Available online at www.sciencedirect.com

SciVerse ScienceDirect

Electronic Notes in Theoretical Computer Science 278 (2011) 215–228

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

On a Logic for Coalitional Games with Priced-Resource Agents

Dario Della Monica [1] Margherita Napoli [2] Mimmo Parente[3]

Dipartimento di Informatica
Università di Salerno
Italy

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, called *Resource-Bounded* ATL (RB-ATL), has been presented in quite a variety of scenarios. Even if the model checking problem for extensions of ATL dealing with resource bounds is usually undecidable, a model checking procedure for RB-ATL has been proposed. In this paper, we introduce a new formalism, called PRB-ATL, based on a different notion of resource bounds and we show that its model checking problem remains in EXPTIME and has a PSPACE lower bound.
Then, we tackle the problem of coalition formation. How and why agents should aggregate is not a new issue and has been deeply investigated, in past and recent years, in various frameworks, as for example in algorithmic game theory, argumentation settings, and logic-based knowledge representation. We face this problem in the setting of priced resource-bounded agents with the goal specified by an ATL formula. In particular we solve the problem of determining the minimal cost coalitions of agents acting in accordance to rules expressed by a priced game arena and satisfying a given formula. We show that such problem is computationally not harder than verifying the satisfaction of the same formula with *fixed* coalitions.

Keywords: multi-agent systems, coalition logics, bounded resources, model checking, coalition formations

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) [10,11], oriented towards the description of collective behaviors.

[1] Email: ddellamonica@unisa.it

[2] Email: napoli@dia.unisa.it

[3] Email: parente@unisa.it

doi:10.1016/j.entcs.2011.10.017

The idea of such logics is that agents can join together in teams (or coalitions) and share resources to accomplish a task (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^+. These logics are natural specification languages for open system, that is, systems whose behavior depends on the interactions with an external entity, usually called the *environment*.

In [9], 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. Recently these two logics have been used for the verification of multi-agent systems (MAS), where the agents are equipped with a limited amount of resources to reach their goal [2,3,6,7] (more on this in the *Related works* section below).

The framework we present here hinges on these approaches and represents a further step towards the formalization of such complex systems: multi-agent systems in which agents can cooperate to perform a task and are subject to a limited availability of resources, that is an intrinsic feature of most real-world systems. In particular formulae of the formalisms proposed in [2,3,6,7] allow one to assign an endowment of resources to the agents by means of the so-called *team operators* (borrowed from ATL). The problem is then to determine whether the agents in the proponent team have a strategy to carry out the assigned goals with that bounded amount of resources, whatever the agents in the opponent team do. Anyway, the treatment of this boundedness presents some weakness, as we will point out below.

Based on the natural observation that, in order to acquire a resource, there is a price to be paid, usually depending also on the availability of the resource on the market, we propose to consider bounded resources that have each a price to be paid by the agents for their use in reaching the goal. Thus differently from the existing approaches, agents are equipped with an amount of money instead of an endowment of resources. Money is in a sense a meta-resource. On one hand, its introduction is essential to model the natural scenario in which acquiring the resources needed to perform the task, has a price that depends on several factors: on their global availability, on the acting agent, and on the current system state. On the other hand, money has the peculiarity of "measuring" the value of all the resources, thus, it makes sense to consider problems of optimization (e.g., minimization of the amount of money needed to acquire the resources to perform a task).

In the previous approaches the notion of boundedness of resources is somehow weak, in the sense that resource bounds only appear in the formulae and are applied solely to the proponent team, but they are not represented inside the model at all. This means that it is possible to ask whether a team can reach a goal with a given amount of resources, but it is not possible to keep trace of the evolution of the availability of resources in the world (in particular, the resource consumption due to the actions of the opponent is not controlled). For example, consider the formula $\langle\langle A^b \rangle\rangle \Box p$, belonging to the formalism proposed in [3]. Its intuitive semantics is that the team A can guarantee that p always holds, independently from the behavior of the opponent $(\mathcal{AG} \setminus A)$, using an amount of resources bounded by b. A model for this formula must contain a loop where the joint actions of agents in the team A do not consume resources, but the joint actions of agents in the opponent team may possibly consume resources, leading to an unlimited consumption of resources. In our opinion, such a behavior is not realistic.

We introduce hence a notion of *global availability* of resources on the market (or in nature) that evolves depending on both proponent and opponent behaviors. Such resources are shared, in the sense that all the agents draw on resources from a shared pool and acquisition of a resource by an agent (independently if the agent belong to the proponent or opponent team) implies that the resources will be available in smaller quantity.

The notion of money used here presents several similarity with the notion of resources used in [3]. Indeed, here money is given to the agents to perform a task (like resources are given to the agents in [3]). Moreover, the consumption of money of the opponent is not controlled (like resource consumption of the opponent in [3]). Money, unlike the other resources, can thus be thought of as a private (non-shared) resource. Additionally, opponent has unlimited economic power, in the sense that opponent's agents are supposed to have enough money to acquire all resources they need (this reflects the choice to not limit the opponent power, as it is usual in game theory, to look for robust strategies of the proponent). Roughly speaking, the opponent can buy everything, except for resources that do not exist anymore.

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 [7], the authors state that whenever actions can produce resources the model checking problem is undecidable. It can be easily argued that the undecidability comes from the unboundedness production of resources, thus we naturally constrain the way in which actions can produce resources: it is possible for an action to produce a resource in a quantity that is not greater than the amount that has already been consumed so far. 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.

Finally, we also tackle the problem of coalition formation. How and why agents should aggregate is not a new issue and has been deeply investigated, in past and recent years, in various frameworks, as for example in algorithmic game theory, argumentation settings, and logic-based knowledge representation (see [8,5]). We face this problem in the setting of priced resource-bounded agents with the goal specified by an ATL formula. In particular we solve the problem of determining the minimal cost coalitions of agents acting in accordance to rules expressed by a priced game arena and satisfying a given formula.

