



ELSEVIER

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss



Model-checking for Resource-Bounded ATL with production and consumption of resources [☆]

Natasha Alechina ^{a,*}, Brian Logan ^a, Hoang Nga Nguyen ^b, Franco Raimondi ^c

^a School of Computer Science, The University of Nottingham, UK

^b Centre for Mobility and Transport, Coventry University, UK

^c Department of Computer Science, Middlesex University, UK

ARTICLE INFO

Article history:

Received 19 August 2016

Accepted 19 August 2016

Available online xxxx

Keywords:

Model-checking

Resources

Coalitional ability

Verification of multi-agent systems

ABSTRACT

Several logics for expressing coalitional ability under resource bounds have been proposed and studied in the literature. Previous work has shown that if only consumption of resources is considered or the total amount of resources produced or consumed on any path in the system is bounded, then the model-checking problem for several standard logics, such as Resource-Bounded Coalition Logic (RB-CL) and Resource-Bounded Alternating-Time Temporal Logic (RB-ATL) is decidable. However, for coalition logics with unbounded resource production and consumption, only some undecidability results are known. In this paper, we show that the model-checking problem for RB-ATL with unbounded production and consumption of resources is decidable but EXPSpace-hard. We also investigate some tractable cases and provide a detailed comparison to a variant of the resource logic RAL, together with new complexity results.

© 2017 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Alternating-Time Temporal Logic (ATL) [1] is widely used in verification of multi-agent systems. ATL can express properties related to coalitional ability, for example, one can state that a group of agents A has a strategy (a choice of actions) such that whatever the actions by the agents outside the coalition, any computation of the system generated by the strategy satisfies some temporal property. A number of variations on the semantics of ATL exist: agents may have perfect recall or be memoryless, and they may have full or partial observability. In the case of fully observable models and memoryless agents, the model-checking problem for ATL is polynomial in the size of the model and the formula, while it is undecidable for partially observable models where agents have perfect recall [2]. Additionally, even in the simple case of fully observable models and memoryless agents, the complexity increases substantially if the model-checking problem takes into account models with *compact* (implicit) representations [2].

In this paper, we consider an extension of perfect recall, fully observable ATL where agents produce and consume resources. The properties we are interested in are related to coalitional ability under resource bounds. Instead of asking whether a group of agents has a strategy to enforce a certain temporal property, we are interested in whether the group has a strategy that can be executed under a certain resource bound (e.g., if the agents have at most b_1 units of resource r_1

[☆] This work was supported by the Engineering and Physical Sciences Research Council [grants EP/K033905/1 and EP/K033921/1].

* Corresponding author.

E-mail addresses: nza@cs.nott.ac.uk (N. Alechina), bsl@cs.nott.ac.uk (B. Logan), hoang.nguyen@coventry.ac.uk (H.N. Nguyen), f.aimondi@mdx.ac.uk (F. Raimondi).

<http://dx.doi.org/10.1016/j.jcss.2017.03.008>

0022-0000/© 2017 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

and b_2 units of resource r_2). Clearly, some actions may no longer be used as part of the strategy if their cost exceeds the bound. There are several ways in which the precise notion of the cost of a strategy can be defined. For example, one can define it as the maximal cost of any path (computation of the system) generated by the strategy, where the cost of a path is the sum of resources produced and consumed by actions on the path. We have chosen a different definition which says that a strategy has a cost at most b if for every path generated by the strategy, every *prefix* of the path has cost at most b . This means that a strategy cannot, for example, start with executing an action that consumes more than b resources, and then ‘make up’ for this by executing actions that produce enough resources to bring the total cost of the path under b . It is however possible to first produce enough resources, and then execute an action that costs more than b , so long as the cost of the path is less than b .

There are also many choices for the precise syntax of the logic and the truth definitions of the formulas. For example, in [3] several versions are given, intuitively corresponding to considering resource bounds both on the coalition A and the rest of the agents in the system, considering a fixed resource endowment of A in the initial state which affects their endowment after executing some actions, etc. In this paper we give a precise comparison of our logic with the variants of \mathcal{L}_{RAL} introduced in [3], and in the process solve an open problem stated in [3]. In [4,5] different syntax and semantics are considered, in which the resource endowment of the whole system is taken into account when evaluating a statement concerning a group of agents A . As observed in [3], subtle differences in truth conditions for resource logics result in the difference between decidability and undecidability of the model-checking problem. In [3], the undecidability of several versions of the logics is proved. Recently, even more undecidability results were shown in [6]. The only decidable cases considered in [3] are an extension of Computation Tree Logic (CTL) [7] with resources (essentially one-agent ATL) and the version where on every path only a fixed finite amount of resources can be produced. Similarly, [4] gives a decidable logic, PRB-ATL (Priced Resource-Bounded ATL), where the total amount of resources in the system has a fixed bound. The model-checking algorithm for PRB-ATL runs in time polynomial in the sizes of the model and the formula, and exponential in the number of resources and the size of the representation (if in binary) of the resource bounds. In [5] an EXPTIME lower bound in the number resources and in the size of the representation (if in binary) of the resource bounds is shown.

The structure of this paper is as follows. In sections 2, 3, and 4, we introduce Resource-Bounded ATL with production and consumption of resources, a model-checking algorithm for it, and prove that the model-checking problem is EXPSPACE-hard. This part of the paper extends [8]. In section 5 we discuss two special cases with feasible model-checking, one of them being a generalisation of the model-checking algorithm for (production-free) RB-ATL introduced in [9] to unbounded resources. In section 6 we give a detailed comparison with the logics in [3] and show that for one of them the model-checking problem is decidable, solving an open problem stated in [3].¹

2. Syntax and semantics of RB±ATL

The logic RB-ATL was introduced in [9]. Here we generalise the definitions from [9] to allow for production as well as consumption of resources. To avoid confusion with the consumption-only version of the logic from [9], we refer to RB-ATL with production and consumption of resources as RB±ATL.

Let $Agt = \{a_1, \dots, a_n\}$ be a set of n agents, $Res = \{res_1, \dots, res_r\}$ be a set of r resources, Π be a set of propositions and $B = \mathbb{N}_\infty^r$ be a set of resource bounds where $\mathbb{N}_\infty = \mathbb{N} \cup \{\infty\}$.

Formulas of RB±ATL are defined by the following syntax

$$\phi, \psi ::= p \mid \neg\phi \mid \phi \vee \psi \mid \langle\langle A^b \rangle\rangle \bigcirc \phi \mid \langle\langle A^b \rangle\rangle \square \phi \mid \langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi$$

where $p \in \Pi$ is a proposition, $A \subseteq Agt$, and $b \in B$ is a resource bound. Here, $\langle\langle A^b \rangle\rangle \bigcirc \phi$ means that a coalition A can ensure that the next state satisfies ϕ under resource bound b . $\langle\langle A^b \rangle\rangle \square \phi$ means that A has a strategy to make sure that ϕ is always true, and the cost of this strategy is at most b . Similarly, $\langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi$ means that A has a strategy to enforce ψ while maintaining the truth of ϕ , and the cost of this strategy is at most b .

We extend the definition of a concurrent game structure with resource consumption and production.

Definition 1. A resource-bounded concurrent game structure (RB-CGS) is a tuple $M = (Agt, Res, S, \Pi, \pi, Act, d, c, \delta)$ where:

- Agt is a non-empty set of n agents, Res is a non-empty set of r resources and S is a non-empty set of states;
- Π is a finite set of propositional variables and $\pi : \Pi \rightarrow \wp(S)$ is a truth assignment which associates each proposition in Π with a subset of states where it is true;

¹ Intuitively, the main difference between our logic (with a decidable model-checking problem) and a version of RAL from [3] where the model-checking problem is undecidable under infinite semantics (considering only infinite computations) is that in our logic, each agent always has an option of executing an *idle* action that does not consume any resources. This means that a finite strategy which conforms to a resource bound and enforces a particular outcome can always be extended to an infinite strategy by choosing the *idle* action. The model-checking problem for the same version of RAL but under finite semantics (considering finite computations) turns out also to be decidable, and a model-checking algorithm for it is obtained as an easy adaptation of the model-checking algorithm for our logic.

- Act is a non-empty set of actions which includes *idle*, and $d : S \times Agt \rightarrow \wp(Act) \setminus \{\emptyset\}$ is a function which assigns to each $s \in S$ a non-empty set of actions available to each agent $a \in Agt$. For every $s \in S$ and $a \in Agt$, *idle* $\in d(s, a)$. We denote joint actions by all agents in Agt available at s by $D(s) = d(s, a_1) \times \dots \times d(s, a_n)$;
- $c : S \times Agt \times Act \rightarrow \mathbb{Z}^r$ is a partial function which maps a state s , an agent a and an action $\alpha \in d(s, a)$ to a vector of integers, where the integer in position i indicates consumption or production of resource res_i by the action (positive value for consumption and negative value for production). We stipulate that $c(s, a, idle) = \vec{0}$ for all $s \in S$ and $a \in Agt$, where $\vec{0} = 0^r$;
- $\delta : S \times Act^{|Agt|} \rightarrow S$ is a partial function that maps every $s \in S$ and joint action $\sigma \in D(s)$ to a state resulting from executing σ in s .

