ON A PRICED RESOURCE-BOUNDED ALTERNATING μ-CALCULUS

Della Monica, Dario[1] and Lenzi, Giacomo[1]
[1]*University of Salerno, Italy*
{*ddellamonica, gilenzi*}*@unisa.it*

Keywords: μ-calculus, Multi-agent Systems, Coalition Logics, Bounded Resources, Model Checking.

Abstract: Much attention has been devoted in artificial intelligence to the verification of multi-agent systems and different logical formalisms have been proposed, such as Alternating Time Temporal Logic (ATL), Alternating μ-calculus (AMC), and Coalition Logic (CL). Recently, logics able to express bounds on resources have been introduced, such as RB-ATL and PRB-ATL, both of them based on ATL. The main contribution of this paper is the introduction and the study of a new formalism for dealing with bounded resources, based on μ-calculus. Such a formalism, called Priced Resource-Bounded Alternating μ-calculus (PRB-AMC), is an extension of both PRB-ATL and AMC. In analogy with PRB-ATL, we introduce a *price* for each resource. By considering that the resources have each a price (which may vary during the game) and that agents can buy them only if they have enough money, several real world scenarios can be adequately described. First, we show that the model checking problem for PRB-AMC is in EXPTIME and has a PSPACE lower bound. Then, we solve the problem of determining the minimal cost coalition of agents. Finally, we show that the satisfiability problem of PRB-AMC is undecidable, when the game is played on arenas with only one state.

1 INTRODUCTION

Much attention has been devoted in the artificial intelligence field to the verification of multi-agent systems. In that regard, different logical formalisms have been proposed, such as *Alternating-time Temporal Logic* (ATL) (Alur et al., 2002), *Alternating μ-calculus* (AMC) (Alur et al., 2002), and *Coalition Logic* (CL) (Pauly, 2002). Such logics allow one to predicate about the abilities of teams of agents with respect to specific tasks. Recently, some efforts have been done towards the definition of more powerful formalisms, which are able to capture also quantitative aspects related to the task to be performed. In particular, we mention RB-ATL (Alechina et al., 2009; Alechina et al., 2010) and RAL (Bulling and Farwer, 2010). By means of formulae of these logics it is possible to assign an endowment of resources to each agent of a team and express the property that the team is able to perform a given task with the available resources. In (Della Monica et al., 2011), a further variation of ATL, called *Priced Resource-Bounded Alternating Time Temporal Logic* (PRB-ATL), has been considered; in this logic a *price* for each resource is introduced and team operators are accordingly extended. By means of these features, several real world scenarios can be adequately described. All the formalisms introduced so far are based on ATL or CL. The main contribution of this paper is the introduction and the study of a new formalism for dealing with bounded resources, based on μ-calculus. Recall that the μ-calculus is an extension of modal logic with least and greatest fixpoints of monotone operators on sets. Intuitively, least fixpoints correspond to inductive definitions (e.g. liveness properties), and greatest fixpoints correspond to coinductive definitions (e.g. safety properties). Nesting fixpoints give further power to the μ-calculus so that it subsumes many temporal, dynamic, and game-theoretic logics used in system verification, artificial intelligence, game theory, etc.

The formalism we propose is called *Priced Resource-Bounded Alternating μ-calculus* (PRB-AMC). It is an extension of both AMC and PRB-ATL.

We study the model checking problem for PRB-AMC, which turns out to be decidable in EXPTIME and PSPACE-hard, analogously to what happens for PRB-ATL (Della Monica et al., 2011). We remark that in our logic, agents can both consume and produce resources. Note that, when production is allowed, the model checking problem can be undecidable (see, e.g, (Bulling and Farwer, 2010)). Our decidability property is due to the fact that although agents can produce resources, the production should not exceed the initial availability of the resources. Such a restriction to the notion of production 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.

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 (Wooldridge and Dunne, 2006; Dunne et al., 2010; Bulling and Dix, 2010). Analogously to what has been done in (Della Monica et al., 2011) for PRB-ATL, here we face this problem in the setting of priced resource-bounded agents with the goal specified by an PRB-AMC formula. In particular we study 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 also the optimal coalition problem is in EXPTIME and has a PSPACE lower bound.