We show that both the model checking problem and the optimal coalition problem are in EXPTIME and have a PSPACE lower bound.

Related works. 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 model checking procedure that runs in time $O(|\varphi|^{2\cdot r+1} \times S)$, where φ is the formula to be checked, S is the model, and r is the number of resources. Thus, if the number of resources is treated as constant, the model checking problem for RB-ATL is in PTIME. However, the problem of determining a lower bound to the model checking problem and, in particular, whether a PTIME algorithm exists even if the number of resources is not treated as a constant factor is left open.

In [7], 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 bounded resources. 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 formalize our logic to consider priced resources, then in section 3 we analyze the complexity of the model checking problem and in section 4 we deal with the problem of finding optimal coalitions. Last, we conclude with some considerations on future research directions.

2 A logical formalization: PRB-ATL

In this section we define the logic *Priced* RB-ATL (PRB-ATL). We start with the introduction of some notations we will use in the rest of the paper. The set of *agents* is $\mathcal{AG} = \{1, 2, \ldots, n\}$ and a *team* is any subset of $\mathcal{AG}$. The set of *resources types*, called also simply *resources*, is $R = \{R_1, R_2, \ldots, R_r\}$, where R_1 represents the resource 'time'. The numbers n and r will be used through the paper to denote the number of agents and resources, respectively. Let $\mathcal{M} = (\mathbb{N} \cup \{\infty\})^r$ denote the set of *global availabilities of resources on the market (or in nature)* and $\mathcal{N} = (\mathbb{N} \cup \{\infty\})^n$ denote the set of *money availabilities for the agents*, where $\mathbb{N}$ is the set of non-negative integers. Given a money availability $\$ \in \mathcal{N}$, by $\$[a]$ we denote the money availability for the agent a, for each $a \in \mathcal{AG}$. Finally, the set Π is a finite set of *atomic propositions*.

The formulae of PRB-ATL are given by the following grammar:

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

where $p \in \Pi$, $A \subseteq \mathcal{AG}$, $\sim \in \{<, \leq, =, \geq, >\}$, $\boldsymbol{b} \in \mathcal{M}$ and $\$ \in \mathcal{N}$. Intuitively, formulae of the kind $\sim \boldsymbol{b}$ test the current availability of resources on the market.

Formulae of the kind $\langle\langle A^{\$}\rangle\rangle\psi$, with $\psi \in \{\bigcirc\varphi, \varphi\mathcal{U}\varphi, \Box\varphi\}$, state that the team A has a strategy such that, for every action performed by the opponent (i.e, $\mathcal{AG} \setminus A$), 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 $a \notin A$.

Formulae of this logic are evaluated with respect to (a location of) a *priced game structure* and an initial availability of resources. Intuitively, a *priced game structure* G is 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 the agents of a team have enough money to acquire them.

We extend the sum operation to sum between vectors component-wise. Additionally, we use the usual component-wise comparison relations between vectors and denote by $\mathbf{0^k}$ the vector of k 0s. When the size of the vector is clear, we will simply write $\mathbf{0}$. Finally, the behavior of ∞ with respect to the sum operation is defined as usual, that is, $\infty + c = c + \infty = \infty + \infty = \infty$, for each constant c.

Formally, priced game structures are defined extending the definitions of concurrent game structure and resource-bounded concurrent game structure given in, respectively, [4] and [3].

Definition 2.1 [priced game structure] A *priced game structure* G is defined as a tuple $\langle Q, \pi, d, qty, \delta, \rho, \boldsymbol{m_0} \rangle$, where:

- Q is the finite set of *locations.*
- $\pi : Q \to 2^{\Pi}$ is the *valuation function.*
- $d : Q \times \mathcal{AG} \to \mathbb{N}$ is the *action function*, that defines the *actions* which are available to each agent $a \in \mathcal{AG}$ at each location $q \in Q$. We assume that each agent has at least one available action at each location, that could be thought of as the action *do-nothing* and without loss of generality we assume it is always the first, denoted thus by the natural number 1. As a consequence, we have that $d(q, a) \geq 1$, for each $a \in \mathcal{AG}, q \in Q$. Given a team A, an *action profile* $\boldsymbol{\alpha_A}$ is a vector assigning an action to each agent $a \in A$. Action profiles represent joint actions of a team. For each location $q \in Q$ and team $A = \{a_1, \ldots, a_k\} \subseteq \mathcal{AG}$, we denote by $D_A(q)$ the set of *action profiles available to the team A at the location q*, defined as $D_A(q) = \{1, \ldots, d(q, a_1)\} \times \ldots \times \{1, \ldots, d(q, a_k)\}$. For the sake of readability, we denote $D_{\mathcal{AG}}(q)$ by $D(q)$. Given a team A, an agent $a \in A$, and an action profile $\boldsymbol{\alpha_A}$, we will refer to the component of the vector $\boldsymbol{\alpha_A}$ corresponding to the agent a as $\boldsymbol{\alpha_A}(a)$. Actions (resp., action profiles) are usually denoted by $\alpha, \alpha_1, \ldots$ (resp., $\boldsymbol{\alpha}, \boldsymbol{\alpha_1}, \ldots$).
- $qty : Q \times \mathcal{AG} \times \mathbb{N} \to \mathbb{Z}^r$ is a partial function, defined over the triples (q, a, i) with $i \in d(q, a)$, defining the amount of resources required by an available action to a given agent at a given location. A negative cost represents a resource consumption, while a positive one represents a resource production. Moreover, we have that $qty(q, a, 1) = \mathbf{0^r}$, that is the vector whose components are all equal to 0, for every $q \in Q, a \in \mathcal{AG}$ (doing nothing neither consumes nor produces resources). With an abuse of notation we also denote by qty the function defining the amount of resources required by an action profile $\boldsymbol{\alpha_A} \in D_A(q)$, that is $qty(q, \boldsymbol{\alpha_A}) = \sum_{a \in A} qty(q, a, \boldsymbol{\alpha_A}(a))$. Finally, we define the function $consd : Q \times \mathcal{AG} \times \mathbb{N} \to \mathbb{N}^r$ in such a way that $consd(q, a, \alpha)$ returns the vector of the resources which are

consumed by an agent a for an action α. This vector is obtained from $qty(q, a, \alpha)$ by replacing the positive components, representing a resource production, with zeros, and the negative components, representing a resource consumption, with their absolute values.