Given a RB-CGS M , we denote the set of all infinite sequences of states (infinite computations) by S^ω and the set of non-empty finite sequences (finite computation) of states by S^+ . For a computation $\lambda = s_0s_1\dots \in S^\omega$ we use the notation $\lambda[i] = s_i$ and $\lambda[i, j] = s_i\dots s_j$.

Given a RB-CGS M and a state $s \in S$, a *joint action by a coalition* $A \subseteq Agt$ is a tuple $\sigma = (\sigma_a)_{a \in A}$ (where σ_a is the action that agent a executes as part of σ , the a th component of σ) such that $\sigma_a \in d(s, a)$. The set of all joint actions for A at state s is denoted by $D_A(s)$. Given a joint action by the grand coalition $\sigma \in D(s)$, σ_A (a projection of σ on A) denotes the joint action executed by A as part of σ : $\sigma_A = (\sigma_a)_{a \in A}$. The set of all possible outcomes of a joint action $\sigma \in D_A(s)$ at state s is:

$$out(s, \sigma) = \{s' \in S \mid \exists \sigma' \in D(s) : \sigma = \sigma'_A \wedge s' = \delta(s, \sigma')\}$$

In the sequel, we use the usual point-wise notation for vector comparison and addition. In particular, $(b_1, \dots, b_r) \leq (d_1, \dots, d_r)$ iff $b_i \leq d_i \forall i \in \{1, \dots, r\}$, $(b_1, \dots, b_r) = (d_1, \dots, d_r)$ iff $b_i = d_i \forall i \in \{1, \dots, r\}$, and $(b_1, \dots, b_r) + (d_1, \dots, d_r) = (b_1 + d_1, \dots, b_r + d_r)$. However, for convenience we define $(b_1, \dots, b_r) < (d_1, \dots, d_r)$ as $(b_1, \dots, b_r) \leq (d_1, \dots, d_r)$ and $(b_1, \dots, b_r) \neq (d_1, \dots, d_r)$. We assume that for any $b \in \mathbb{N}$, $b \leq \infty$, $b + \infty = \infty$ and $\infty - b = \infty$. Given a function f returning a vector, we also denote by f_i the function that returns the i -th component of the vector returned by f .

The cost of a joint action $\sigma \in D_A(s)$ is defined as $cost_A(s, \sigma) = \sum_{a \in A} c(s, a, \sigma_a)$ and the subscript A is omitted when $A = Agt$.

Given a RB-CGS M , a *strategy for a coalition* $A \subseteq Agt$ is a mapping $F_A : S^+ \rightarrow Act^{|A|}$ such that, for every $\lambda s \in S^+$, $F_A(\lambda s) \in D_A(s)$. A computation $\lambda \in S^\omega$ is consistent with a strategy F_A iff, for all $i \geq 0$, $\lambda[i+1] \in out(\lambda[i], F_A(\lambda[0, i]))$. We denote by $out(s, F_A)$ the set of all computations λ starting from s that are consistent with F_A .

Given a bound $b \in B$, a computation $\lambda \in out(s, F_A)$ is b -consistent with F_A iff, for every $i \geq 0$,

$$\sum_{j=0}^i cost_A(\lambda[j], F_A(\lambda[0, j])) \leq b$$

Note that this definition implies that the cost of every prefix of the computation is below b .

The set of all computations starting from state s that are b -consistent with F_A is denoted by $out(s, F_A, b)$. F_A is a b -strategy iff $out(s, F_A) = out(s, F_A, b)$ for any state s .

Given a RB-CGS M and a state s of M , the truth of a RB \pm ATL formula ϕ with respect to M and s is defined inductively on the structure of ϕ as follows:

- $M, s \models p$ iff $s \in \pi(p)$;
- $M, s \models \neg\phi$ iff $M, s \not\models \phi$;
- $M, s \models \phi \vee \psi$ iff $M, s \models \phi$ or $M, s \models \psi$;
- $M, s \models \langle\langle A^b \rangle\rangle \bigcirc \phi$ iff $\exists b$ -strategy F_A such that for all $\lambda \in out(s, F_A)$: $M, \lambda[1] \models \phi$;
- $M, s \models \langle\langle A^b \rangle\rangle \square \phi$ iff $\exists b$ -strategy F_A such that for all $\lambda \in out(s, F_A)$ and $i \geq 0$: $M, \lambda[i] \models \phi$; and
- $M, s \models \langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi$ iff $\exists b$ -strategy F_A such that for all $\lambda \in out(s, F_A)$, $\exists i \geq 0$: $M, \lambda[i] \models \psi$ and $M, \lambda[j] \models \phi$ for all $j \in \{0, \dots, i-1\}$.

Since the infinite resource bound version of RB \pm ATL modalities correspond to the standard ATL modalities, we will write $\langle\langle A^\infty \rangle\rangle \bigcirc \phi$, $\langle\langle A^\infty \rangle\rangle \phi \mathcal{U} \psi$, $\langle\langle A^\infty \rangle\rangle \square \phi$ as $\langle\langle A \rangle\rangle \bigcirc \phi$, $\langle\langle A \rangle\rangle \phi \mathcal{U} \psi$, $\langle\langle A \rangle\rangle \square \phi$, respectively. When the context is clear, we will sometimes write $s \models \phi$ instead of $M, s \models \phi$.

Note that although we only consider infinite paths, the condition that the *idle* action is always available and costs $\vec{0}$ makes the model-checking problem easier (we only need to find a strategy with a finite prefix under bound b to satisfy formulas of the form $\langle\langle A^b \rangle\rangle \bigcirc \phi$ and $\langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi$, and then the strategy can make the *idle* choice forever).

As an example of the expressivity of the logic, consider the model in Fig. 1 with two agents a_1 and a_2 and two resources r_1 and r_2 . Let us assume that $c(s_1, a_1, \alpha) = \langle -2, 1 \rangle$ (action α produces 2 units of r_1 and consumes one unit of r_2), $c(s, a_2, \beta) = \langle 1, -1 \rangle$ and $c(s, a_1, \gamma) = \langle 5, 0 \rangle$. Then agent a_1 on its own has a strategy to enforce a state satisfying p under resource bound of 3 units of r_1 and 1 unit of r_2 ($M, s_1 \models \langle\langle \{a_1\}^{(3,1)} \rangle\rangle \top \mathcal{U} p$): a_1 has to select action α in s_1 which requires it to consume one unit of r_2 but produces two units of r_1 , and then action γ in s that requires 5 units of r_1 which is now within the resource bound since the previous action has produced 2 units. All outcomes of this strategy lead to s' where

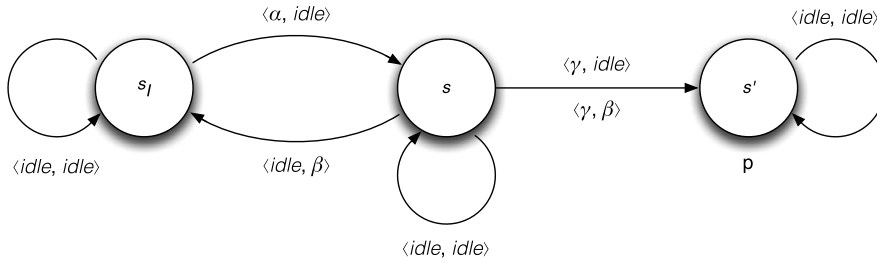


Fig. 1. An example with consumption and production of resources.

p holds. After this, a_1 has to select *idle* forever, which does not require any resources. Any smaller resource bound is not sufficient. However, both agents have a strategy to enforce the same outcome under a smaller resource bound of just one unit of r_2 ($M, s_l \models \langle\langle a_1, a_2 \rangle\rangle^{(0,1)} \top \mathcal{U} p$): agent a_2 needs to select β and a_1 *idle* in s until the agents have gone through the loop between s_l and s four times and accumulated enough of resource r_1 to enable agent a_1 to perform γ in s .

3. Model checking RB \pm ATL

The model-checking problem for RB \pm ATL is the question whether, for a given RB-CGS structure M , a state s in M and an RB \pm ATL formula ϕ_0 , $M, s \models \phi_0$. In this section we prove the following theorem:

Theorem 1. *The model-checking problem for RB \pm ATL is decidable.*

To prove decidability, we give an algorithm which, given a structure $M = (\text{Agt}, \text{Res}, S, \Pi, \pi, \text{Act}, d, c, \delta)$ and a formula ϕ_0 , returns the set of states $[\phi_0]_M$ satisfying ϕ_0 : $[\phi_0]_M = \{s \mid M, s \models \phi_0\}$ (see Algorithm 1).