Finally, we show that the satisfiability problem of PRB-AMC is undecidable, when the game is played on a one-point arena, that is, the underlying graph is constituted by a single vertex. (Notice that such an undecidability result does not immediately extend to generic graphs.) While the result seems to be weak per se, we conjecture that the problem is undecidable in the general setting and we hope to use the present result as a preliminary step towards the proof of the general case.

2 SYNTAX AND SEMANTICS

The scenario is the same as PRB-ATL. So, we have a set $\mathcal{AG}$ of n agents, a set RES of r resources, the set $\mathcal{M} = (\mathbf{N} \cup \{\infty\})^r$ of resource availabilities, the set $\mathcal{N} = (\mathbf{N} \cup \{\infty\})^n$ of money availabilities, where $\mathbf{N}$ is the set of all natural numbers $0,1,2,\ldots$. We let $\vec{b}, \vec{m}$ range over $\mathcal{M}$ and $\vec{\$}$ range over $\mathcal{N}$. Moreover, given a vector $\vec{\$}$, we will refer to the component corresponding to the agent a as $\vec{\$}[a]$.

On the logical side, we use a set of atomic propositions Π and a set of fixpoint variables VAR, to be used in μ-calculus formulas. The syntax of formulas is as follows:

$$\phi ::= p \mid X \mid \neg\phi \mid \phi \wedge \phi \mid \langle\langle A^{\vec{\$}} \rangle\rangle \bigcirc \phi \mid \mu X.\phi(X) \mid \ \sim \vec{b}$$

where $p \in \Pi$, $X \in VAR$, $A \subseteq \mathcal{AG}$, $\vec{\$} \in \mathcal{N}$, $\vec{b} \in \mathcal{M}$ and $\sim \in \{<,>,=,\leq,\geq\}$. Moreover, $\mu X.\phi(X)$ is defined only when X occurs in an even number of negations in ϕ, so that formulas define monotonic operators on sets and we can apply Knaster-Tarski Fixpoint Theorem (Tarski, 1955). Recall that the greatest fixpoint operator $\nu X.\phi(X)$ can be defined as usual, that is, $\nu X.\phi(X) = \neg\mu X.\neg\phi(\neg X)$.

The semantics is based on *priced game structures with environment*, i.e., tuples $G = (Q, \pi, ENV, d, qty, \delta, \rho)$. They are analogous to the priced game structures used in (Della Monica et al., 2011), the only new ingredient being the environment $ENV : VAR \to 2^{Q\times\mathcal{M}}$, with which we can evaluate formulas containing fixpoint variables. Recall that:

The semantics is based on priced game structures with environment analogous to the ones used in (Della Monica et al., 2011), i.e. tuples $G = (Q, \pi, ENV, d, qty, \delta, \rho)$; here there is one extra feature, that is an environment $ENV : VAR \to 2^{Q\times\mathcal{M}}$, with which we can evaluate formulas containing fixpoint variables. Recall that:

- Q is a finite set of locations, usually denoted $q, q_1, q_2, \ldots$.
- $\pi : Q \to 2^{\Pi}$ is a labeling function assigning to each location the set of all atomic propositions which are true on it.
- $d(q,a)$ is the number of actions available for the agent a on state q. We code actions with numbers from 1 to $d(q,a)$. We assume that $d(q,a) \geq 1$ (there is always at least one action available) and the action 1 means "doing nothing".
  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 $\vec{\alpha}_A$, we will refer to the component of the vector $\vec{\alpha}_A$ corresponding to the agent a as $\vec{\alpha}_A[a]$. Actions (resp., action profiles) are usually denoted by $\alpha, \alpha_1, \ldots$ (resp., $\vec{\alpha}, \vec{\alpha}_1, \ldots$).
- $qty(q,a,\alpha)$ is an element of $\mathbf{Z}^r$ representing the quantity of resources consumed or produced by the agent a while performing the action $\alpha \in d(q,a)$ on the location q ($\mathbf{Z}$ is the set of integers). Positive components represent resource productions, negative ones represent resource consumptions. $qty(q,a,1)$ is the zero vector, for all $q \in Q$, $a \in \mathcal{AG}$. With an abuse of notation we also de-

