On a Logic for Coalitional Games with Priced-Resource Agents

Dario Della Monica[1], Margherita Napoli[1], and Mimmo Parente[1]

Dept. of Computer Science, Universitá di Salerno, Fisciano (Italy),
{ddellamonica|napoli|parente}@unisa.it

Abstract. *Alternating-time Temporal Logic* (ATL) and *Coalition Logic* (CL) are well-established logical formalisms particularly suitable to model games between dynamic coalitions of agents (like e.g. the system and the environment). Recently, the ATL formalism has been extended in order to take into account boundedness of the resources needed for a task to be performed. The resulting logic is known as *Resource-bounded* ATL (RB-ATL) and has been presented in quite a variety of scenarios. Model checking RB-ATL in very general setting is usually undecidable. Nevertheless, model checking procedures for semantically or syntactically restricted versions of RB-ATL have been proposed. In this paper, we analyze the problem of coalitions of agents that need to perform complex tasks, by using resources with a variable price. We highlight a certain number of problems and considerations, based on different interpretations of shortage of resources, leading to different scenarios.

1 Introduction

Automated verification of multi-agent systems is a significant topic in the recent literature in artificial intelligence [1]. The need of modeling this kind of systems has inspired logical formalisms, the most famous being the *Alternating-time Temporal Logics* [4] and the *Coalition Logic* (CL) [7,8], oriented towards the description of collective behaviors.

The idea of such logics is that agents can join together in team (or coalitions) and share resources to accomplish a task (or reach a goal). In particular, Alternating-time Temporal Logics have been introduced in [4], where the full alternating-time temporal language, denoted by ATL^*, has been presented, along with two significant fragments, namely, ATL and ATL^+.

In [6], Goranko has studied the relationship between the (expressive power of the) two formalisms. In particular, he has shown that CL can be embedded into ATL. Anyway, none of the two logics takes into account the boundedness of the resources. Approaches towards verification of multi-agent systems under resource constraints can be found in [2,3,5]. In [2], Alechina et al. introduce the logic RBCL, whose language extends the one of CL with explicit representation of resource bounds. In [3], the same authors propose an analogous extension for ATL, called RB-ATL, and give a PTIME model checking procedure mostly based on the one for ATL. In [5], Bulling and Farwer introduce the logics RAL and RAL^*. The former represents a generalization of Alechina et al.'s RB-ATL, the latter is ATL^* extended with resource bounds. The authors study several syntactic and semantic variants of RAL and RAL^* with respect to the (un)decidability of the model checking problem. In particular, while previous approaches only conceive actions consuming resources, they introduce the notion of actions producing resources. It turned out that such a new notion makes the model checking problem undecidable.

The paper is structured as follows. In the next section, we make some considerations about scenarios proposed in the literature, we illustrate our proposals, and show that the model checking problem is still decidable. Then, in the last section, we propose new scenarios for which the model checking problem is under study.

2 Our scenario

This section contains an epistemic discussion about the formalization of a multi-agent system in which agents can cooperate to perform a task and are subject to a limited availability of resources, that is an intrinsic feature of any real-world system. Our discussion hinges on existing approaches in the literature (see e.g. [2,3,5]) and represents an attempt to do a further step towards the formalization of such complex systems.

Formulas of the formalisms proposed in [2,3,5] allow one to assign an endowment of resources to the agents by means of the so-called *team operators*, (borrowed from ATL) and to state that a team of agents can perform a task. Due to the nesting of the team operators in a formula (which reflects the fact that coalitions may change in a game), during the execution of the task, the agents can be provided with a new endowment of resources to perform subtasks. This is somehow unrealistic, as it does not take into account issues related to procurement of resources. In particular, a very significant present-day issue is that resources are available on the market (or in nature) in limited amount, and the cost for achieving them depends on such an availability.

First improvement. Thus, our first proposal is to introduce the notion of *price* of resources. Unlike the existing approaches, agents are equipped with an amount of money instead of an endowment of resources. They can use money for getting resources. Notice that money cannot be considered as a resource like the others for at least two reasons. First, according to our aim of assigning, to each resource, a price that is variable depending on several factors (e.g., the current availability of the resource on the market), it is necessary to introduce the new component money with the special ability of "measuring" the value of all the resources, thus making it possible for the agent to acquire them when needed. Second, since the money as the special ability of "measuring" the value of the resources makes sense to consider problems of optimization (e.g., minimization of the amount of money needed to acquire the resources to perform a task). Formulas of our logic state that a team of agents is able to perform a given task provided with a given amount of money. We also introduce a notion of *global availability* of resources on the market, the intended meaning being that, whenever an agent acquires resources from the market, the global availability is decreased, whenever it produces resources, the global availability is increased. The price of resources can be any function of the several components into play. In our approach, prices of resources depend on their global availability, the acting agent, and the physical location.