- $\delta : Q \times \mathbb{N}^n \to Q$ is the *transition function*, that defines the next location reached from q if the agents perform the action profile $\boldsymbol{\alpha} \in \mathbb{N}^n$. It is a partial function defined over the set of pairs $(q, \boldsymbol{\alpha}) \in Q \times \mathbb{N}^n$ such that $\boldsymbol{\alpha} \in D(q)$.
- $\rho : \mathcal{M} \times Q \times \mathcal{AG} \to \mathbb{N}^r$ 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.
- $\boldsymbol{m_0}$ is the initial global availability of resources. It represents the resource availability on the market at the initial state of the system.

In order to give the formal semantics we must define the following notions.

Definition 2.2 [configuration and computation] A *configuration* c of a priced game graph G is a pair $\langle q, \boldsymbol{m} \rangle \in Q \times \mathcal{M}$. A *computation* over G is an infinite sequence of configurations of G $\lambda = c_1 c_2 \ldots$, such that, for each i, if $c_i = \langle q_i, \boldsymbol{m_i} \rangle$ and $c_{i+1} = \langle q_{i+1}, \boldsymbol{m_{i+1}} \rangle$, then there exists a transition $\delta(q_i, \boldsymbol{\alpha}) = q_{i+1}$, with $\boldsymbol{\alpha} = \langle \alpha_1, \ldots, \alpha_n \rangle$, such that $\boldsymbol{m_{i+1}} = \boldsymbol{m_i} + qty(q_i, \alpha)$.

Let $\lambda = c_1 c_2 \ldots$ be a computation. We denote by $\lambda[i]$ the i-th configuration c_i in λ and by $\lambda[i, j]$, with $i \leq j$, the finite sequence of configurations $c_i c_{i+1} \ldots c_j$ in λ.

Definition 2.3 [strategy] A *strategy* F_A for the team of agents A is a function which associates, to each finite sequence of configurations $c_1 c_2 \ldots c_s$, with $s \geq 1$, and $c_s = \langle q_s, \boldsymbol{m_s} \rangle$, an action profile $\boldsymbol{\alpha_A} \in D_A(q_s)$.

In other words, a strategy F_A determines the behavior, that is action profile $\boldsymbol{\alpha_A}$, of the agents in the team A. Anyway, for each action profile $\boldsymbol{\alpha_A}$ and configuration $c \in Q \times \mathcal{M}$, depending on the different action profiles of the opponent team $\mathcal{AG} \setminus A$, there are several possibilities for the next configuration, called *outcomes of $\boldsymbol{\alpha_A}$ at the configuration c*.

In determining such outcomes, we do not consider unfeasible action profiles of the proponent (resp., opponent) team, that is, action profiles which consume an amount of resources greater than the current availability $\boldsymbol{m}$ or produce an amount of resources greater than $\boldsymbol{m_0} - \boldsymbol{m}$. (Notice that an action profiles can be unfeasible even if each action of a single agent is feasible) This reflects the natural concept that the proponent team cannot conceive a strategy based on unfeasible action profiles. Analogously, the opponent team cannot prevent the achievement of a goal by the proponent through an unfeasible action profile.

Formally, given a configuration $c = \langle q, \boldsymbol{m} \rangle$, the set $out(c, \boldsymbol{\alpha_A})$ of the outcomes of $\boldsymbol{\alpha_A}$ at the configuration c, contains $\langle q', \boldsymbol{m'} \rangle$ if there exists $\boldsymbol{\alpha_{\mathcal{AG}}}$ extending $\boldsymbol{\alpha_A}$ such that

- $q' = \delta(q, \boldsymbol{\alpha}_{\mathcal{AG}})$,
- $\boldsymbol{m'} = \boldsymbol{m} + qty(q, \boldsymbol{\alpha}_{\mathcal{AG}})$,
- $\mathbf{0} \leq qty(q, \boldsymbol{\alpha}_{\boldsymbol{A}}) + \boldsymbol{m} \leq \boldsymbol{m}_{\mathbf{0}}$.
- $\mathbf{0} \leq qty(q, \boldsymbol{\alpha}_{\mathcal{AG}\setminus \boldsymbol{A}}) + \boldsymbol{m} \leq \boldsymbol{m}_{\mathbf{0}}$, where $\boldsymbol{\alpha}_{\mathcal{AG}\setminus \boldsymbol{A}}$ is the restriction of $\boldsymbol{\alpha}_{\mathcal{AG}}$ to the team $\mathcal{AG} \setminus A$.

As a consequence of the above definition, given a configuration c, a strategy F_A gives rise to a tree of computations. A computation $\lambda = c_1 c_2 \ldots$ is called an *outcome of the strategy* F_A *from the configuration* c, if $c_1 = c$ and $c_{i+1} \in out(c_i, F_A(\lambda[1, i]))$, for each $i \geq 1$. The set of such computations is denoted by $out(c, F_A)$.

Finally, we introduce the concept of consistent strategy, whose outcomes are such that in every configuration the agents have enough money to realize it.

Definition 2.4 [$(\$, \boldsymbol{m}_{\mathbf{0}})$-strategy] Let $\$ \in \mathcal{N}$ and $\boldsymbol{m}_{\mathbf{0}} \in \mathcal{M}$. A strategy F_A is said to be a $(\$, \boldsymbol{m}_{\mathbf{0}})$*-strategy* if for each $\lambda = c_1 c_2 \ldots$, with $c_i = \langle q_i, \boldsymbol{m_i} \rangle$ for all i, belonging to $out(c_1, F_A)$, the following conditions hold, for every $i \geq 1$ and $a \in A$:

- $\mathbf{0}^r \leq \boldsymbol{m_i} \leq \boldsymbol{m}_{\mathbf{0}}$,
- $\sum_{j=1}^{i} \rho(\boldsymbol{m_j}, q_j, a) \cdot consd(q_j, a, F_A(q_1 \ldots q_j)(a)) \leq \$[a]$.