note by qty the function defining the amount of resources required by an action profile $\vec{\alpha}_A \in D_A(q)$, that is $qty(q, \vec{\alpha}_A) = \sum_{a \in A} qty(q, a, \vec{\alpha}_A(a))$.
- $\delta(q, \langle \alpha_1, \ldots, \alpha_n \rangle)$ is the transition function giving the state reached from q when the n agents perform the action profile $\langle \alpha_1, \ldots, \alpha_n \rangle \in D(q)$.
- $\rho(\vec{m}, q, a)$ is the price of the r resources depending on resource availability $\vec{m} \in \mathcal{M}$, the location $q \in Q$, and the agent a.

In order to define the semantics of PRB-AMC, we must introduce the notion of strategy. Unlike (Della Monica et al., 2011), here it is enough to consider only one-step strategies.

Let us fix the initial global availability of resources $\vec{m}_0$ and let A be a set of agents. A one-step strategy F_A for A is a function giving for each $(q, \vec{m}) \in Q \times \mathcal{M}$ an action profile $\vec{\alpha}_A$ containing a move $\vec{\alpha}[a]$ for each $a \in A$. The outcome of a one-step strategy on $(q, \vec{m})$ is the set of all configurations $(q', \vec{m}') \in Q \times \mathcal{M}$ such that there is an extension $\vec{\alpha}_{\mathcal{AG}}$ of $\vec{\alpha}_A$ to $\mathcal{AG}$ such that:
- $q' = \delta(q, \vec{\alpha}_{\mathcal{AG}})$,
- $\vec{m}' = \vec{m} + qty(q, \vec{\alpha}_{\mathcal{AG}})$,
- $0 \leq \vec{m} + qty(q, \vec{\alpha}_{\mathcal{AG} \setminus A}) \leq \vec{m}_0$,

where $\vec{\alpha}_{\mathcal{AG} \setminus A}$ is the restriction of $\vec{\alpha}_{\mathcal{AG}}$ to $\mathcal{AG} \setminus A$.

A one-step $(\vec{\$}, \vec{m}_0)$-strategy F_A is a strategy such that for every $(q', \vec{m}')$ in the outcome of F_A on $(q, \vec{m})$, we have:
- $0 \leq \vec{m}' \leq \vec{m}_0$;
- $\rho(\vec{m}, q, a) \cdot consumed(q, a, F_A(q)[a]) \leq \$[a]$, for all $a \in A$,

where $consumed(q, a, \alpha)$ is obtained from $qty(q, a, \alpha)$ by replacing the positive components, representing resource productions, with zeros, and the negative ones, representing resource consumptions, with their absolute values.

We define the semantics of our logic in two steps. As a first step, we define a preliminary *pre-modelhood* relation, and as a second step, we define the proper *modelhood* relation, that makes use of the former one. The *pre-modelhood* relation is a quinary relation, denoted by:

$$G, \vec{m}_0, q, \vec{m} \models_0 \phi,$$

where G is a priced game structure with environment, $\vec{m}_0$ is the initial availability, q is a location, $\vec{m}$ is the current availability and ϕ is a formula.

We always suppose that $\vec{m} \leq \vec{m}_0$ and $\vec{m}_0$ has the same infinite components as $\vec{m}$.