Algorithm 1 Labelling ϕ_0 .

```

function RB $\pm$ ATL-LABEL( $M, \phi_0$ )
  for  $\phi' \in \text{Sub}(\phi_0)$  do
    case  $\phi' = p, \neg\phi, \phi \wedge \psi, \langle\langle A \rangle\rangle \circ \phi, \langle\langle A \rangle\rangle \phi \mathcal{U} \psi, \langle\langle A \rangle\rangle \square \phi$ 
      standard, see [1]
    case  $\phi' = \langle\langle A^b \rangle\rangle \circ \phi$ 
       $[\phi']_M \leftarrow \text{Pre}(A, [\phi]_M, b)$ 
    case  $\phi' = \langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi$ 
       $[\phi']_M \leftarrow \{s \mid s \in S \wedge$ 
        UNTIL-STRATEGY( $\text{node}_0(s, b), \langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi\}$ 
      }
    case  $\phi' = \langle\langle A^b \rangle\rangle \square \phi$ 
       $[\phi']_M \leftarrow \{s \mid s \in S \wedge \text{BOX-STRATEGY}(\text{node}_0(s, b), \langle\langle A^b \rangle\rangle \square \phi)\}$ 
  return  $[\phi_0]_M$ 

```

Given ϕ_0 , we produce a set of subformulas $\text{Sub}(\phi_0)$ of ϕ_0 in the usual way, however, in addition, if $\langle\langle A^b \rangle\rangle \gamma \in \text{Sub}(\phi_0)$, its infinite resource version $\langle\langle A \rangle\rangle \gamma$ is added to $\text{Sub}(\phi_0)$. $\text{Sub}(\phi_0)$ is ordered in increasing order of complexity, and the infinite resource version of each modal formula comes before the bounded version. Note that if a state s is not annotated with $\langle\langle A \rangle\rangle \gamma$ then s cannot satisfy the bounded resource version $\langle\langle A^b \rangle\rangle \gamma$.

We then proceed by cases. For all formulas in $\text{Sub}(\phi_0)$ apart from $\langle\langle A^b \rangle\rangle \circ \phi$, $\langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi$ and $\langle\langle A^b \rangle\rangle \square \phi$ we essentially run the standard ATL model-checking algorithm [1].

Labelling states with $\langle\langle A^b \rangle\rangle \circ \phi$ makes use of a function $\text{Pre}(A, \rho, b)$ which, given a coalition A , a set $\rho \subseteq S$ and a bound b , returns a set of states s in which A has a joint action σ_A with $\text{cost}(s, \sigma_A) \leq b$ such that $\text{out}(s, \sigma_A) \subseteq \rho$. Labelling states with $\langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi$ and $\langle\langle A^b \rangle\rangle \square \phi$ is more complex, and in the interests of readability we provide separate functions: UNTIL-STRATEGY for $\langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi$ formulas is shown in Algorithm 2, and BOX-STRATEGY for $\langle\langle A^b \rangle\rangle \square \phi$ formulas is shown in Algorithm 3.

Both algorithms proceed by depth-first and-or search of M . We record information about the state of the search in a search tree of nodes. A *node* is a structure which consists of a state of M , the resources available to the agents A in that state (if any), and a finite path of nodes leading to this node from the root node. Edges in the tree correspond to joint actions by all agents. Note that the resources available to the agents in a state s on a path constrain the edges from the corresponding node to be those actions σ_A where $\text{cost}(s, \sigma_A)$ is less than or equal to the available resources. For each node n in the tree, we have a function $s(n)$ which returns its state, $p(n)$ which returns the nodes on the path and $e(n)$ which returns the vector of resource availabilities in $s(n)$ as a result of following $p(n)$. The function $\text{node}_0(s, b)$ returns the root node, i.e., a node n_0 such that $s(n_0) = s$, $p(n_0) = []$ and $e_i(n_0) = b_i$ for all resources i . The function $\text{node}(n, \sigma, s')$ returns a node n' where $s(n') = s'$, $p(n') = [p(n) \cdot n]$ and for all resources i , $e_i(n') = e_i(n) - \text{cost}_i(s(n), \sigma)$.

Both UNTIL-STRATEGY and BOX-STRATEGY take a search tree node n and a formula $\phi' \in \text{Sub}(\phi_0)$ as input, and have similar structure. They first check if the infinite resource version of ϕ' (i.e., ϕ' where the outermost coalition modality has bound ∞)

is false in the state represented by node n , $s(n)$. (Note that the state represented by n has already been labelled by the resource bounded subformulas ϕ and ψ .) If so, they return false immediately, terminating search of the current branch of the search tree (lines 2–3 of Algorithms 2 and 3). UNTIL-STRATEGY also returns true if the second argument ψ of ϕ' is true in $s(n)$ (lines 8–9 of Algorithm 2). Both UNTIL-STRATEGY and BOX-STRATEGY check whether the state $s(n)$ has been encountered before on $p(n)$, i.e., $p(n)$ ends in a loop. In the case of UNTIL-STRATEGY, if the loop is unproductive (i.e., resource availability has not increased since the previous occurrence of $s(n)$ on the path), then the loop is not necessary for a successful strategy, and search on this branch is terminated returning false (lines 4–5). If on the other hand the loop strictly increases the availability of at least one resource i and does not decrease the availability of other resources, then $e_i(n)$ is replaced with ∞ , as a shorthand denoting that any finite amount of i can be produced by repeating the loop sufficiently many times (lines 6–7 of Algorithm 2). If all resource values have been replaced by ∞ , UNTIL-STRATEGY returns true (lines 10–11), since the branch satisfies the infinite resource version $\langle\langle A \rangle\rangle\phi\mathcal{U}\psi$ of ϕ' , and an arbitrary amount of any resource can be accumulated along the path. For BOX-STRATEGY the loop check is slightly different. If the loop decreases the amount of at least one resource without increasing the availability of any other resource, it cannot form part of a successful strategy, and the search terminates returning false (lines 4–5 of Algorithm 3). If a non-decreasing loop is found, then it is possible to maintain the invariant formula ϕ forever without expending any resources, and the search terminates returning true (lines 6–7).

Algorithm 2 Labelling $\langle\langle A^b \rangle\rangle\phi\mathcal{U}\psi$.

```

1: function UNTIL-STRATEGY( $n, \langle\langle A^b \rangle\rangle\phi\mathcal{U}\psi$ )
2:   if  $s(n) \not\models \langle\langle A^\infty \rangle\rangle\phi\mathcal{U}\psi$  then
3:     return false
4:   if  $\exists n' \in p(n) : s(n') = s(n) \wedge (\forall j : e_j(n') \geq e_j(n))$  then
5:     return false
6:   for  $i \in \{i \in \text{Res} \mid \exists n' \in p(n) : s(n') = s(n) \wedge (\forall j : e_j(n') \leq e_j(n)) \wedge$ 
    $e_i(n') < e_i(n)\}$  do
7:      $e_i(n) \leftarrow \infty$ 
8:   if  $s(n) \models \psi$  then
9:     return true
10:  if  $e(n) = \infty$  then
11:    return true
12:   $\text{Act}A \leftarrow \{\sigma \in D_A(s(n)) \mid \text{cost}(s(n), \sigma) \leq e(n)\}$ 
13:  for  $\sigma \in \text{Act}A$  do
14:     $O \leftarrow \text{out}(s(n), \sigma)$ 
15:     $\text{strat} \leftarrow \text{true}$ 
16:    for  $s' \in O$  do
17:       $\text{strat} \leftarrow \text{strat} \wedge$ 
18:        UNTIL-STRATEGY( $\text{node}(n, \sigma, s'), \langle\langle A^b \rangle\rangle\phi\mathcal{U}\psi$ )
19:    if  $\text{strat}$  then
20:      return true
21:  return false

```

If none of the if statements evaluates to true, then, in both UNTIL-STRATEGY and BOX-STRATEGY, search continues by considering each action available at $s(n)$ in turn. For each action $\sigma \in \text{Act}A$, the algorithm checks whether a recursive call of the algorithm returns true in all outcome states of σ (i.e., σ is part of a successful strategy). If such a σ is found, the algorithm returns true. Otherwise the algorithm returns false. Note that the argument ϕ' is passed through the recursive calls unchanged: information about the resources available to the agents in $s(n)$ as a result of following $p(n)$ is encoded in the search nodes.

Lemma 1. Algorithm 1 terminates.

Proof. All the cases in Algorithm 1 apart from $\langle\langle A^b \rangle\rangle\phi\mathcal{U}\psi$ and $\langle\langle A^b \rangle\rangle\Box\phi$ can be computed in time polynomial in $|M|$ and $|\phi|$. The cases for $\langle\langle A^b \rangle\rangle\phi\mathcal{U}\psi$ and $\langle\langle A^b \rangle\rangle\Box\phi$ involve calling the UNTIL-STRATEGY and BOX-STRATEGY procedures, respectively, for every state in S . In order to prove that these procedures terminate, we are going to show that there is no infinite sequence of calls to UNTIL-STRATEGY or BOX-STRATEGY.