The two above conditions state the *consistency* of a computation with respect to the $\boldsymbol{m}$ and $\$$, where the dot operator denotes the usual scalar product of vectors. Observe that, in the second condition, only the money availability of the team A is tested. Actually, we suppose that the opponent team $\mathcal{AG} \setminus A$ always have money enough to make its choice. Notice also that, as another consequence of the second condition, the actions *producing* resources do not cause a reimbursement of money to the agents.

Observe that, as it is usual when dealing with temporal logics, we guarantee that priced game structures are non-blocking, in the sense that a finite prefix of a computation satisfying the two conditions can always be followed by a next configuration without violating the consistency. Actually, a team A can always choose all the do-nothing actions of its agents, and the opponent team have choices which do not require an amount of resources greater than the resource availability.

We now can give the semantics of PRB-ATL formulae. The truth of a formula of PRB-ATL is defined with respect to a priced game structure G and a configuration $c = \langle q, \boldsymbol{m} \rangle$. The definition of the semantics of PRB-ATL is completed by the definition of the satisfaction relation $\models$, as follows:

- $(G, c) \models p$ iff $p \in \pi(q)$
- $(G, c) \models \neg\psi$ iff $(G, c) \not\models \psi$
- $(G, c) \models \psi_1 \wedge \psi_2$ iff $(G, c) \models \psi_1$ and $(G, c) \models \psi_2$
- $(G, c) \models \langle\langle A^{\$} \rangle\rangle \bigcirc \psi$ iff there exists a $(\$, \boldsymbol{m}_{\mathbf{0}})$-strategy F_A such that, for all $\lambda \in out(c, F_A)$, it holds that $(G, \lambda[1]) \models \psi$
- $(G, c) \models \langle\langle A^{\$} \rangle\rangle \psi_1 \mathcal{U} \psi_2$ iff there exists a $(\$, \boldsymbol{m}_{\mathbf{0}})$-strategy F_A such that, for all $\lambda \in out(c, F_A)$, there exists $i \geq 0$ such that $(G, \lambda[i]) \models \psi_2$ and, for all $0 \leq j < i$,

it holds that $(G, \lambda[j]) \models \psi_1$

- $(G, c) \models \langle\langle A^{\boldsymbol{\$}} \rangle\rangle \Box \psi$ iff there exists a $(\boldsymbol{\$}, \boldsymbol{m_0})$-strategy F_A such that, for all $\lambda \in out(c, F_A)$, it holds that $(G, \lambda[i]) \models \psi$ for all $i \geq 0$
- $(G, c) \models \sim \boldsymbol{b}$ iff $\boldsymbol{m} \sim \boldsymbol{b}$

3 Model checking

In this section we study the model checking problem for PRB-ATL and we show that it is in EXPTIME and has a PSPACE lower bound. The model checking problem consists in verifying whether a formula φ is satisfied with respect to a configuration $c = \langle q, \boldsymbol{m} \rangle$ of a priced game structure G, $(G, c) \models \varphi$.

3.1 A model-checking algorithm

The algorithm for model checking our logic is mostly based on the ones proposed in [4] and [3] for model checking, respectively, ATL and its resource-bounded extension RB-ATL (see Algorithm 1, where $\mathcal{M}^{\leq \boldsymbol{m}}$ denotes the set $\{\boldsymbol{m'} \in \mathcal{M} \mid \boldsymbol{m'} \leq \boldsymbol{m}\}$, for a resource availability $\boldsymbol{m} \in \mathcal{M}$). Roughly speaking, it computes, for each sub-formula φ' of the formula φ to be checked against a model G, the set of configurations in which φ' holds. Note that in our setting it is not sufficient to compute the set of states where a sub-formula holds, we need to take into account also the current resource availability. The main issues when dealing with bounds on resources are the following. First, the set of sub-formulae must be replaced by an extended set of formulae, called Sub^+, that includes also, for each sub-formula of the form $\langle\langle A^{\boldsymbol{\$}} \rangle\rangle \varphi'$, all the formulae $\langle\langle A^{\boldsymbol{\$'}} \rangle\rangle \varphi'$, with $\boldsymbol{\$'} < \boldsymbol{\$}$. Second, we need to take trace not only of the states but also of the configurations since we must take into account the resource availability on the market to guarantee that during the computation the needed resources are available, as well as to be able to compute the current prices of these 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. Let us stress that such a requirement is crucial to preserve decidability even when actions are allowed to produce resources.

Algorithm 1 heavily relies on the function $Pre(A, [\varphi], \boldsymbol{\$}, G, \boldsymbol{m})$ (see Algorithm 2) which computes the pre-image of a set of configurations with respect to the transition relation of G and according to the resource and money availabilities. More precisely, it returns all the configurations $\langle q', \boldsymbol{m'} \rangle$ for which there exists an action profile $\boldsymbol{\alpha_A}$ such that for all $\boldsymbol{\alpha_{\mathcal{AG}}}$ extending $\boldsymbol{\alpha_A}$, the configuration $\langle q'', \boldsymbol{m''} \rangle$ satisfies φ where $q'' = \delta(q', \boldsymbol{\alpha_{\mathcal{AG}}})$, $\boldsymbol{m''} = \boldsymbol{m'} + qty(q', \boldsymbol{\alpha_{\mathcal{AG}}})$, and the conditions of Definition 2.4 are satisfied.