The definition of $\models_0$ is by induction on ϕ, and the clauses are:
- $G, \vec{m}_0, q, \vec{m} \models_0 p$ iff $p \in \pi(q)$;
- $G, \vec{m}_0, q, \vec{m} \models_0 X$ iff $(q, \vec{m}) \in ENV(X)$;
- $G, \vec{m}_0, q, \vec{m} \models_0 \neg\phi$ iff not $G, \vec{m}_0, q, \vec{m} \models_0 \phi$;
- $G, \vec{m}_0, q, \vec{m} \models_0 \phi \wedge \phi'$ iff $G, \vec{m}_0, q, \vec{m} \models_0 \phi$ and $G, \vec{m}_0, q, \vec{m} \models_0 \phi'$;
- $G, \vec{m}_0, q, \vec{m} \models_0 \langle\langle A^{\vec{\$}} \rangle\rangle \bigcirc \phi$ iff there exists a $(\vec{\$}, \vec{m}_0)$-strategy F_A such that, for all configurations $(q', \vec{m}')$ in the output of F_A, it holds that $G, \vec{m}_0, q', \vec{m}' \models \phi$;
- $G, \vec{m}_0, q, \vec{m} \models_0 \mu X.\phi(X)$ iff $(q, \vec{m})$ belongs to the smallest set E such that $E = \{(q', \vec{m}') | G[X := E], \vec{m}_0, q', \vec{m}' \models_0 \phi\}$, where $G[X := E]$ is the same priced structure with environment as G, except that $ENV(X) = E$;
- $G, \vec{m}_0, q, \vec{m} \models_0 \sim \vec{b}$ iff $\vec{m} \sim \vec{b}$.

Finally, the proper *modelhood* relation is defined:

$$G, q, \vec{m} \models \phi \leftrightarrow G, \vec{m}, q, \vec{m} \models_0 \phi.$$

3 EXPRESSIVENESS

Recall from (Della Monica et al., 2011) that PRB-ATL has the following syntax:

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

where $p \in \Pi$, $A \subseteq \mathcal{AG}$, $\vec{\$} \in \mathcal{N}$, $\vec{b} \in \mathcal{M}$ and $\sim \in \{<, >, =, \leq, \geq\}$.

Intuitively $\langle\langle A^{\vec{\$}} \rangle\rangle \phi \mathcal{U} \phi'$ means that A can ensure ϕ until ϕ' holds, and $\langle\langle A^{\vec{\$}} \rangle\rangle \Box \phi$ means that A can ensure that ϕ holds forever.

So, PRB-ATL extends ATL, hence also the temporal logic CTL. Moreover, it is well known that CTL (resp., ATL) can be efficiently translated into μ-calculus (resp., the alternation-free fragment of AMC), but not conversely, and that CTL* (resp., ATL*) can be translated into the μ-calculus (resp., AMC), but not conversely.

In our more general setting, we extend the previous results as follows:

Theorem 3.1. PRB-ATL *can be translated in* PRB-AMC.

Proof. The proof hinges on the model checking algorithm for PRB-ATL. In fact, in order to make it clear that these operators are fixpoint definable, it suffices to rewrite the subroutines of the model checking algorithm 1 of (Della Monica et al., 2011) for the operators $\langle\langle A^{\vec{\$}} \rangle\rangle \phi_1 \mathcal{U} \phi_2$ and $\langle\langle A^{\vec{\$}} \rangle\rangle \Box \phi$.

We intend that the vector $\vec{\$}$ can contain finite and infinite components. The rewriting process goes by induction on the sum of the finite components of $\vec{\$}$.

We say that $\vec{\$}$ is zero-infinite if it consists only of zeros and infinites, and, for every $\vec{\$}$, we denote by $\vec{\$}_0$ the least vector with the same infinite components as $\vec{\$}$

(which is necessarily zero-infinite). In the algorithms we assume the convention $\infty - \infty = \infty$.

Rather than distinguishing two subroutines for zero and nonzero money assignments as in (Della Monica et al., 2011), we distinguish two subroutines for zero-infinite and non-zero-infinite money assignments.

In all our subroutines we replace the *Pre* operators with next operators $\langle\langle A^{\vec{\$}}\rangle\rangle \bigcirc \phi$, which are available in PRB-AMC.

We fix a priced arena with environment G and an initial availability $\vec{m}$. Given a formula ϕ, we use the notation $[\phi]$ to denote the set $\{(q', \vec{m}') \mid G, \vec{m}, q', \vec{m}' \models_0 \phi\}$, where $\models_0$ is the auxiliary pre-modelhood relation defined in the previous section. By definition of the proper modelhood relation $\models$, we have $(q, \vec{m}) \in [\phi]$ if and only if $G, q, \vec{m} \models \phi$, for each $q \in Q$.