Second improvement. Another aspect that has not been fully analyzed in the literature is the problem of actions producing resources. On the one hand, in [2,3], actions can only consume resources; on the other hand, in [5], the authors state that whenever actions can produce resources the model checking problem is undecidable. In this paper, we show how to constrain the way in which actions can produce resources, still preserving the decidability of the model checking problem. The idea is that it is possible, at a given time, for an action to produce a resource in a quantity that is not greater than the amount that has already been consumed so far. This implies that, even if actions can produce resources, the global availability of the market will never be greater than the initial global availability, that is crucial for the model checking algorithm. Such a notion makes sense as, in practical terms, it allows one to model significant real-world scenarios, such as, acquiring memory by a program, leasing a car during a travel, and, in general, any scenario in which an agent is releasing resources previously acquired.

2.1 Team and task

So far, we have talked about teams (or coalitions) of agents performing a task. But we have not clarified yet the two notions of team and task. First of all, a task is a goal that has to be reached and, for what concerns us, is represented by a logical formula that has to be satisfied. A team of agents is a subset of agents that act collectively in order to perform a task. To this end, they select a strategy that univocally determines their behavior in each possible configuration of the system. Nevertheless, the behavior of the remaining agents, that we collectively denote as the *opponent*, is undetermined. Aim of the team is to guarantee that the task is performed independently of the opponent's behavior, that is, the task must be guaranteed for each possible opponent's strategy.

The formalism that naturally fits our intention is the logic ATL, that allows one to fix a strategy for the agents of a team and to force an 'LTL-like' property, representing the task, to be true over all the possible executions (or outcomes) of the system. Obviously, its syntax and semantics will be extended in order to deal with resource constraints.

2.2 The special resource 'time'

One can be interested in answering questions of the kind "is it possible for the team A of agents to complete the task in x time-unit?". It is clear that the resource 'time' neither can be bought nor rented. It is in a certain sense out of the control of the agents, as it is only possible to specify that a task should be executed within some given time constraints, while it is not possible to administer it. Thus, resource 'time' will be treated in a special way with respect to other resources.

2.3 Game structure

A *game structure* G is based on a graph whose vertices, called *locations*, are labeled by atomic propositions. In each location, each agent can choose among a non-empty set of actions to be performed. Any possible combination of actions gives rise to *transitions*, that are the edges of the graph. In general, actions consume and produce resources. Each resource has a price that is variable and depends on, inter alia, the current availability of that resource on the market. Thus, a transition can be executed if the resources needed to perform the actions are available and each agent has enough money to acquire them.

Let $\mathbb{Z}$ denote the set of integers, $\mathbb{N}$ denote the set of non-negative integers, and let $\mathcal{M} = (\mathbb{N} \cup \{\infty\})^r$. A game structure G is a tuple $\langle Q, \mathcal{AP}, V, Ag, \Sigma, \Delta, R, t, c, \rho \rangle$, where:

- Q is the finite set of *locations*, $\mathcal{AP}$ is the finite set of *propositional letters*, and $V : Q \to 2^{\mathcal{AP}}$ is the *valuation function*;
- $Ag = \{a_1, a_2, \ldots, a_n\}$ is the finite set of *agents*, and Σ is the finite set of *actions*, denoted by $\alpha_1, \alpha_2, \ldots$,
- $\Delta : Q \times Ag \to 2^{\Sigma}$ is the *action function* that defines the possible actions of an agent in a given location. By an abuse of notation, we use Δ also to denote the function from Q to 2^{Σ^n} defined as $\Delta(q) = \Delta(q, a_1) \times \ldots \times \Delta(q, a_n)$;
- $R = \{R_1, \ldots, R_r\}$ is the finite set of *resource types.* It contains the particular resource *time*, denoted by R_1;
- $t : Q \times \Sigma^n \to Q$ is the *transition function* over the set of locations. It is a partial function defined for any pair $(q, \langle \alpha_1, \ldots, \alpha_n \rangle)$ such that $\langle \alpha_1, \ldots, \alpha_n \rangle \in \Delta(q)$;
- $c : \Sigma \times Ag \to \mathbb{Z}^r$ is the *cost* function, assigning a cost to each action performed by an agent. A negative cost represents a resource consumption, while a positive cost represents a resource production;
- $\rho : \mathcal{M} \times Ag \times Q \to \mathbb{N}^r$ is the *price function*, denoting the price of each resource, depending on the current resource availability, the acting agent, and the current location. Without loss of generality, we can assume the price of the resource 'time' to be always zero, as it is a resource that cannot be acquired and thus its price is meaningless.