Let M be the maximum component occurring in the initial resource availability vector $\boldsymbol{m_0}$ and S be the maximum component occurring in the money endowment vectors $\boldsymbol{\$_i}$ occurring in φ. Thus the proposed algorithm runs in time $O(|\varphi| \cdot |G| \cdot M^r \cdot S^n)$. Indeed, the outermost loop (line 1) is executed at most $|Sub^+(\varphi)| = |\varphi| \cdot S^n$ times. The cases in which φ' is of the form $\langle\langle A^{\boldsymbol{\$}} \rangle\rangle \psi_1 \mathcal{U} \psi_2$ or $\langle\langle A^{\boldsymbol{\$}} \rangle\rangle \Box \psi$ require 2 loops.

Algorithm 1 $MC(\varphi, G, q, \boldsymbol{m})$ `// returns` $true$ `iff` $(G, \langle q, \boldsymbol{m}\rangle) \models \varphi$

```
for all $\varphi' \in Sub^+(\varphi)$ do
    if $\varphi' = p$ then
        $[\varphi'] \leftarrow \{\langle q', \boldsymbol{m'}\rangle \mid p \in \pi(q'), \boldsymbol{m'} \leq \boldsymbol{m_0}\}$
    else if $\varphi' = \sim \boldsymbol{b}$ then
        $[\varphi'] \leftarrow \{\langle q', \boldsymbol{m'}\rangle \mid \boldsymbol{m'} \sim \boldsymbol{b}$ and $\boldsymbol{m'} \leq \boldsymbol{m_0}\}$
    else if $\varphi' = \neg\psi$ then
        $[\varphi'] \leftarrow (Q \times \mathcal{M}^{\leq \boldsymbol{m_0}}) \setminus [\psi]$
    else if $\varphi' = \psi_1 \wedge \psi_2$ then
        $[\varphi'] \leftarrow [\psi_1] \cap [\psi_2]$
    else if $\varphi' = \langle\langle A^{\boldsymbol{\$}}\rangle\rangle \bigcirc \psi$ then
        $[\varphi'] \leftarrow Pre(A, [\psi], \boldsymbol{\$}, G, \boldsymbol{m})$
    else if $\varphi' = \langle\langle A^{\boldsymbol{0^n}}\rangle\rangle \psi_1 \mathcal{U} \psi_2$ then
        $\tau \leftarrow [false], \sigma \leftarrow [\psi_2]$
        while $\tau \neq \sigma$ do
            $\tau \leftarrow \sigma$
            $\sigma \leftarrow \tau \cup (Pre(A, \tau, \boldsymbol{0^n}, G, \boldsymbol{m}) \cap [\psi_1])$
        end while
        $[\varphi'] \leftarrow \sigma$
    else if $\varphi' = \langle\langle A^{\boldsymbol{\$}}\rangle\rangle \psi_1 \mathcal{U} \psi_2$, with $\boldsymbol{\$} \geq \boldsymbol{0^n}$, $\neg(\boldsymbol{\$} = \boldsymbol{0^n})$ then
        $\tau \leftarrow [\langle\langle A^{\boldsymbol{0^n}}\rangle\rangle \psi_1 \mathcal{U} \psi_2]$
        for all $\boldsymbol{\$'}$ s.t. $\boldsymbol{\$'} \leq \boldsymbol{\$}$, $\neg(\boldsymbol{\$'} = \boldsymbol{\$})$ do
            $\sigma \leftarrow \tau \cup (Pre(A, [\langle\langle A^{\boldsymbol{\$'}}\rangle\rangle \psi_1 \mathcal{U} \psi_2], \boldsymbol{\$} - \boldsymbol{\$'}, G, \boldsymbol{m}) \cap [\psi_1])$
            while $\tau \neq \sigma$ do
                $\tau \leftarrow \sigma$
                $\sigma \leftarrow \tau \cup (Pre(A, \tau, \boldsymbol{0^n}, G, \boldsymbol{m}) \cap [\psi_1])$
            end while
        end for
        $[\varphi'] \leftarrow \sigma$
    else if $\varphi' = \langle\langle A^{\boldsymbol{0^n}}\rangle\rangle \Box \psi$ then
        $\tau \leftarrow [true], \sigma \leftarrow [\psi]$
        while $\tau \neq \sigma$ do
            $\tau \leftarrow \sigma$
            $\sigma \leftarrow [\psi] \cap Pre(A, \tau, \boldsymbol{0^n}, G, \boldsymbol{m})$
        end while
        $[\varphi'] \leftarrow \sigma$
    else if $\varphi' = \langle\langle A^{\boldsymbol{\$}}\rangle\rangle \Box \psi$, with $\boldsymbol{\$} \geq \boldsymbol{0^n}$, $\neg(\boldsymbol{\$} = \boldsymbol{0^n})$ then
        $\tau \leftarrow [\langle\langle A^{\boldsymbol{0^n}}\rangle\rangle \Box \psi]$
        for all $\boldsymbol{\$'}$ s.t. $\boldsymbol{\$'} \leq \boldsymbol{\$}$, $\neg(\boldsymbol{\$'} = \boldsymbol{\$})$ do
            $\sigma \leftarrow \tau \cup (Pre(A, [\langle\langle A^{\boldsymbol{\$'}}\rangle\rangle \Box \psi], \boldsymbol{\$} - \boldsymbol{\$'}, G, \boldsymbol{m}) \cap [\psi])$
            while $\tau \neq \sigma$ do
                $\tau \leftarrow \sigma$
                $\sigma \leftarrow \tau \cup (Pre(A, \tau, \boldsymbol{0^n}, G, \boldsymbol{m}) \cap [\psi])$
            end while
        end for
        $[\varphi'] \leftarrow \sigma$
    end if
end for
return $(\langle q, \boldsymbol{m}\rangle \in [\varphi])$
```