Assume to the contrary that n_1, n_2, \dots is an infinite sequence of nodes in an infinite sequence of recursive calls to UNTIL-STRATEGY or BOX-STRATEGY. Then, since the set of states is finite, there is an infinite subsequence n_{i_1}, n_{i_2}, \dots of n_1, n_2, \dots such that for all j , $s(n_{i_j}) = s$ for some state s (the state is the same for all the nodes in the subsequence). We show that then there is an infinite subsequence n'_1, n'_2, \dots of n_{i_1}, n_{i_2}, \dots such that for $k < j$, $e(n'_k) \leq e(n'_j)$. The proof is very similar to the proof of Lemma f in [10, p. 70] and proceeds by induction on the number of resources r . For $r = 1$, since $e(n)$ is always positive, the claim is immediate. Assume the lemma holds for r and let us show it for $r + 1$. Then there is an infinite subsequence m'_1, m'_2, \dots of n_{i_1}, n_{i_2}, \dots where for all resources $i \in \{1, \dots, r\}$ $e_i(m'_k) \leq e_i(m'_j)$ for $k < j$. Clearly there are two nodes m'_{j_1} and m'_{j_2} in this sequence such that $e_{r+1}(m'_{j_1}) \leq e_{r+1}(m'_{j_2})$ (since there are only finitely many positive integers

Algorithm 3 Labelling $\langle\langle A^b \rangle\rangle \square \phi$.

```

1: function BOX-STRATEGY( $n, \langle\langle A^b \rangle\rangle \square \phi$ )
2:   if  $s(n) \not\models \langle\langle A^\infty \rangle\rangle \square \phi$  then
3:     return false
4:   if  $\exists n' \in p(n) : s(n') = s(n) \wedge (\forall j : e_j(n') \geq e_j(n)) \wedge$ 
       $(\exists j : e_j(n') > e_j(n))$  then
5:     return false
6:   if  $\exists n' \in p(n) : s(n') = s(n) \wedge (\forall j : e_j(n') \leq e_j(n))$  then
7:     return true
8:    $Act_A \leftarrow \{\sigma \in D_A(s(n)) \mid cost(s(n), \sigma) \leq e(n)\}$ 
9:   for  $\sigma \in Act_A$  do
10:     $O \leftarrow out(s(n), \sigma)$ 
11:     $strat \leftarrow true$ 
12:    for  $s' \in O$  do
13:       $strat \leftarrow strat \wedge$ 
14:        BOX-STRATEGY( $node(n, \sigma, s'), \langle\langle A^b \rangle\rangle \square \phi$ )
15:    if  $strat$  then
16:      return true
17:   return false

```

which are smaller than $e_{r+1}(m'_{j_1})$). Hence $e(m'_{j_1}) \leq e(m'_{j_2})$. The same argument can be repeated to show that there is a node m'_{j_3} in the sequence such that $e(m'_{j_2}) \leq e(m'_{j_3})$, etc. Hence, if there is an infinite sequence of nodes corresponding to an infinite sequence of recursive calls, then there is an infinite subsequence of that sequence of nodes, where the nodes in the subsequence have the same state and the same or growing resource availability. The existence of the latter subsequence would constitute a contradiction, because comparable resource availability vectors for nodes with the same state will lead to termination after finitely many steps (so there cannot be an infinite sequence of recursive calls). To see why this is so, consider the simpler case of BOX-STRATEGY first. In Algorithm 3, when in a node n we discover that previously we encountered a node n' such that $s(n) = s(n')$ and $e(n') \leq e(n)$, we return true. So the BOX-STRATEGY will terminate after encountering just one pair of nodes with the same state where the second node has the same or higher resource availability. Hence, there cannot be an infinite sequence of calls to the BOX-STRATEGY. Consider UNTIL-STRATEGY. Given a subsequence n_1, n_2, \dots, n_r (where r is the number of resource types) of nodes with the same state such that $e(n_i) \leq e(n_{i+1})$, either $e(n_i) = e(n_{i+1})$ (and Algorithm 2 returns false) or $e(n_i) \leq e(n_{i+1})$ with $e_j(n_i) < e_j(n_{i+1})$ for some j and one or more resource types are reset to ∞ . When all resource types are reset to ∞ , Algorithm 2 returns true. In either case, the existence of such a subsequence implies termination after finitely many steps, which contradicts our original assumption that there is an infinite sequence of recursive calls. Hence, Algorithm 1 terminates. \square

Before we prove correctness of UNTIL-STRATEGY and BOX-STRATEGY, we need some auxiliary notions. Let n be a node where one of the procedures returns true. We will refer to $tree(n)$ as the tree representing the successful call to the procedure. In particular, if the procedure returns true before any recursive call is made, then $tree(n) = n$. Otherwise the procedure returns true because there is an action $\alpha \in Act_A$ such that for all $s' \in out(s(n), \alpha)$ the procedure returns true in $n' = node(n, \alpha, s')$. In this case, $tree(n)$ has n as its root and trees $tree(n')$ are the children of n . We refer to the action α as n_{act} (the action that generates the children of n). For the sake of uniformity, if $tree(n) = n$ then we set n_{act} to be *idle*. Such a tree corresponds to a strategy F where for each path $n \cdots m$ from the root n to a node m in $tree(n)$, $F(s(n) \cdots s(m)) = m_{act}$.

A strategy F for satisfying $\langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi$ is \mathcal{U} -economical for a node n if, intuitively, no path generated by it contains a loop that does not increase any resource. A strategy is \square -economical for a node n if, intuitively, no path generated by it contains a loop that decreases some resources and does not increase any other resources. Formally, a strategy F is \mathcal{U} -economical for n if

- F satisfies $\langle\langle A^{e(n)} \rangle\rangle \phi \mathcal{U} \psi$ at $s(n)$, i.e., F is a $e(n)$ -strategy and $\forall \lambda \in out(s(n), F)$, $\exists i \geq 0 : \lambda[i] \models \psi$ and $\lambda[j] \models \phi$ for all $j \in \{0, \dots, i\}$;
- The path $p(n) \cdot n$ is already \mathcal{U} -economical, i.e., $\forall n' \in p(n) \cdot n, n'' \in p(n') : s(n'') = s(n') \Rightarrow e(n'') \not\geq e(n')$;
- Every state is reached by F \mathcal{U} -economically, i.e., for each computation $s_0 s_1 \dots s_k \dots \in out(s(n), F)$ and each $j < k \leq i$ where i is the smaller index such that s_i satisfies ψ , $s_j = s_k \Rightarrow cost(s_j \dots s_k) \not\geq \bar{0}$ with $cost(s_j \dots s_k) = \sum_{l=j, \dots, k-1} cost(\lambda[l], F(\lambda[0, l]))$; and
- Every state is reached by F \mathcal{U} -economically with respect to the path $p(n)$, i.e., for every computation $s_0 s_1 \dots s_k \dots \in out(s(n), F)$, $\forall n' \in p(n) : s(n') = s_k \Rightarrow e(n') \not\geq e(n) - cost(s_0 \dots s_k)$.

A strategy F is \square -economical if:

- F satisfies $\langle\langle A^{e(n)} \rangle\rangle \square \phi$ at $s(n)$, i.e., F is a $e(n)$ -strategy and $\forall \lambda \in out(s(n), F)$, $\forall i \geq 0 : \lambda[i] \models \phi$;
- The path $p(n) \cdot n$ is already \square -economical, i.e., $\forall n' \in p(n) \cdot n, n'' \in p(n') : s(n'') = s(n') \Rightarrow e(n'') \not\geq e(n')$;

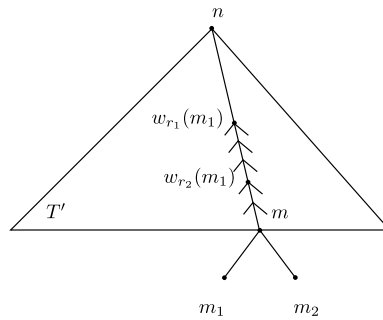


Fig. 2. Tree T and $T' = \text{prune}(T, m)$.

- Every state is reached by F \square -economically, i.e., for each computation $s_0 s_1 \dots s_k \dots \in \text{out}(s(n), F) \forall j < k : s_j = s_k \Rightarrow \text{cost}(s_j \dots s_k) \not\leq 0$;
- Every state is reached by F \square -economically with respect to the path $p(n)$, i.e., for every computation $s_0 s_1 \dots s_k \dots \in \text{out}(s(n), F), \forall n' \in p(n) : s(n') = s_k \Rightarrow e(n') \not\leq e(n) - \text{cost}(s_0 \dots s_k)$.

Note that any strategy F satisfying $\langle\langle A^{e(n)} \rangle\rangle \phi \mathcal{U} \psi$ ($\langle\langle A^{e(n)} \rangle\rangle \square \phi$) at $s(n)$ can be converted to an economical one by eliminating unproductive loops, as stated by the following proposition.

Proposition 1. *There is a strategy to satisfy $\langle\langle A^{e(n)} \rangle\rangle \phi \mathcal{U} \psi$ (resp., $\langle\langle A^{e(n)} \rangle\rangle \square \phi$) at $s(n)$ iff there is an \mathcal{U} -economical (resp., \square -economical) strategy to satisfy $\langle\langle A^{e(n)} \rangle\rangle \phi \mathcal{U} \psi$ (resp., $\langle\langle A^{e(n)} \rangle\rangle \square \phi$) at $s(n)$.*

Next we prove correctness of UNTIL-STRATEGY. The next lemma essentially shows that replacing a resource value with ∞ in Algorithm 2 is harmless. For the inductive step of the proof, we need the following notion. Given a tree $tree(n)$, we call its pruning, denoted as $\text{prune}(tree(n), m_1, \dots, m_k)$, the tree obtained by removing all children of some nodes m_1, \dots, m_k that have only leaves as children in $tree(n)$.

Lemma 2. *Let $n = \text{node}_0(s, b)$ be a node where UNTIL-STRATEGY returns true. Let f be a function that for each leaf n' of $tree(n)$ returns $f(n') \in \mathbb{N}^r$ such that $f_i(n') = e_i(n')$ if $e_i(n') \neq \infty$ ($f_i(n')$ can be any natural number if $e_i(n') = \infty$). Then, there is a strategy F such that for every leaf n' of the tree induced by F , $e(n') \geq f(n')$ holds.*

Proof. By induction on the structure of $tree(n)$.