2.4 A logical formalization: PRB-ATL

We now define the logic *Priced* RB-ATL (PRB-ATL). The formulae are given by the following grammar.

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\langle A^{\$}\rangle\rangle \bigcirc \varphi \mid \langle\langle A^{\$}\rangle\rangle \Box\varphi \mid \langle\langle A^{\$}\rangle\rangle \varphi\mathcal{U}\varphi \mid \sim \vec{b}$$

where $p \in \mathcal{AP}$, $A \subseteq Ag$, $\sim \in \{<, \leq, =, \geq, >\}$, and $\vec{b} \in \mathcal{M}$. Moreover, $\$ \in (\mathbb{N} \cup \{\infty\})^n$ is a vector representing the availability of money of the agents. For each agent $a \in Ag$, by $\$_a$ we denote the availability of money for the agent a. Intuitively, formulae of the kind $\sim \vec{b}$ tests the current availability of resources on the market. Formulae of the kind $\langle\langle A^{\$}\rangle\rangle\psi$, with $\psi \in \{\bigcirc\varphi, \Box\varphi, \varphi\mathcal{U}\varphi\}$ state that the team A has a strategy such that, for every action performed by the opponent (i.e, $Ag \setminus A$), ψ is satisfied, and such that the total expenses of each agent $a \in A$ is less than or equal to $\$_a$. Without loss of generality, we can assume $\$_a = \infty$ for each $a \notin A$.

In order to give the formal semantics we must first define the following notions. From now on, let G be a generic game structure. We extend the sum operation to sum between vectors component-wise. Additionally, we use the usual component-wise comparison relations between vectors.

Definition 1 (configuration and computation). *A* configuration *of G is a pair $c = \langle q, \vec{m}\rangle \in Q \times \mathcal{M}$. A* finite computation *(resp.,* infinite computation*) over G is a finite (resp., infinite) sequence of configurations (of G) $\pi = c_1 c_2 \ldots$, such that, for each i, if $c_i = \langle q_i, \vec{m_i}\rangle$ and $c_{i+1} = \langle q_{i+1}, \vec{m_{i+1}}\rangle$, then there exists a transition $t(q_i, \vec{\alpha}) = q_{i+1}$, with $\vec{\alpha} = \langle \alpha_1, \ldots, \alpha_n\rangle$, such that $\vec{m_{i+1}} = \vec{m_i} + \sum_{j=1}^{n} c(\alpha_j, a_j)$.*

Given a team A, a *move* of A, denoted $\vec{\alpha_A}$, is a vector of actions α_a, for all $a \in A$, representing the action performed by the agents of A. To represent the possible moves of the team A at any location q, we extend the function Δ with the function $\hat{\Delta}_A : Q \to 2^{\Sigma^{|A|}}$ representing the Cartesian product of $\Delta(q, a)$, for all $a \in A$.

Definition 2 (strategy). *A* strategy *F_A for the team of agents A is a function which associates, to each finite computation $\pi = c_1 c_2 \ldots c_s$, with $c_s = \langle q, \vec{m}\rangle$, a move $\vec{\alpha_A}$, such that $\vec{\alpha_A} \in \hat{\Delta}_A(q)$. A strategy is said to be* memoryless *if $F_A(\pi) = F_A(\pi')$ for each pair of computations π, π' such that $\pi = c_1 c_2 \ldots c_s$, $\pi' = c'_1 c'_2 \ldots c'_{s'}$, $c_s = \langle q, \vec{m}\rangle$, $c'_s = \langle q, \vec{m'}\rangle$.*

In other words, a strategy F_A determines the behavior of the agents in the team A. Anyway, for each move $\vec{\alpha_A}$ (of the team A) and location $q \in Q$, depending on the move of the opponent, there are several possibilities for the next location. The set including all such possibilities is called the set of *outcomes of the move $\vec{\alpha_A}$ (of the team A) at location q*, denoted by $out(q, \vec{\alpha_A})$. As a consequence, given an initial location q_1, a strategy F_A corresponds to a tree of computations, called *outcomes of the strategy F_A from the location q_1* and denoted by $out(q_1, F_A)$.

Finally, in order to prevent actions producing resources to cause a reimbursement of money to the agent, we define $cons : \Sigma \times Ag \to \mathbb{Z}^r$ in such a way that $cons(\alpha, a)$ returns the vector obtained from the vector $c(\alpha, a)$ by replacing the positive components with zeros and the negative components with the corresponding absolute value.