Algorithm 2 $Pre(A, [\varphi], \boldsymbol{\$}, G, \boldsymbol{m})$

```
$res \leftarrow \emptyset$
for all $\langle q', \boldsymbol{m'}\rangle \in Q \times \mathcal{M}^{\leq \boldsymbol{m_0}}$ do
    for all $\boldsymbol{\alpha_A} \in D_A(q')$ s.t. $\boldsymbol{0} \leq qty(q, \boldsymbol{\alpha_A}) + \boldsymbol{m} \leq \boldsymbol{m_0}$ do
        $ins \leftarrow true$
        for all $\langle q'', \boldsymbol{m''}\rangle$ where $(i)$ $q'' = \delta(q', \boldsymbol{\alpha_{\mathcal{AG}}})$ for some $\boldsymbol{\alpha_{\mathcal{AG}}} \in D(q')$
                        that generalizes $\boldsymbol{\alpha_A}$, $(ii)$ $\boldsymbol{m''} = \boldsymbol{m'} + qty(q, \boldsymbol{\alpha_{\mathcal{AG}}})$, and
                        $(iii)$ $\boldsymbol{0} \leq qty(q, \boldsymbol{\alpha_{\mathcal{AG} \setminus A}}) + \boldsymbol{m} \leq \boldsymbol{m_0}$ do
            if $\langle q'', \boldsymbol{m''}\rangle \notin [\varphi]$ or $\neg(\boldsymbol{0^r} \leq \boldsymbol{m''} \leq \boldsymbol{m_0})$
                or $\neg(\boldsymbol{\$}[a] \geq \rho(\boldsymbol{m'}, q', a) \cdot consd(q', a, \boldsymbol{\alpha_{\mathcal{AG}}}(a))$ for some $a \in A$ then
                $ins \leftarrow false$
            end if
        end for
        if $ins$ then
            $res \leftarrow res \cup \{\langle q', \boldsymbol{m'}\rangle\}$
        end if
    end for
end for
return $res$
```

The outer one (lines 19 and 36, respectively) is executed at most S^n times, while the inner one (lines 21 and 38, respectively) is executed at most $|Q \times \mathcal{M}^{\leq \boldsymbol{m_0}}| \leq |Q| \cdot M^r$ times. Since $|Q| \leq |G|$, we have the above asymptotic complexity. Notice that the complexity of the function Pre can be ignored as it is absorbed in the complexity of the main algorithm. To sum up, the algorithm runs in exponential time with respect to the size of the input.

Theorem 3.1 *Model checking* PRB-ATL *can be solved in exponential time in the number of agents and resources.*

3.2 PSPACE-hardness

Here, we provide a lower bound to the complexity of the model checking problem for PRB-ATL. To this aim, we will reduce the problem of determining the truth value of *Fully Quantified Boolean Formulae*, *TQBF* problem [12], to the model checking problem for PRB-ATL. A fully quantified Boolean formula is a Boolean formula in which all the Boolean variables occur inside the scope of an existential or universal quantifier. A fully quantified Boolean formula is said to be in prenex normal form if all the quantifiers appear at the beginning of the formula and each quantifier's scope is everything following it. Any formula may be easily put into prenex normal form, thus we consider formulae in this form only. Without loss of generality, we can also assume that the Boolean quantifier-free part of the formula is in conjunctive normal form with clauses having at most three literals, where a literal is either a variable or its negation. For example, $\forall x \exists y[(x \vee y) \wedge (\neg x \vee \neg y)]$ and $\exists x_1 \forall x_2 \exists x_3[(x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3)]$ are fully quantified Boolean formulae in the desired normal form. *TQBF* is the problem of determining whether a fully quantified Boolean formula is true.

Let $\Phi = Q_1 x_1 Q_2 x_2 \ldots Q_k x_k[(x_1^1 \vee x_2^1 \vee x_3^1) \wedge \ldots \wedge (x_1^h \vee x_2^h \vee x_3^h)]$ be a fully quantified Boolean formula, where Q_i are quantifiers, x_i are Boolean variables, for $1 \leq i \leq k$, and x_1^j, x_2^j, x_3^j are literals, for $1 \leq j \leq h$. We must exhibit a priced game structure G, a configuration c of G, and a PRB-ATL formula φ such that $(G, c) \models \varphi$ if and only if Φ is true.

The priced game structure G_Φ corresponging to Φ is computed as follows. (As an example, in Figure 1 the graph of the priced game structure generated from the formula $\Phi = \exists x_1 \forall x_2 \exists x_3[(x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3)]$ is shown). The set of locations is made up of the following locations:

- 3 locations, denoted $q_i, q_i^\top, q_i^\bot$ for each quantifier Q_i,
- the location q_{k+1},
- a location q_{c_i} for each clause c_i,
- 2 locations, denoted q_{x_i} and $q_{\neg x_i}$, for each variable x_i,
- the locations $q_\top$.

There exist $r = 2 \cdot k$ resources, two for each Boolean variable, that is, $R = \{R_{x_1}^\top, R_{x_1}^\bot, \ldots, R_{x_k}^\top, R_{x_k}^\bot\}$ and only one agent ($\mathcal{AG} = \{1\}$). Initially, there is only one item available for each resource, that is, the vector $\boldsymbol{m_0}$ has all components equal to 1. Notice that, since there is only one agent, there are only 2 possible teams, namely $\emptyset$ and $\mathcal{AG}$. This means that the reduction (and the hardness result) also applies to the extension of CTL with prices for bounded resources.

The idea of the reduction is that from any location q_i there are only two significant transitions, leading to $q_i^\top$ and $q_i^\perp$. The intended meaning of the transition leading to $q_i^\top$ (resp., $q_i^\perp$), which is called *true transition* (resp., *false transition*), is to assign the truth value *true* (resp., *false*) to the Boolean variable x_i. We need a machinery to remember such an assignment. To this aim, we make use of the resources and their limited availability: the transition leading from q_i to $q_i^\top$ (resp., $q_i^\perp$) is such that exactly 1 item of the resource $R_{x_i}^\top$ (resp., $R_{x_i}^\perp$) is consumed. Analogously, there exists a transition starting from the location q_{x_i} (resp., $q_{\neg x_i}$) and leading to $q_\top$. The transition leading from q_{x_i} (resp., $q_{\neg x_i}$) to $q_\top$ is such that exactly 1 item of the resource $R_{x_i}^\perp$ (resp., $R_{x_i}^\top$) is consumed. In this way, if in the location q_i the true (resp., false) transition has been chosen, when the game is on the location $q_{\neg x_i}$ (resp., q_{x_i}), the agent is not able to choose the transition leading to $q_\top$, indicating that the literal is false. There exists only one atomic proposition, namely p, that is true only over the location $q_\top$.