Let us begin with the subroutine for $\phi = \langle\langle A^{\vec{\$}}\rangle\rangle \psi_1 \mathcal{U} \psi_2$ when $\vec{\$}$ is zero-infinite.

1: $\tau \leftarrow [false]$
2: $\sigma \leftarrow [\psi_2]$
3: **while** $\tau \neq \sigma$ **do**
4: $\quad \tau \leftarrow \sigma$
5: $\quad \sigma \leftarrow \tau \cup ([\langle\langle A^{\vec{\$}}\rangle\rangle \bigcirc \tau] \cap [\psi_1])$
6: **end while**
7: $[\phi] \leftarrow \sigma$

Now we observe that the while loop (line 3) calculates fixpoints. More precisely, it is equivalent to a simultaneous assignment $\sigma, \tau := \mu X. \psi_2 \vee (\langle\langle A^{\vec{\$}}\rangle\rangle \bigcirc X \wedge \psi_1)$. By replacing the while loop with a fixpoint assignment we obtain the algorithm:

1: $\tau \leftarrow [false]$
2: $\sigma \leftarrow [\psi_2]$
3: $\sigma, \tau \leftarrow \mu X. \psi_2 \vee (\langle\langle A^{\vec{\$}}\rangle\rangle \bigcirc X \wedge \psi_1)$
4: $[\phi] \leftarrow \sigma$

where it is clear that the semantics of ϕ is definable in PRB-AMC.

Likewise, if $\vec{\$}$ is not zero-infinite then we have:

1: $\tau \leftarrow [\langle\langle A^{\vec{\$}_0}\rangle\rangle \psi_1 \mathcal{U} \psi_2]$
2: **for all** $\vec{\$}' < \vec{\$}$ with the same infinites as $\vec{\$}$ **do**
3: $\quad \sigma \leftarrow \tau \cup ([\langle\langle A^{\vec{\$}-\vec{\$}'}\rangle\rangle \bigcirc \langle\langle A^{\vec{\$}'}\rangle\rangle \psi_1 \mathcal{U} \psi_2] \cap [\psi_1])$
4: $\quad$ **while** $\tau \neq \sigma$ **do**
5: $\quad\quad \tau \leftarrow \sigma$
6: $\quad\quad \sigma \leftarrow \tau \cup ([\langle\langle A^{\vec{\$}_0}\rangle\rangle \bigcirc \tau] \cap [\psi_1])$
7: $\quad$ **end while**
8: **end for**
9: $[\phi] \leftarrow \sigma$

The first line of the algorithm is a fixpoint definition by the zero-infinite case. Moreover, in each iteration of the for loop (line 2), the first assigment is a fixpoint definition by induction, and the while loop (line 4) is equivalent to a fixpoint assignment on the variables σ and τ. By replacing the while loop with this fixpoint assignment, we have a fixpoint definition of σ and τ at the end of every iteration of the for loop. So, at the end of the algorithm we have a fixpoint definition of the semantics of ϕ.

The situation is analogous for $\phi = \langle\langle A^{\vec{\$}}\rangle\rangle \Box \psi$. Let us begin with the subroutine for $\langle\langle A^{\vec{\$}}\rangle\rangle \Box \psi$ when $\vec{\$}$ is zero-infinite.

1: $\tau \leftarrow [true]$
2: $\sigma \leftarrow [\psi]$
3: **while** $\tau \neq \sigma$ **do**
4: $\quad \tau \leftarrow \sigma$
5: $\quad \sigma \leftarrow [\langle\langle A^{\vec{\$}}\rangle\rangle \bigcirc \tau] \cap [\psi]$
6: **end while**
7: $[\phi] \leftarrow \sigma$

In this case, the while loop (line 3) calculates a greatest fixpoint, i.e., $\sigma, \tau := \nu X. \langle\langle A^{\vec{\$}}\rangle\rangle \bigcirc X \wedge \psi$. By replacing the while loop with a fixpoint assignment, we have a fixpoint definition of the semantics of ϕ.