Let $\pi = c_1 c_2 \ldots \in out(q_0, F_A)$, where $c_i = \langle q_i, \vec{m_i}\rangle$ for all i, be a computation and let $\vec{\alpha}^i = \langle \alpha_a^i \rangle_{a \in Ag}$ be the move performed by the agents at the configuration c_i for all i.

Definition 3 (consistent computations). *The computation π is $(\$, \vec{m_1})$-consistent for a strategy F_A if, for each $i \geq 0$, $\vec{0} \leq \vec{m_i} \leq \vec{m_1}$, and $a \in A$*

$$\sum_{j=1}^{i} \rho(\vec{m}_j, a, q_j) \cdot cons(\alpha_a^j, a) \leq \$_a.$$

The semantics of the logic can be defined as usual and we omit it here.

2.5 Model checking

The model checking problem consists in verifying whether a formula φ is satisfied in a location q of a game structure G, with an initial resource availability $\vec{m} \in \mathcal{M}$.

The algorithm for model checking our logic is mostly based on the one proposed in [4] and used in [3] for model checking, respectively, ATL and its resource-bounded extension RB-ATL. Roughly speaking, it works by computing, for each sub-formula ψ of the formula φ to be model checked, the set of states in which ψ holds. The main difficulties when dealing with bounds on resources are the following. First, the set of sub-formulae must be replaced by an extended set of formulae (see [3]), that includes, for each sub-formula of the form $\langle\langle A^{\$}\rangle\rangle\psi$, all the formulae $\langle\langle A^{\$'}\rangle\rangle\psi$ for each $\$' < \$$. Second, the state does not correspond anymore to the vertices of the game structure, but to configurations, that is, pairs $\langle q, \vec{m}\rangle \in Q \times \mathcal{M}$. Third, during the analysis of the computations over the game structure, the algorithm must take into account the resource availability on the market in order to guarantee that in each instant of the computation all the resources are still available, as well as to be able to compute the current prices of resources, that depend also on their availability. Finally, it must be ensured that, even if actions can produce resources, availability of each resource may not be higher than the initial availability.

Thus, we now state the main result, without showing the proof in this preliminary version. Full details of both the formalization and the algorithm will appear in a forthcoming paper.

Let M be the greater component appearing in the initial resource availability vector $\vec{m}$.

Theorem 1. *The model checking problem for* PRB-ATL *is decidable in time* $O(M^r \times |\varphi|^{r+1} \times |G|)$.

3 Discussion

A further line of research in which we intend to investigate is when, given a formula in our logic, the coalitions are unknown, that is they are not specified and we may ask whether, for each nested sub-formula, there exists a team and a money endowment such that the formula is satisfied. More precisely, given a formula Ψ where $\langle\langle X_i^{\$_i}\rangle\rangle$ are the team operators occurring in it, we want to compute the coalitions X_i such that Ψ is satisfied with minimum expense in terms of both money and resources. Let us notice that if the minimality condition is not requested, then the problem can be trivially solved.

Another feature we are investigating is when each agent has a price. In this scenario, in which agents are themselves resources to be acquired to perform the task, it makes sense to consider the problem of deciding which team is able to perform the task at the minimum cost.

References

1. Thomas Ågotnes, Wiebe van der Hoek, and Michael Wooldridge. On the logic of coalitional games. In *AAMAS*, pages 153–160, 2006.
2. Natasha Alechina, Brian Logan, Nguyen Hoang Nga, and Abdur Rakib. A logic for coalitions with bounded resources. In *Proc. of the 21st International Joint Conference on Artificial Intelligence*, IJCAI '09, pages 659–664, 2009.
3. Natasha Alechina, Brian Logan, Nguyen Hoang Nga, and Abdur Rakib. Resource-bounded alternating-time temporal logic. In *Proc. of the 9th International Conference on Autonomous Agents and Multiagent Systems: Volume 1*, AAMAS '10, pages 481–488, 2010.
4. Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman. Alternating-time temporal logic. *Journal of ACM*, 49:672–713, September 2002.
5. Nils Bulling and Berndt Farwer. On the (un-)decidability of model checking resource-bounded agents. In *Proc. of the 19th European Conference on Artificial Intelligence*, ECAI '10, pages 567–572, 2010.

6. Valentin Goranko. Coalition games and alternating temporal logics. In *Proc. of the 8th Conference on Theoretical Aspects of Rationality and Knowledge*, TARK '01, pages 259–272, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.
7. Marc Pauly. A logical framework for coalitional effectivity in dynamic procedures. *Bulletin of Economic Research*, 53(4):305–324, 2001.
8. Marc Pauly. A modal logic for coalitional power in games. *Journal of Logic and Computation*, 12(1):149–166, 2002.