Base Case: Let $tree(n)$ contain only its root. The proof is obvious for any strategy.

Inductive Step: Let us consider a pruning T of $tree(n)$. By the induction hypothesis, any tree T' that has a less complex structure than T has a strategy to generate at least $f(n') \in \mathbb{N}^r \leq e(n')$ for all leaves n' of T' .

In the following, given nodes n, n_1, \dots, n_k , we denote by $n(n_1, \dots, n_k)$ the depth-1 tree which has n as its root and n_1, \dots, n_k as the immediate leaves of n .

Let $m(m_1, \dots, m_k)$ be an arbitrary depth-1 sub-tree of T (see Fig. 2). By removing m_1, \dots, m_k from T , we obtain a pruning T' of T .

Let $n \dots m \cdot m_i$ be a path in T from the root n to one of the leaves m_i . For each resource r the availability of which turns to ∞ at m_i , there must be a node, denoted by $w_r(m_i)$, in the path $n \dots m \cdot m_i$ which is used to turn the availability of r to ∞ at m_i , that is, $w_r(m_i)$ is such that $s(w_r(m_i)) = s(m_i)$, $e_i(w_r(m_i)) \leq e_i(m_i)$ for each i , and $e_r(w_r(m_i)) < e_r(m_i)$. We may repeat the path from $w_r(m_i)$ to m_i several times to generate enough resource availability for r . We call the path from $w_r(m_i)$ to m_i together with all the immediate child nodes of those along the path the column graph from $w_r(m_i)$ to m_i . Each time, an amount of $g_r = e_r(m) - \text{cost}_r(m_{\text{act}}) - e_r(w_r(m_i))$ is generated. Then, the minimal number of times to repeat the path from $w_r(m_i)$ to m_i is $h_r(m_i) = \lceil \frac{f_r(m_i) - (e_r(m) - \text{cost}_r(m_{\text{act}}))}{g_r} \rceil$.

Note that we need to repeat at each m_i for each resource r the path from $w_r(m_i)$ to m_i $h_r(m_i)$ times. To record the number of times the path has been repeated, we attach to each m_i a counter $\hat{h}_r(m_i)$ for each r and write the new node of m_i as $m_i^{\hat{h}_r(m_i)}$.

Initially, $\hat{h}_r(m_i) = 0$ for all r and for all nodes m_i . A step (see Fig. 3) of the repetition is done as follows: let $m_i^{\hat{h}_r(m_i)}$ be some node such that $\hat{h}_r(m_i) < h_r(m_i)$. Let $m_j^{\hat{h}_r(m_j)}$ be the sibling of $m_i^{\hat{h}_r(m_i)}$ ($j \neq i$). We extend from $m_i^{\hat{h}_r(m_i)}$ the column graph from $w_r(m_i)$ to m_i ; each new m_j ($j \neq i$) is annotated with $\hat{h}_r(m_j)$ (same as before) and the new m_i is annotated with $\hat{h}_r(m_i)$ except that $\hat{h}_r(m_i)$ is increased by 1. We repeat the above step until $\hat{h}_r(m_i) = h_r(m_i)$ (it must terminate due to the fact that $h_r(m_i) < \infty$ for all r and m_i).

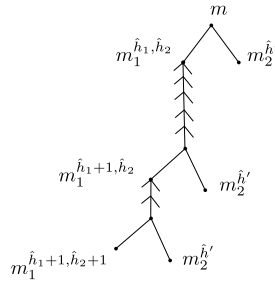


Fig. 3. Repeating steps to generate resources.

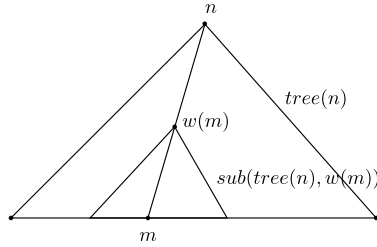


Fig. 4. $w(m)$ of m in $tree(n)$.

At the end, we obtain a tree where all leaves $m_i^{\hat{h}(m_i)}$ have $\hat{h}_r(m_i) = h_r(m_i)$ for all r , hence the availability of r is at least f_r . Let $E(m)$ be the extended tree from m .

Let $F_{T'}$ be the strategy generated by T' . We extend $F_{T'}$ with $E(m)$ for every occurrence of m' in $F_{T'}$ such that $s(m') = s(m)$, and denote this extended strategy $F_{T'}^E$. For all leaves m' in $E(m)$ other than m_i , let $sub(T, m')$ be some sub-tree of T starting from a node m'' such that $s(m'') = s(m')$. Then, we extend $F_{T'}^E$ with $sub(T, m')$ for every occurrence of m' in $F_{T'}^E$. We finally obtain a tree F_T which satisfies the condition that all leaves l have resource availability of at least $f(l)$. \square

Corollary 1. If $UNTIL-STRATEGY(node_0(s, b), \langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi)$ returns true then $s \models \langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi$.

Lemma 3. If $UNTIL-STRATEGY(n, \langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi)$ returns false, then there is no strategy satisfying $\langle\langle A^{e(n)} \rangle\rangle \phi \mathcal{U} \psi$ from $s(n)$ that is \mathcal{U} -economical for n .

Proof. We prove the lemma by induction on the height in the recursion tree of $UNTIL-STRATEGY$ calls.

Base Case: If false is returned by the first if-statement, then $s(n) \not\models \langle\langle A \rangle\rangle \phi \mathcal{U} \psi$; this also means there is no strategy satisfying $\langle\langle A^{e(n)} \rangle\rangle \phi \mathcal{U} \psi$ from $s(n)$.

If false is returned by the second if-statement, then any strategy satisfying $\langle\langle A^{e(n)} \rangle\rangle \phi \mathcal{U} \psi$ from $s(n)$ is not economical.

Inductive Step: If false is not returned by the first two if-statements, then, for all actions $\sigma \in ActA$, there exists $s' \in out(s(n), \sigma)$ such that $UNTIL-STRATEGY(n', \langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi)$ (where $n' = node(n, \sigma, s')$) returns false. By induction hypothesis, there is no strategy satisfying $\langle\langle A^{e(n')} \rangle\rangle \phi \mathcal{U} \psi$ from $s(n')$ that is \mathcal{U} -economical for n' . Assume to the contrary that there is a strategy satisfying $\langle\langle A^{e(n')} \rangle\rangle \phi \mathcal{U} \psi$ from $s(n')$ that is \mathcal{U} -economical for n' . Let $\sigma = F(s(n))$, then $\sigma \in ActA$. Obviously, for all $s' \in out(s(n), \sigma)$, $F'(\lambda) = F(s(n)\lambda)$ is an economical strategy from $n' = node(n, \sigma, s')$. This is a contradiction; hence, there is no strategy satisfying $\langle\langle A^{e(n)} \rangle\rangle \phi \mathcal{U} \psi$ from $s(n)$ that is \mathcal{U} -economical for n . \square

Corollary 2. If $UNTIL-STRATEGY(node_0(s, b), \langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi)$ returns false then $s \not\models \langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi$.

Now we turn to Algorithm 3 for labelling states with $\langle\langle A^b \rangle\rangle \square \phi$. First we show its soundness.

Lemma 4. Let $n = node_0(s, b)$. If $BOX-STRATEGY(n, \langle\langle A^b \rangle\rangle \square \phi)$ returns true then $s(n) \models \langle\langle A^b \rangle\rangle \square \phi$.

Proof. Recall that, for each node m in $tree(n)$, we denote by $sub(tree(n), m)$ the sub-tree of $tree(n)$ rooted at m . For each leaf m of $tree(n)$, let $w(m)$ denote one of the nodes in $p(m)$ such that $s(w(m)) = s(m)$ and $e(w(m)) \leq e(m)$ (see Fig. 4).

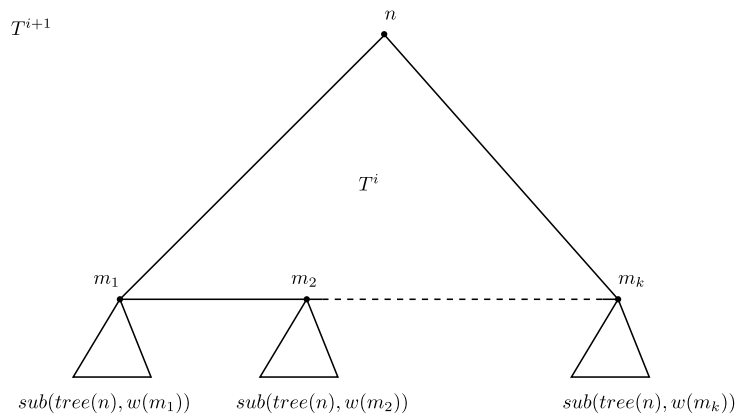


Fig. 5. One step in constructing the strategy.

Let us expand $tree(n)$ as follows:

- T^0 is $tree(n)$;
- T^{i+1} is T^i where all its leaves m are replaced by $sub(tree(n), w(m))$ (see Fig. 5).

