDMAS}}$ used for specifying and verifying properties of HDMAS is based on the Alternating-time Temporal Logic ATL. However, the strategic operator $\langle\langle C, N\rangle\rangle$ employed in $\mathcal{L}_{\text{HDMAS}}$ takes two arguments: C represents the number of controllable agents and N represents the number of uncontrollable agents currently present in the system. Each of the arguments C and N may be a concrete number or a variable that can be quantified over.

We now fix a set of atomic propositions $\Phi = \{p_1, p_2, ...\}$ and a set of two special variables $Y = \{y_1, y_2\}$ ranging over $\mathbb{N}$ and representing the numbers of controllable and uncontrollable agents respectively. Then we define the set of terms $T = Y \cup \mathbb{N}$ which will be used as arguments of the strategic operators.

The logic $\mathcal{L}_{\text{HDMAS}}$ has two sorts of formulae, defined by mutual induction by the following formal grammars, where free (and bound) occurrences of variables are defined like in first-order logic: *path formulae*:

$$\chi ::= \mathsf{X}\,\varphi \mid \mathsf{G}\,\varphi \mid \psi\,\mathsf{U}\,\varphi$$

where φ, ψ are *state formulae*, defined by:

$$\varphi ::= \top \mid p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid \langle\langle t_1, t_2\rangle\rangle\,\chi \mid \forall y\varphi \mid \exists y\varphi$$

where $p \in \Phi$, $t_1 \in T \setminus \{y_2\}$, $t_2 \in T \setminus \{y_1\}$, $y \in Y$, and χ is a path formula. The cases of $\forall y\varphi$ and $\exists y\varphi$ are subject to the following syntactic constraint: all free occurrences of y in φ must have a positive polarity, viz. must be in the scope of an even number of negations. The propositional connectives $\perp, \rightarrow, \leftrightarrow$ are defined as usual, as well as $\mathsf{F}\,\psi := \top\,\mathsf{U}\,\psi$.

The semantics of the temporal operators is like in LTL and ATL, and the semantics of the quantifiers is like in first-order logic. Intuitively, the formula $\langle\langle t_1, t_2\rangle\rangle\,\chi$ says that "*a coalition of (at least) t_1 controllable agents can ensure against (at most) t_2 uncontrollable agents that any possible evolution of the system satisfies the objective χ*", For the formal semantics, we refer to the full paper [5].

As an example, the closed formula $\varphi = \langle\langle 7, 5\rangle\rangle\,\mathsf{X}\,p$ is satisfied in state s_1 of $\mathcal{M}$ in Figure 1. Indeed, any joint strategy that prescribes ε to 3 of the controllable agents and act_3 to 4 of them guarantees that guard g_2 is satisfied, enforcing transition from s_1 to s_3.

Some remarks on the formulae in $\mathcal{L}_{\text{HDMAS}}$:

(1) y_1 can only occur in the first position of $\langle\langle t_1, t_2\rangle\rangle$ and y_2 can only occur in the second position;

(2) by virtue of its semantics, the strategic operator is *monotonic* with respect to the number of controllable agents and *anti-monotonic* with respect to the number of uncontrollable agents, i.e. if $\langle\langle t_1, t_2\rangle\rangle\,\chi$ holds in a HDMAS model, then $\langle\langle t'_1, t'_2\rangle\rangle\,\chi$ also holds for every $t'_1 \geq t_1$ and $t'_2 \leq t_2$.

4 TRANSFORMATION TO NORMAL FORM

Using these properties, we prove that on *finite* models (viz., with a finite number of states) every formula in $\mathcal{L}_{\text{HDMAS}}$ is equivalent to one in *normal form*, in the fragment $\mathcal{L}^{\mathsf{NF}}_{\text{HDMAS}}$, where the formulae are defined as before except that the clauses for $\forall y\varphi$ and $\exists y\varphi$ are replaced with the following (where χ is a temporal objective)

$$\exists y_1\langle\langle y_1, t_2\rangle\rangle\chi \mid \forall y_2\exists y_1\langle\langle y_1, y_2\rangle\rangle\chi \mid \forall y_2\langle\langle t_1, y_2\rangle\rangle\chi \mid \exists y_1\forall y_2\langle\langle y_1, y_2\rangle\rangle\chi$$

In addition, in each case above no variable quantified in the prefix of the formula may occur free in χ. The equivalence between the two languages is of crucial importance, as our model checking algorithm applies only to $\mathcal{L}^{\mathsf{NF}}_{\text{HDMAS}}$ formulae. The restriction that quantification in formulae in normal form does not span across multiple temporal objectives enables us to obtain fixpoint characterizations for formulae of the types listed above. That, in turn, allows us to retain the basic structure of the recursive model checking algorithm for ATL (cf. [1] or [4]). A linear-time procedure to transform any formula in $\mathcal{L}_{\text{HDMAS}}$ to one in $\mathcal{L}^{\mathsf{NF}}_{\text{HDMAS}}$ is presented in the full paper [5]. Essentially, it pushes the quantifications in front of temporal operators and eliminates some of them by using monotonicity-based equivalences such as (among others) $\forall y_1\langle\langle y_1, t\rangle\rangle\,\chi \equiv \langle\langle 0, t\rangle\rangle\,\chi[0/y_1]$ and $\exists y_2\langle\langle t, y_2\rangle\rangle\,\chi \equiv \langle\langle t, 0\rangle\rangle\,\chi[0/y_2]$ (where $\chi[t_1/t_2]$ is the formula obtained by substituting all occurrences of t_2 with t_1 in χ) and by using standard boolean transformations. For example, the normal form of $\forall y_1\big(\langle\langle y_1, 5\rangle\rangle\,(\forall y_2\langle\langle y_1, y_2\rangle\rangle\,\mathsf{X}\,p_1)\ \mathsf{U}\ (\exists y_2\langle\langle y_1, y_2\rangle\rangle\,\mathsf{F}\,p_2)\big)$ is $\langle\langle 0, 5\rangle\rangle\,\big((\forall y_2\langle\langle 0, y_2\rangle\rangle\,\mathsf{X}\,p_1)\ \ \mathsf{U}\ \ (\forall y_2\langle\langle 0, y_2\rangle\rangle\,\mathsf{F}\,p_2)\big)$