Finally, if $\vec{\$}$ is not zero-infinite then we have:

1: $\tau \leftarrow [\langle\langle A^{\vec{\$}_0}\rangle\rangle \Box \psi]$
2: **for all** $\vec{\$}' < \vec{\$}$ with the same infinites as $\vec{\$}$ **do**
3: $\quad \sigma \leftarrow \tau \cup ([\langle\langle A^{\vec{\$}-\vec{\$}'}\rangle\rangle \bigcirc \langle\langle A^{\vec{\$}'}\rangle\rangle \Box \psi] \cap [\psi])$
4: $\quad$ **while** $\tau \neq \sigma$ **do**
5: $\quad\quad \tau \leftarrow \sigma$
6: $\quad\quad \sigma \leftarrow \tau \cup ([\langle\langle A^{\vec{\$}_0}\rangle\rangle \bigcirc \tau] \cap [\psi])$
7: $\quad$ **end while**
8: **end for**
9: $[\phi] \leftarrow \sigma$

The first line of the algorithm is a fixpoint definition by the zero-infinite case. Moreover, in each iteration of the for loop (line 2), the first line is a fixpoint definition by induction, and the while loop (line 4) calculates a least fixpoint. So by replacing the while loop with a fixpoint assignment on σ and τ, we have a fixpoint definition of σ and τ at the end of every iteration of the for loop. At the end of the algorithm, we have a fixpoint definition of the semantics of ϕ.

$\Box$

Notice that the existence of an *efficient* translation from PRB-ATL to PRB-AMC (like the one of CTL into μ-calculus) is an open problem currently under investigation.

4 MODEL CHECKING

In (Della Monica et al., 2011), the authors consider the model checking problem for PRB-ATL, proving that it is in EXPTIME and it is PSPACE-hard. In this paper, we extend these results to PRB-AMC.

Theorem 4.1. *The model checking problem for* PRB-AMC *is in EXPTIME and it is PSPACE-hard.*

Proof. The PSPACE-hardness directly follows from the one of the model checking problem for PRB-ATL.

To prove the EXPTIME upper bound, we provide an exponential time recursive algorithm, called *set* (see Algorithm 1, where $\mathcal{M}^{\leq \vec{m}'}$ denotes the set $\{\vec{m} \in \mathcal{M} \mid \vec{m} \leq \vec{m}'\}$, for a resource availability $\vec{m}' \in \mathcal{M}$), which, given a priced game structure with environment G, a formula ϕ, and a resource availability $\vec{m}'$, outputs the set of all configurations $(q, \vec{m})$, with $\vec{m} \leq \vec{m}'$, which verify ϕ in G. The algorithm is a combination of those in (Della Monica et al., 2011) and (Emerson, 1996).

Note that the time complexity of the algorithm is $O((|G| \times |M|^r)^{|\phi|})$, while the space complexity is $O(|G| \times |M|^r)$, where M is the maximum component occurring in the initial resource availability vector $\vec{m}'$.

Finally, in order to check whether a formula ϕ is true over a game structure G and a configuration $(q, \vec{m})$ in G, the model checking algorithm simply consists in verifying if $(q, \vec{m})$ belongs to the output of $set(\phi, G, \vec{m})$.

Algorithm 1 $set(\phi, G, \vec{m}')$
`// computes the set of configurations`
$(q,\vec{m})$ `such that` $\vec{m} \leq \vec{m}'$ `and` $G, \vec{m}', q, \vec{m} \models_0 \phi$