Let $T = \lim_{i \rightarrow \infty} T^i$, then T is a strategy for $\langle\langle A^b \rangle\rangle \Box \phi$. \square

Lemma 5. If $\text{BOX-STRATEGY}(n, \langle\langle A^b \rangle\rangle \Box \phi)$ returns false, then there is no strategy satisfying $\langle\langle A^{e(n)} \rangle\rangle \Box \phi$ from $s(n)$ that is \Box -economical for n .

Proof. We prove the lemma by induction on the height in the recursion tree of BOX-STRATEGY calls.

Base Case: If false is returned by the first if-statement, then $s(n) \not\models \langle\langle A \rangle\rangle \Box \phi$; this also means there is no strategy satisfying $\langle\langle A^{e(n)} \rangle\rangle \Box \phi$ at $s(n)$.

If false is returned by the second if-statement, then any strategy satisfying $\langle\langle A^{e(n)} \rangle\rangle \Box \phi$ at $s(n)$ is not \Box -economical.

Inductive Step: If false is not returned by the first two if-statements, for all actions $\sigma \in \text{Act}A$, there exists $s' \in \text{out}(s(n), \sigma)$ such that $\text{BOX-STRATEGY}(n', \langle\langle A^b \rangle\rangle \Box \phi)$ (where $n' = \text{node}(n, \sigma, s')$) returns false. Assume to the contrary that there is a strategy F satisfying $\langle\langle A^{e(n)} \rangle\rangle \Box \phi$ from $s(n)$ that is \Box -economical for n . Let $\sigma = F(s(n))$, then $\sigma \in \text{Act}A$. Obviously, for all $s' \in \text{out}(s(n), \sigma)$, $F'(\lambda) = F(s(n)\lambda)$ is a strategy \Box -economical for $n' = \text{node}(n, \sigma, s')$. This is a contradiction; hence, there is no strategy satisfying $\langle\langle A^{e(n)} \rangle\rangle \Box \phi$ from $s(n)$ that is \Box -economical for n . \square

Then, we have the following result directly:

Corollary 3. If $\text{BOX-STRATEGY}(\text{node}_0(s, b), \langle\langle A^b \rangle\rangle \Box \phi)$ returns false then $s \not\models \langle\langle A^b \rangle\rangle \Box \phi$.

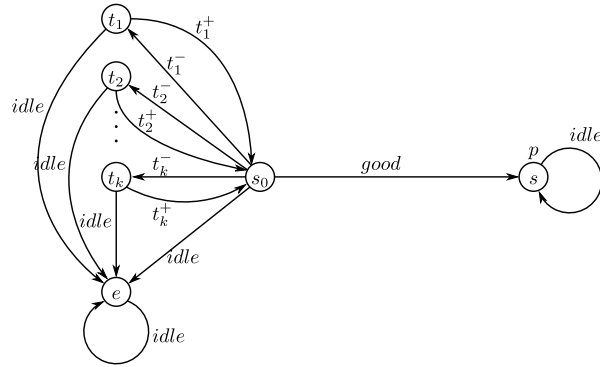
4. Lower bound

In this section we show that the lower bound complexity for the model-checking problem for $\text{RB}\pm\text{ATL}$ is EXPSPACE , by reducing from the reachability problem of Petri Nets. Note that the exact complexity of the reachability problem of Petri Nets is still an open question (although it is known to be decidable and EXPSPACE -hard, [10]). The exact complexity of the $\text{RB}\pm\text{ATL}$ model-checking problem is also unknown. Note that an upper bound for the $\text{RB}\pm\text{ATL}$ model-checking problem would also be an upper bound for the reachability problem of Petri Nets due to the reduction below. Even an Ackermannian upper bound for this problem is still open [11]. This suggests that determining an upper bound for the $\text{RB}\pm\text{ATL}$ model-checking problem is also a hard problem.

A Petri net is a tuple $N = (P, T, W, M)$ where:

- P is a finite set of places;
- T is a finite set of transitions;
- $W : P \times T \cup T \times P \rightarrow \mathbb{N}$ is a weighting function; and
- $M : P \rightarrow \mathbb{N}$ is an initial marking.

A transition $t \in T$ is M -enabled iff $W(r, t) \leq M(r)$ for all $r \in P$. The result of performing t is a marking M' where $M'(r) = M(r) - W(r, t) + W(t, r)$, denoted as $M[t] M'$.

Fig. 6. Structure $I_{N,M'}$.

A marking M' is reachable from M iff there exists a sequence

$$M_0 [t_1] M_1 [t_2] \dots [t_n] M_n$$

where $M_0 = M$ and $n \geq 0$ such that $M_n \geq M'$ (where $M \geq M'$ iff $M(r) \geq M'(r)$ for all $r \in P$). It is known that the lower bound for the complexity of this version of the reachability problem (with $M_n \geq M'$ rather than $M_n = M'$) is EXPSPACE [10, p. 73].

We present a reduction from an instance of the reachability problem of Petri Nets to an instance of the model-checking problem of RB \pm ATL.

Given a net $N = (P, T, W, M)$ and a marking M' , we construct a RB-CGS $I_{N,M'} = (\{1\}, P, S, \{p\}, \pi, Act, d, c, \delta)$ (see Fig. 6) where:

- $S = \{s_0\} \cup T \cup \{s, e\}$;
- $\pi(p) = \{s\}$;
- $Act = \{idle, good\} \cup \{t^-, t^+ \mid t \in T\}$;
- $d(s_0, 1) = \{idle, good\} \cup \{t^- \mid t \in T\}$;
- $d(s, 1) = d(e, 1) = \{idle\}$;
- $d(t, 1) = \{idle, t^+\}$ for all $t \in T$;
- $c(x, 1, idle) = \bar{0}$ for all $x \in S$;
- $c(s_0, 1, good) = M'$;
- $c_r(s_0, 1, t^-) = W(r, t)$ for all $r \in P$ and $t \in T$;
- $c_r(t, 1, t^+) = -W(t, r)$ for all $r \in P$ and $t \in T$;
- $\delta(x, idle) = e$ for all $x \in S \setminus \{s\}$;
- $\delta(s, idle) = s$;
- $\delta(s_0, good) = s$;
- $\delta(s_0, t^-) = t$ for all $t \in T$;
- $\delta(t, t^+) = s_0$ for all $t \in T$.

The following is straightforward:

Lemma 6. Given a net $N = (P, T, W, M)$ and a marking M' , M' is reachable from M iff $I_{N,M'}, s_0 \models \langle\langle 1^M \rangle\rangle \top \mathcal{U} p$.

Proof. (\Rightarrow): Assume that M' is reachable from M , then there exists a sequence

$$M_0 [t_1] M_1 [t_2] \dots [t_n] M_n$$

where $M_0 = M$ and $n \geq 0$ such that $M_n \geq M'$.

Then, we consider the following strategy F for agent 1:

- $F(s_0) = t_1^-$, note that $M \geq c(s_0, 1, t_1^-)$ and $\delta(s_0, t_1^-) = t_1$;
- $F(s_0 t_1) = t_1^+$, note that $M - (c(s_0, 1, t_1^-) + c(t_1, 1, t_1^+)) = M_1 \geq \bar{0}$ and $\delta(t_1, t_1^+) = s_0$;
- $F(s_0 t_1 s_0) = t_2^-$, note that $c(s_0, 1, t_2^-) \leq M_1$, $M - (c(s_0, 1, t_1^-) + c(t_1, 1, t_1^+) + c(s_0, 1, t_2^-)) = M_1 - c(s_0, 1, t_2^-) \geq \bar{0}$ and $\delta(s_0, t_2^-) = t_2$;
- $F(s_0 t_1 s_0 t_2) = t_2^+$, note that $M - (c(s_0, 1, t_1^-) + c(t_1, 1, t_1^+) + c(s_0, 1, t_2^-) + c(t_2, 1, t_2^+)) = M_2 \geq \bar{0}$ and $\delta(t_2, t_2^+) = s_0$;
- \vdots

- $F(s_0 t_1 s_0 t_2 \dots s_0 t_n) = t_n^+$, note that $M - (c(s_0, 1, t_1^-) + c(t_1, 1, t_1^+) + c(s_0, 1, t_2^-) + c(t_2, 1, t_2^+) + \dots + c(s_0, 1, t_n^-) + c(t_n, 1, t_n^+)) = M_n \geq M' \geq \bar{0}$ and $\delta(t_n, t_n^+) = s_0$; and
- $F(s_0 t_1 s_0 t_2 \dots s_0 t_n s_0) = \text{good}$, note that $c(s_0, 1, \text{good}) = M'$, $M - (c(s_0, 1, t_1^-) + c(t_1, 1, t_1^+) + c(s_0, 1, t_2^-) + c(t_2, 1, t_2^+) + \dots + c(s_0, 1, t_n^-) + c(t_n, 1, t_n^+) + c(s_0, 1, \text{good})) = M_n - M' \geq \bar{0}$ and $\delta(s_0, \text{good}) = s$.

Since $s \models p$, it is straightforward that F is a strategy satisfying $\langle\langle 1^M \rangle\rangle \top \mathcal{U} p$ from s_0 .

(\Leftarrow): Assume that $s_0 \models \langle\langle 1^M \rangle\rangle \top \mathcal{U} p$, then there exists a strategy F which satisfies $\langle\langle 1^M \rangle\rangle \top \mathcal{U} p$ from s_0 .