5 SYMBOLIC MODEL CHECKING OF $\mathcal{L}^{\mathsf{NF}}_{\textbf{HDMAS}}$

The (global) model checking algorithm for $\mathcal{L}^{\mathsf{NF}}_{\text{HDMAS}}$ is very similar in structure to that for global model checking of ATL, but is performed symbolically, by involving translation to Presburger arithmetic (PrA). It is an iterative procedures of computing controllable pre-images that the fixpoint characterizations of the temporal operators G and U yield. The main difference is in the computation of the controllable pre-images of a set of states, which is now based on checking the truth of formulae of Presburger arithmetic that describe the effect of the guards on the transitions in the model. Therefore, the complexity of the (global) model checking problem for $\mathcal{L}^{\mathsf{NF}}_{\text{HDMAS}}$ is bounded above by the complexity of checking the truth of a PrA-formula, which in turns depends not just on its size, but more precisely on the numbers of quantifier alternations and of quantified variables in any quantifier block (cf. [8]). For the full language $\mathcal{L}^{\mathsf{NF}}_{\text{HDMAS}}$ the worst-case complexity is Σ^{EXP}_3, but when the number of either controllable or uncontrollable agents is fixed or bounded, it drops to NP-complete if the number of actions is unbounded, resp. P-complete if that number is fixed or bounded.

ACKNOWLEDGMENTS

This work was supported by research grant 2015-04388 of the Swedish Research Council.

REFERENCES

[1] Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman. 2002. Alternating-Time Temporal Logic. *J. ACM* 49, 5 (2002), 672–713.

[2] Dana Angluin, James Aspnes, Zoë Diamadi, Michael J. Fischer, and René Peralta. 2004. Computation in networks of passively mobile finite-state sensors. In *Proceedings of the Twenty-Third ACM PODC St. John's, Canada.* 290–299.

[3] Roderick Bloem, Swen Jacobs, Ayrat Khalimov, Igor Konnov, Sasha Rubin, Helmut Veith, and Josef Widder. 2016. Decidability in Parameterized Verification. *SIGACT News* 47, 2 (2016), 53–64.

[4] Nils Bulling, Valentin Goranko, and Wojciech Jamroga. 2015. Logics for Reasoning About Strategic Abilities in Multi-player Games. In *Models of Strategic Reasoning: Logics, Games, and Communities*, J. van Benthem, S. Ghosh, and R. Verbrugge (Eds.). Springer Berlin Heidelberg, 93–136.

[5] Riccardo De Masellis and Valentin Goranko. 2020. Logic-based specification and verification of homogeneous dynamic multi-agent systems. *Autonomous Agents and Multi-Agent Systems* 34, 2 (2020), 34. https://doi.org/10.1007/s10458-020-09457-8

[6] Javier Esparza, Pierre Ganty, Jérôme Leroux, and Rupak Majumdar. 2016. Model Checking Population Protocols. In *Proc. of 36th IARCS Annual Conference on FSTTCS, Chennai, India.* 27:1–27:14.

[7] Steven M. German and A. Prasad Sistla. 1992. Reasoning about Systems with Many Processes. *J. ACM* 39, 3 (1992), 675–735.

[8] Christoph Haase. 2018. A survival guide to Presburger Arithmetic. *SIGLOG News* 5, 3 (2018), 67–82.

[9] Wojciech Jamroga and Thomas Ågotnes. 2007. Modular Interpreted Systems. In *Proceedings of AAMAS.* ACM, 131:1–131:8.

[10] Wojciech Jamroga, Michal Knapik, and Damian Kurpiewski. 2018. Model Checking the SELENE E-Voting Protocol in Multi-agent Logics. In *Proceedings of the Third International Joint Conference on Electronic Voting, E-Vote-ID 2018 (Lecture Notes in Computer Science, Vol. 11143).* Springer, 100–116.

[11] Panagiotis Kouvaros and Alessio Lomuscio. 2016. Parameterised verification for multi-agent systems. *Artificial Intelligence* 234 (2016), 152–189.

[12] Panagiotis Kouvaros, Alessio Lomuscio, Edoardo Pirovano, and Hashan Punchihewa. 2019. Formal Verification of Open Multi-Agent Systems. In *Proc. of AAMAS '19.* 179–187.

[13] Truls Pedersen and Sjur Kristoffer Dyrkolbotn. 2013. Agents Homogeneous: A Procedurally Anonymous Semantics Characterizing the Homogeneous Fragment of ATL. In *Proc. of PRIMA 2013.* 245–259.

[14] Jean-François Raskin, Mathias Samuelides, and Laurent Van Begin. 2005. Games for Counting Abstractions. *Electr. Notes Theor. Comput. Sci.* 128, 6 (2005), 69–85.

[15] Meritxell Vinyals, Juan A. Rodriguez-Aguilar, and Jesus Cerquides. 2011. A Survey on Sensor Networks from a Multiagent Perspective. *The Computer Journal* 54, 3 (March 2011), 455–470. https://doi.org/10.1093/comjnl/bxq018