if $\phi = \sim \vec{b}$ **then**
    return $\{(q,\vec{m}) \mid \vec{m} \sim \vec{b}$ and $\vec{m} \leq \vec{m}'\}$
else if $\phi = p$ /* $p \in \Pi$ */ **then**
    return $\{(q,\vec{m}) \mid p \in \pi(q), \vec{m} \leq \vec{m}'\}$
else if $\phi = X$ /* $X \in VAR$ */ **then**
    return $\{(q,\vec{m}) \mid (q,\vec{m}) \in ENV(X), \vec{m} \leq \vec{m}'\}$
else if $\phi = \neg\psi$ **then**
    return $(Q \times \mathcal{M}^{\leq \vec{m}'}) \setminus set(\psi, G, \vec{m}')$
else if $\phi = \psi_1 \wedge \psi_2$ **then**
    return $set(\psi_1, G, \vec{m}') \cap set(\psi_2, G, \vec{m}')$
else if $\phi = \langle\langle A^{\vec{\$}} \rangle\rangle \bigcirc \psi$ **then**
    return $Pre(A, \psi, \vec{\$}, G, \vec{m}')$
else if $\phi = \mu X.\psi(X)$ **then**
    $X' \leftarrow \emptyset$
    $X \leftarrow set(\psi(X), G, \vec{m}')$
    while $X' \neq X$ **do**
        $X' = X$
        $X = set(\psi(X), G, \vec{m}')$
    end while
    return X
end if

□

Observe that the problem is PSPACE-complete when the number of resources is constant.

5 THE OPTIMAL COALITION PROBLEM

In (Della Monica et al., 2011), an optimality problem is introduced, called the *Optimal Coalition problem* (*OC*). This is the problem of finding the coalitions which achieve the given formulas with least cost, if such coalitions exist. Formally, we introduce team variables $Y_1, \ldots, Y_k$ (we use Y to avoid confusion with fixpoint variables), and we admit formulas $\phi(Y_1^{\vec{\$}_1}, \ldots, Y_k^{\vec{\$}_k})$ containing the team variables $Y_1, \ldots, Y_k$ (in place of some of the teams) with the corresponding money endowments $\vec{\$}_1, \ldots, \vec{\$}_k$. We denote by $\phi[Y_1, \ldots, Y_k / A_1, \ldots, A_k]$ the formula in which each team variable Y_i is replaced by the team $A_i \subseteq \mathcal{AG}$. We fix a priced game structure G, a location q of G and an initial global availability $\vec{m}$. The output is a triple $\langle res, A^*, cost \rangle$ where:

- $res \in \{true, false\}$ and $res = true$ iff there is a vector of teams $\langle A_1, \ldots, A_k \rangle$ such that $G, q, \vec{m} \models \phi[Y_1, \ldots, Y_k / A_1, \ldots, A_k]$;
- if $res = true$, A^* is a vector which minimizes the cost (otherwise A^* is undefined);
- $cost = \Sigma_{i=1}^{k} \vec{\$}_i \cdot A_i$ is the cost of the vector of teams, where A_i is the characteristic vector of A_i seen as a subset of $\mathcal{AG}$, and $\cdot$ denotes scalar product between vectors.

We have the following result:

Theorem 5.1. *In* PRB-AMC*, the OC problem is in EXPTIME and it is PSPACE-hard.*

Proof. We check the cost of all possible $(2^n)^k$ vectors of teams by calling each time the model checking algorithm of the previous section. As we have seen, this algorithm is in *EXPTIME*; so also the *OC* problem is.

The *PSPACE*-hardness follows from hardness of the decisional version, and hardness of the latter follows from the proof of Theorem 3.2 of (Della Monica et al., 2011) (again because the PRB-ATL formulas used there actually belong to PRB-AMC). □

6 AN UNDECIDABILITY RESULT

In this section we show the following result:

Theorem 6.1. *It is undecidable whether a formula of* PRB-AMC *is satisfiable in a one point arena (i.e. an arena where Q is a singleton).*

To prove the theorem we reduce to our satisfiability problem a well-known undecidable problem, the solvability of equations $A(\mathbf{n}) = B(\mathbf{n})$, where $\mathbf{n}$ is a vector of variables ranging over $\mathbb{N}$ and A and B are polynomials with coefficients in $\mathbb{N}$, see (Matiyasevich, 1993). In this section we let the letters $m,n,p,\dots$ range over $\mathbb{N}$.