Since there is only one agent, $\text{out}(s_0, F)$ contains a single path $s_0 \dots s \dots$. Obviously, e cannot be visited on the prefix $s_0 \dots s$; hence $s_0 \dots s$ must have the form $s_0 t_1 s_0 t_2 \dots t_n s_0 s$ for some $t_1, \dots, t_n \in T$. Furthermore,

- $F(s_0) = t_1^-, c(s_0, 1, t_1^-) \leq M$,
- $F(s_0 t_1) = t_1^+, c(s_0, 1, t_1^-) + c(t_1, 1, t_1^+) \leq M$,
- \vdots
- $F(s_0 t_1 \dots t_{n-1} s_0) = t_n^-, c(s_0, 1, t_1^-) + c(t_1, 1, t_1^+) + \dots + c(s_0, 1, t_n^-) \leq M$,
- $F(s_0 t_1 \dots t_{n-1} s_0 t_n) = t_n^+, c(s_0, 1, t_1^-) + c(t_1, 1, t_1^+) + \dots + c(s_0, 1, t_n^-) + c(t_n, 1, t_n^+) \leq M$, and
- $F(s_0 t_1 \dots t_{n-1} s_0 t_n s_0) = \text{good}$, $c(s_0, 1, t_1^-) + c(t_1, 1, t_1^+) + \dots + c(s_0, 1, t_n^-) + c(t_n, 1, t_n^+) + M' \leq M$.

Therefore,

- t_1 is M -enabled, let $M_1 = M - (c(s_0, 1, t_1^-) + c(t_1, 1, t_1^+))$,
- t_2 is M_1 -enabled, let $M_2 = M_1 - (c(s_0, 1, t_2^-) + c(t_2, 1, t_2^+)) = M - (c(s_0, 1, t_1^-) + c(t_1, 1, t_1^+) + c(s_0, 1, t_2^-) + c(t_2, 1, t_2^+))$,
- \vdots
- t_n is M_{n-1} -enabled, let $M_n = M_{n-1} - (c(s_0, 1, t_n^-) + c(t_n, 1, t_n^+)) = M - (c(s_0, 1, t_1^-) + c(t_1, 1, t_1^+) + \dots + c(s_0, 1, t_n^-) + c(t_n, 1, t_n^+)) \geq M'$.

Hence, we have $M[t_1]M_1[t_2] \dots [t_n]M_n$. As $M_n \geq M'$, M' is reachable from M . \square

We have the following result:

Corollary 4. *The lower bound for the model-checking problem complexity of $\text{RB}\pm\text{ATL}$ is EXPSpace.*

5. Feasible cases

In the previous section, we have seen that the model-checking problem for $\text{RB}\pm\text{ATL}$ is EXPSpace-hard. There are, however, several tractable special cases of the model-checking problem. Here we consider two of them: model-checking $\text{RB}\pm\text{ATL}$ with a single resource, and model-checking RB-ATL ($\text{RB}\pm\text{ATL}$ with only consumption of resources).

5.1. Model-checking $\text{RB}\pm\text{ATL}$ with a single resource

For the case when $|\text{Res}| = 1$, the problem whether $M, s \models \phi_0$ is decidable in PSPACE.

Theorem 2. *The upper bound for the model-checking problem complexity of $\text{RB}\pm\text{ATL}$ with a single resource is PSPACE.*

Proof. All the cases in Algorithm 1 apart from $\langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi$ and $\langle\langle A^b \rangle\rangle \square \phi$ can be computed in time polynomial in $|M|$ and $|\phi|$. The cases for $\langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi$ and $\langle\langle A^b \rangle\rangle \square \phi$ are more computationally expensive. They involve calling the UNTIL-STRATEGY and the BOX-STRATEGY procedures, respectively, for every state in S . The procedures explore the model in a depth-first manner, one path at a time. Their space requirement corresponds to the maximal length of such a path. Note that unlike depth-first search, UNTIL-STRATEGY and BOX-STRATEGY in the general case (multiple resources) do not terminate when they encounter a loop, that is a path containing two nodes with the same state: \dots, n_1, \dots, n_2 where $s(n_1) = s(n_2)$, since in the general case $e(n_1)$ and $e(n_2)$ may be incomparable. However, for a single resource, it will always be the case that either $e(n_1) = e(n_2)$, or $e(n_1) < e(n_2)$, or $e(n_1) > e(n_2)$. Inspection of UNTIL-STRATEGY and BOX-STRATEGY shows that they will return in all of these cases. Hence, we never need to keep a stack of more than $|S|$ nodes, which requires polynomial space. \square

The result above can be generalised to the case when $|\text{Res}| > 1$, but the formula ϕ_0 is of a special form, where at most one resource is non- ∞ in each bound. To be precise, ϕ_0 is such that in each resource bound b occurring in it, for at most one resource i , $b_i \neq \infty$.

5.2. Model-checking RB-ATL

In this section, we briefly revisit the problem of model-checking RB-ATL (the logic where only consumption of resources is considered). The syntax of RB-ATL is the same as the syntax of RB \pm ATL, and the models are the class of RB-CGS with no production of resource (all action costs are non-negative). We will refer to such models as RB-CGS $^-$. A symbolic model-checking algorithm for that logic was introduced in [9] (without infinite resource bounds). Here we re-state the algorithm and discuss upper and lower bounds for the complexity of RB-ATL model-checking.

The algorithm uses an abbreviation $split(b)$ that takes a resource bound b and returns the set of all pairs $(d, d') \in \mathbb{N}_\infty \times \mathbb{N}_\infty$ such that:

1. $d + d' = b$;
2. $d_i = d'_i = \infty$ for all $i \in \{1, \dots, r\}$ such that $b_i = \infty$; and
3. d has at least one non-0 value.

We assume that $split(b)$ is partially ordered in increasing order of the second component d' (so that if $d'_1 < d'_2$, then (d_1, d'_1) precedes (d_2, d'_2)).

The algorithm is similar to the symbolic model-checking algorithm for ATL given in [1]. The main differences from the algorithm for ATL is the addition of costs of actions, and, instead of working with a straightforward set of subformulas $Sub(\phi_0)$ of a given formula ϕ_0 , we work with an extended set of subformulas $Sub^+(\phi_0)$. $Sub^+(\phi_0)$ includes $Sub(\phi_0)$, and in addition:

- if $\langle\langle A^b \rangle\rangle \Box \phi \in Sub(\phi_0)$, then $\langle\langle A^{d'} \rangle\rangle \Box \phi \in Sub^+(\phi_0)$ for all d' such that $(d, d') \in split(b)$;
- if $\langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi \in Sub(\phi_0)$, then $\langle\langle A^{d'} \rangle\rangle \phi \mathcal{U} \psi \in Sub^+(\phi_0)$ for all d' such that $(d, d') \in split(b)$.

We assume that $Sub^+(\phi_0)$ is partially ordered in the increasing order of complexity and of resource bounds (so e.g., for $b \leq b'$, $\langle\langle A^b \rangle\rangle \Box \psi$ precedes $\langle\langle A^{b'} \rangle\rangle \Box \psi$).

Theorem 3. *Given an RB-CGS $^-$ $M = (Agt, Res, S, \Pi, \pi, Act, d, c, \delta)$ and an RB-ATL formula ϕ_0 , there is an algorithm which returns the set of states $[\phi_0]_M$ satisfying ϕ_0 : $[\phi_0]_M = \{s \in S \mid M, s \models \phi_0\}$, which runs in time $O(|\phi_0|^{2r+1} \times |M|^2)$ where r is $|Res|$, assuming that the components of the resource bounds are encoded in unary.²*

Proof. Let Pre as before be a function which given a coalition A , a set $\rho \subseteq S$ and a bound b returns a set of states s in which A has a move σ_A with cost $cost(s, \sigma_A) \leq b$ such that $out(s, \sigma_A) \subseteq \rho$. Note that $Pre(A, \rho, b)$ can be computed in $O(|\delta|)$. We assume that δ is already stored in a data structure where entries for $\delta(s, \sigma)$ are grouped together by state s . Then for each state s in S , partition the set of transitions σ from s into buckets where σ_A is the same joint action by A . This can be done in one pass over transitions from s . For each bucket, if the cost of σ_A is at most b , check whether $\delta(s, \sigma')$ for all σ' in this bucket is in ρ . If yes, add s to $Pre(A, \rho, b)$. Observe that each transition in δ is only traversed twice (once to put it in a bucket, and once to check its outcome).

Consider Algorithm 4 (where $\bar{0} \stackrel{\infty}{\leftarrow} b$ is a vector whose i th component is ∞ if the i th component of b is ∞ , and 0 otherwise). Note that $|split(b)|$ is $O(\beta^r)$, where β is the largest component occurring in b . If ϕ_0 contains operators with bounds containing components other than 0 and ∞ , $|Sub^+(\phi_0)|$ is $O(|\phi_0| \times \beta^r)$, or $O(|\phi_0| \times |\phi_0|^r)$ provided that vector components are encoded in unary. The main loop is executed for each formula in $Sub^+(\phi_0)$. The outer **foreach** loops for $\langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi$ and $\langle\langle A^b \rangle\rangle \Box \phi$ are executed $O(|split(b)|) = O(|\phi_0|^r)$ times, and the inner **while** loops are executed at most $|S|$ times calling an $O(|\delta|)$ function Pre . This gives complexity of the whole algorithm $O(|\phi_0| \times |\phi_0|^r \times |\phi_0|^r \times |S| \times |\delta|) = O(|\phi_0|^{2r+1} \times |M|^2)$. \square

6. Comparison with RAL

In this section, we compare RB \pm ATL with the logics introduced in [3], in particular with the logic pr-rf-RAL'. In [3], it is shown that the model-checking problem for pr-rf-RAL' with infinite semantics is undecidable. The decidability of the model-checking problem for pr-rf-RAL' with finite semantics is stated in [3] as an open problem. Here we show that model-checking for pr-rf-RAL' with finite semantics is decidable.