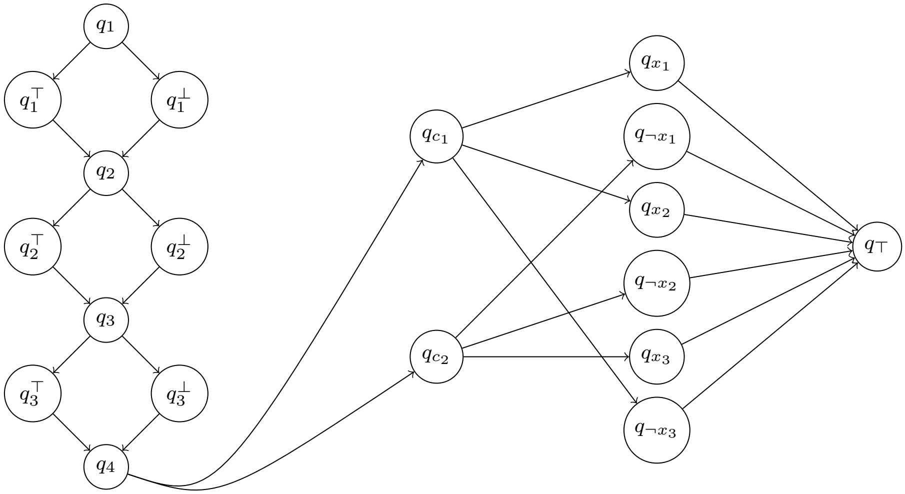

Fig. 1. The graph of the priced game structure G_Φ corresponding to the fully quantified Boolean formula $\Phi = \exists x_1 \forall x_2 \exists x_3 [(x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3)]$.

Let Φ be a fully quantified Boolean formula with k Boolean variables, let $\boldsymbol{m_0}$ be the vector, of length $2 \cdot k$, with all components equal to 1, and let $\varphi_\Phi = \langle\langle A_1^{\mathbf{0}}\rangle\rangle \bigcirc \langle\langle \mathcal{AG}^{\mathbf{0}}\rangle\rangle \bigcirc \langle\langle A_2^{\mathbf{0}}\rangle\rangle \bigcirc \langle\langle \mathcal{AG}^{\mathbf{0}}\rangle\rangle \bigcirc \ldots \langle\langle A_k^{\mathbf{0}}\rangle\rangle \bigcirc \langle\langle \mathcal{AG}^{\mathbf{0}}\rangle\rangle \bigcirc \langle\langle \mathcal{AG}^{\mathbf{0}}\rangle\rangle \bigcirc \langle\langle \emptyset^{\mathbf{0}}\rangle\rangle \bigcirc \langle\langle \mathcal{AG}^{\mathbf{0}}\rangle\rangle \bigcirc p$, with $A_i = \emptyset$ if Q_i is an universal quantifier, $A_i = \mathcal{AG}$, otherwise, for all $1 \leq i \leq k$.

It is possible to show that $(G_\Phi, \langle q_1, \boldsymbol{m_0}\rangle) \models \varphi_\Phi$ if and only if Φ is true.

Theorem 3.2 *The model checking problem for* PRB-ATL *is PSPACE-hard.*

4 Optimal coalitions

In this section we define the problem of determining optimal coalitions that are capable to satisfy a PRB-ATL formula. In doing that, we introduce the notion of *parametric* PRB-ATL *formula*, that is, a PRB-ATL formula in which *parametric team operators* $\langle\langle X^{\$}\rangle\rangle$ may occur in the place of the classical team operators $\langle\langle A^{\$}\rangle\rangle$. If $\langle\langle X^{\$}\rangle\rangle$ is a parametric team operator, then X is a *team variable*.

Given a parametric PRB-ATL formula φ with team variables $\boldsymbol{X} = \langle X_1, \ldots, X_k\rangle$, and a vector $\boldsymbol{A} = \langle A_1, \ldots, A_k\rangle$ of k teams, we denote by $\varphi[\boldsymbol{X}/\boldsymbol{A}]$ the PRB-ATL formula obtained from φ by replacing every occurrence of X_i with A_i, for every $1 \leq i \leq k$. We denote by Sol_φ the set of the vectors $\boldsymbol{A}$ of teams such that $(G, c) \models \varphi[\boldsymbol{X}/\boldsymbol{A}]$.

Finally, we associate a *cost* to each PRB-ATL formula by means of the function *f_cost*, defined as follows. Suppose that a team A is represented by means of the characteristic vector of the set A, that is, A is a vector of n binary components such that the i-th component is 1 if and only if the agent i belongs to the team A, for each $1 \leq i \leq n$. For every PRB-ATL formula φ, containing the team operators $\langle\langle A_1^{\$_1}\rangle\rangle, \langle\langle A_2^{\$_2}\rangle\rangle, \ldots, \langle\langle A_k^{\$_k}\rangle\rangle$,

$$f\_cost(\varphi) = \sum_{i=1}^{k}(\$_{\boldsymbol{i}} \cdot A_i)$$

4.1 The problem

In this section, we introduce the problem of finding an optimal (with respect to the function *f_cost*) vector of teams that satisfies a given parametric PRB-ATL formula.

Given a parametric PRB-ATL formula φ, a priced game structure G, a configuration c of G, the *Optimal Coalition* problem for PRB-ATL (*OC*, for short) consists in finding a vector $\boldsymbol{A}$, if any, such that $(G, c) \models \varphi[\boldsymbol{X}/\boldsymbol{A}]$ and that minimizes the function *f_cost*.