The first step of the reduction, which is standard, is to start from an equation $A(\mathbf{n}) = B(\mathbf{n})$ and to express solvability of the equation via solvability of a finite system $\Sigma(A,B)$ of equations of the form $m = a$ (with $a \in \mathbb{N}$) , $m = n + p$ and $m = n \times p$. The second step is the following lemma:

Lemma 6.1. *Let Σ be a finite system of relations of the form $m = a$ ($a \in \mathbb{N}$), $m = n + p$ and $m = n \times p$, with a set X of unknown variables. Then one can find effectively:*

- *a set R_Σ of resources and a subset Q_Σ of R_Σ*
- *a formula A_Σ of* PRB-AMC *over R_Σ satisfiable in a one point arena*
- *a formula Δ_Σ of* PRB-AMC

such that in every one point model M of A_Σ, Σ holds in M if and only if M verifies $\exists Q_\Sigma \Delta_\Sigma$.

The proof of the lemma is omitted for lack of space and will be provided in a future extended version. Now, the theorem follows from the next Corollary of Lemma 6.1.

Corollary 6.1. *Let Σ be a finite system of equations of the form $m = a$ ($a \in \mathbb{N}$), $m = n + p$ and $m = n \times p$. Then one can find effectively:*

- *a set R_Σ of resources*
- *a formula A_Σ over R_Σ*
- *a formula Δ_Σ of* PRB-AMC

such that Σ is solvable if and only if $A_\Sigma \wedge \Delta_\Sigma$ is satisfiable in a one point arena in PRB-AMC*.*

7 CONCLUSIONS

In this paper, we have presented an extension of μ-calculus, called PRB-AMC, suitable for modeling collective behavior of groups of agents acting in environment where resource availability is limited.

The present work follows previous approaches in that direction (Alechina et al., 2010; Bulling and Farwer, 2010; Della Monica et al., 2011), the main difference being the formalism underlying the logic, namely, the μ-calculus instead of the Alternating-time Temporal Logic. Even though our logic is more expressive than logics introduced in previous work, in particular PRB-ATL, the complexity of both the model checking problem and the optimal coalition problem is not harder than in PRB-ATL, i.e, EXPTIME with PSPACE lower bound. The exact complexity of both problems is conjectured to be EXPTIME-complete. Additionally, we have explored the satisfiability problem for PRB-AMC, proving its undecidability in the particular case when the game structure is an arena with only one state. The satisfiability problem in the general case is an interesting open problem currently under study.

REFERENCES

Alechina, N., Logan, B., Nga, N. H., and Rakib, A. (2009). A logic for coalitions with bounded resources. In *Proc. of the 21st International Joint Conference on Artificial Intelligence*, IJCAI '09, pages 659–664.

Alechina, N., Logan, B., Nga, N. H., and Rakib, A. (2010). 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.

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

Bulling, N. and Dix, J. (2010). Modelling and verifying coalitions using argumentation and ATL. *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial*, 14(46):45–73.

Bulling, N. and Farwer, B. (2010). 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.

Della Monica, D., Napoli, M., and Parente, M. (2011). On a Logic for Coalitional Games with Priced-Resource Agents. Accepted for publication to LAMAS 2011 `http://www.dia.unisa.it/dottorandi/dario.dellamonica/pubs/lamas11.pdf`.

Dunne, P. E., Kraus, S., Manisterski, E., and Wooldridge, M. (2010). Solving coalitional resource games. *Artificial Intelligence*, 174(1):20–50.

Emerson, A. (1996). Model checking and the mu-calculus. In *Descriptive Complexity and Finite Models*, pages 185–214.

Matiyasevich, Y. (1993). *Hilbert's Tenth Problem*. The MIT Press. Foreword by Martin Davis and Hilary Putnam.

Pauly, M. (2002). A modal logic for coalitional power in games. *Journal of Logic and Computation*, 12(1):149–166.

Tarski, A. (1955). A lattice-theoretical fixpoint theorem and its applications. *Pacific Journal of Mathematics*, 5(2):285–309.

Wooldridge, M. and Dunne, P. E. (2006). On the computational complexity of coalitional resource games. *Artificial Intelligence*, 170(10):835–871.