Thus, an algorithm to solve such problem takes the following parameters as input:

- a PRB-ATL formula φ, over team variables $\boldsymbol{X} = \langle X_1, X_2, \ldots, X_k\rangle$
- the priced game structure G,
- a configuration c of G,

and outputs the triple $\langle res, \boldsymbol{A}^*, cost\rangle$, where:

- $res \in \{true, false\}$ is $true$ if and only if there exists a vector of teams $\boldsymbol{A}$ such that $(G, c) \models \varphi[\boldsymbol{X}/\boldsymbol{A}]$,
- $\boldsymbol{A}^*$ is a vector of teams such that $f\_cost(\varphi[\boldsymbol{X}/\boldsymbol{A}^*]) = \min_{\boldsymbol{A} \in Sol_\varphi} f\_cost(\varphi[\boldsymbol{X}/\boldsymbol{A}])$; its value is undefined if $Sol_\varphi = \emptyset$,
- $cost \in \mathbb{N}$ is the value of the function *f_cost* applied to $\varphi[\boldsymbol{X}/\boldsymbol{A}^*]$.

Our aim is to show that *OC* has the same complexity as the model checking problem for PRB-ATL.

The upper bound is given by a brute force algorithm that tries to solve the model checking problem for each possible vector of teams. Since there are $(2^n)^k$ different vectors of teams, the algorithm will call $(2^n)^k$ times the algorithm for model checking PRB-ATL.

For the lower bound, consider the decisional version of the *OC* problem: given a parametric PRB-ATL formula φ, a priced game structure G, a configuration c of G, and an integer U, decide if there is a vector $\boldsymbol{A}$ such that $(G, c) \models \varphi[\boldsymbol{X}/\boldsymbol{A}]$ and *f_cost* applied to $\varphi[\boldsymbol{X}/\boldsymbol{A}]$ does not exceed U. The hardness for such a problem directly descends from Theorem 3.2.

Theorem 4.1 *The decisional OC problem is in between PSPACE and EXPTIME.*

5 Discussion

In this paper we have studied the model checking problem for the logic PRB-ATL, very suitable for modeling scenarios in which teams of agents have to perform a task and they are subject to boundedness of resources.

Several formalisms have already been proposed to model such a kind of real-word situations [3,7]. Our main contribution is to present a new formalism, called PRB-ATL, based on a stronger notion of resource bounds and we show that its model checking problem We also introduce the problem, called the *optimal coalitions* (*OC*) problem, of finding optimal coalitions (with respect to the amount of money needed by the different coalitions) capable to perform the task. We have shown that both the model checking problem for PRB-ATL and the *OC* problem are in EXPTIME and have a PSPACE lower bound. As a matter of fact, to solve the *OC* problem we give an algorithm that explores the entire space of solutions to compute the optimum. The problem of finding the exact characterization of the computational complexity is an open problem and currently is under investigation.

Further research directions concern the study of variants of the logic, to naturally express more kinds of scenarios. As an example, it is interesting in our opinion to consider the money endowment as a component of the game arena (i.e., the model), instead of explicitly specifying a fresh money availability for each occurrence of a team operator in the formula. Furthermore, related to this argument, one can consider the money availability not as an input of the problem, but rather as a parameter to minimize, to establish how much money each agent should be provided with, to perform a given task.

References

[1] Ågotnes, T., W. van der Hoek and M. Wooldridge, *On the logic of coalitional games*, in: *AAMAS*, 2006, pp. 153–160.

[2] Alechina, N., B. Logan, N. H. Nga and A. Rakib, *A logic for coalitions with bounded resources*, in: *Proc. of the 21st International Joint Conference on Artificial Intelligence*, IJCAI '09, 2009, pp. 659–664.

[3] Alechina, N., B. Logan, N. H. Nga and A. Rakib, *Resource-bounded alternating-time temporal logic*, in: *Proc. of the 9th International Conference on Autonomous Agents and Multiagent Systems: Volume 1*,

AAMAS '10, 2010, pp. 481–488.
URL http://portal.acm.org/citation.cfm?id=1838206.1838274

[4] Alur, R., T. A. Henzinger and O. Kupferman, *Alternating-time temporal logic*, Journal of ACM **49** (2002), pp. 672–713.

[5] Bulling, N. and J. Dix, *Modelling and verifying coalitions using argumentation and ATL*, Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial **14** (2010), pp. 45–73.

[6] Bulling, N. and B. Farwer, *Expressing properties of resource-bounded systems: The logics RTL* and RTL*, in: J. Dix, M. Fisher and P. Novák, editors, *Computational Logic in Multi-Agent Systems (CLIMA X)* (2009), pp. 22–45.

[7] Bulling, N. and B. Farwer, *On the (un-)decidability of model checking resource-bounded agents*, in: *Proc. of the 19th European Conference on Artificial Intelligence*, ECAI '10, 2010, pp. 567–572.

[8] Dunne, P. E., S. Kraus, E. Manisterski and M. Wooldridge, *Solving coalitional resource games*, Artificial Intelligence **174** (2010), pp. 20–50.
URL http://www.sciencedirect.com/science/article/pii/S0004370209001076

[9] Goranko, V., *Coalition games and alternating temporal logics*, in: *Proc. of the 8th Conference on Theoretical Aspects of Rationality and Knowledge*, TARK '01 (2001), pp. 259–272.

[10] Pauly, M., *A logical framework for coalitional effectivity in dynamic procedures*, Bulletin of Economic Research **53** (2001), pp. 305–324.
URL http://dx.doi.org/10.1111/1467-8586.00136

[11] Pauly, M., *A modal logic for coalitional power in games*, Journal of Logic and Computation **12** (2002), pp. 149–166.

[12] Sipser, M., "Introduction to the Theory of Computation," International Thomson Publishing, 1996, 1st edition.