6.1. The logic pr-rf-RAL'

The logical language pr-rf-RAL' is a proponent-restricted and resource-flat version of RAL without the release operator (for a complete description of RAL and its variants, we refer the reader to [3] and its technical report version [12]; in fact the name pr-rf-RAL' comes from [12]).

² It is possible to give an algorithm which is linear in $|M|$, by reducing the problem to AND-OR search as in [1] and in addition keeping track of the costs of paths. Such a linear algorithm is assumed for the complexity result stated in [9].

Algorithm 4 Model-checking RB-ATL.

```

function RB-ATL-LABEL( $M, \phi_0$ )
  for  $\phi' \in \text{Sub}^+(\phi)$  do
    case  $\phi' = p, \neg\phi, \phi \wedge \psi$ 
      standard, see [1]
    case  $\phi' = \langle\langle A^b \rangle\rangle \bigcirc \phi$  [ $\phi'$ ] $_M \leftarrow \text{Pre}(A, [\phi]_M, b)$ 
    case  $\phi' = \langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi$  where  $b$  is such that for all  $i, b_i \in \{0, \infty\}$ :
       $\rho \leftarrow [\text{false}]_M; \tau \leftarrow [\psi]_M;$ 
      while  $\tau \not\subseteq \rho$  do
         $\rho \leftarrow \rho \cup \tau; \tau \leftarrow \text{Pre}(A, \rho, b) \cap [\phi]_M$ 
      od
      [ $\phi'$ ] $_M \leftarrow \rho$ 
    case  $\phi' = \langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi$  where  $b$  is such that for some  $i, b_i \notin \{0, \infty\}$ :
       $\rho \leftarrow [\text{false}]_M; \tau \leftarrow [\text{false}]_M$ 
      foreach  $d' \in \{d' \mid (d, d') \in \text{split}(b)\}$  do
         $\tau \leftarrow \text{Pre}(A, [\langle\langle A^{d'} \rangle\rangle \phi \mathcal{U} \psi]_M, d) \cap [\phi]_M$ 
        while  $\tau \not\subseteq \rho$  do
           $\rho \leftarrow \rho \cup \tau; \tau \leftarrow \text{Pre}(A, \rho, \bar{0} \stackrel{\infty}{\leftarrow} b) \cap [\phi]_M$ 
        od
      [ $\phi'$ ] $_M \leftarrow \rho$ 
    case  $\phi' = \langle\langle A^b \rangle\rangle \square \phi$  where  $b$  is such that for all  $i, b_i \in \{0, \infty\}$ :
       $\rho \leftarrow [\text{true}]_M; \tau \leftarrow [\phi]_M$ 
      while  $\rho \not\subseteq \tau$  do
         $\rho \leftarrow \tau; \tau \leftarrow \text{Pre}(A, \rho, b) \cap [\phi]_M$ 
      od
      [ $\phi'$ ] $_M \leftarrow \rho$ 
    case  $\phi' = \langle\langle A^b \rangle\rangle \square \phi$  where  $b$  is such that for some  $i, b_i \notin \{0, \infty\}$ :
       $\rho \leftarrow [\text{false}]_M; \tau \leftarrow [\text{false}]_M$ 
      foreach  $d' \in \{d' \mid (d, d') \in \text{split}(b)\}$  do
         $\tau \leftarrow \text{Pre}(A, [\langle\langle A^{d'} \rangle\rangle \square \phi]_M, d) \cap [\phi]_M$ 
        while  $\tau \not\subseteq \rho$  do
           $\rho \leftarrow \rho \cup \tau; \tau \leftarrow \text{Pre}(A, \rho, \bar{0} \stackrel{\infty}{\leftarrow} b) \cap [\phi]_M$ 
        od
      [ $\phi'$ ] $_M \leftarrow \rho$ 
  return [ $\phi_0$ ] $_M$ 

```

The syntax of pr-rf-RAL' is defined using *endowment functions* (or just endowments) rather than resource bounds. An endowment is a function $\eta : \text{Agt} \times \text{Res} \rightarrow \mathbb{N} \cup \{\infty\}$. We will sometimes write $\eta_a(r)$ instead of $\eta(a, r)$. Let En denote the set of all possible endowments.

Formulas of pr-rf-RAL' are defined as follows:

$$\phi, \psi ::= p \mid \neg\phi \mid \phi \wedge \psi \mid \langle\langle A \rangle\rangle^\eta \bigcirc \phi \mid \langle\langle A \rangle\rangle^\eta \square \phi \mid \langle\langle A \rangle\rangle^\eta \phi \mathcal{U} \psi$$

where $p \in \Pi$, $A \subseteq \text{Agt}$, $A \neq \emptyset$ and $\eta \in \text{En}$.

Formulas of pr-rf-RAL' are interpreted on resource-bounded models (RBM) which are RB-CGS (i.e., CGS structures extended with resources) whose transitions are in general not total, i.e., at a state, an agent is not required to have any available actions. This means that there may be a state in an RBM model which does not have any successor. An RBM is defined as follows:

Definition 2. An RBM is a tuple $M = (\text{Agt}, Q, \Pi, \pi, \text{Act}, d, o, \text{Res}, t)$ where Agt , Act , Q , Π , Res , and o are defined as Agt , Act except that *idle* is not required to be in Act , S , Π , Res , and δ , respectively, in Definition 1 and:

- $\pi : Q \rightarrow \wp(\Pi)$ specifies propositional valuation;
- $d : \text{Agt} \times Q \rightarrow \wp(\text{Act})$ specifies available actions;
- $t : \text{Act} \times \text{Res} \rightarrow \mathbb{Z}$ for an action $\alpha \in \text{Act}$ and a resource $r \in \text{Res}$ specifies the consumption of r by α if $t(\alpha, r) \leq 0$ or the production of r by α if $t(\alpha, r) > 0$. Let $\text{cons}(\alpha, r) = -\min\{0, t(\alpha, r)\}$ and $\text{prod}(\alpha, r) = \max\{0, t(\alpha, r)\}$.

Resource availability is modelled by resource-quantity mappings (rqm) $\rho : \text{Res} \rightarrow \mathbb{Z} \cup \{\infty\}$.

Given a RBM M , $Q^{\leq \omega} = Q^\omega \cup Q^+$ denotes the set of all finite and infinite sequences over Q . A sequence $\lambda \in Q^{\leq \omega}$ is a *path* in M iff there exist transitions in M between adjacent states in λ . A finite or infinite sequence $\lambda = (q_0, \eta^0), (q_1, \eta^1), \dots$ over $Q \times \text{En}$ is a *resource-extended path* (r-path) in M iff q_0, q_1, \dots is a path in M .

Given a coalition A , an endowment η and an rqm ρ , an (A, η) -share for ρ is a function $\text{sh} : A \times \text{Res} \rightarrow \mathbb{N}$ where:

- $\forall r \in \text{Res} : \rho(r) > 0 \Rightarrow \sum_{a \in A} \text{sh}(a, r) = \rho(r);$
- $\forall a \in A, r \in \text{Res} : \eta_a(r) \geq \text{sh}(a, r).$

Let $\text{Share}(A, \eta, \rho)$ denote the set of all possible (A, η) -shares for ρ . It is straightforward that $\text{Share}(A, \eta, \rho)$ is always finite and it is \emptyset iff $\sum_{a \in A} \eta_a(r) < \rho(r)$, i.e., resource endowment for agents in A is not enough to create a share.

As defined in [3], a path is maximal if it cannot be extended due to unavailability of resources. Given an endowment η and a strategy F_A for a coalition A , a maximal r -path $\lambda = (q_0, \eta^0), (q_1, \eta^1), \dots$ of M is an (η, F_A) -path starting from a state q_0 iff:

- $\eta^0 = \eta$;
- $\forall a \in A, r \in \text{Res}, i \geq 0, i < |\lambda| : \eta_a^i(r) \geq 0$;
- $\forall i \geq 0, i < |\lambda| - 1 : \exists \sigma \in D(q_i)$ such that:
 - $F_A(q_0 \dots q_i) = \sigma_A$;
 - $o(q_i, \sigma) = q_{i+1}$;
 - $\exists \text{sh}_i \in \text{Share}(A, \eta, \rho) : \forall a \in A, r \in \text{Res} : \eta_a^{i+1}(r) = \eta_a^i(r) + \text{prod}(\sigma_a, r) - \text{sh}_i(a, r)$ where ρ is such that $\rho(r) = \sum_{a \in A} \text{cons}(\sigma_a, r)$.

Then, $\text{out}(q_0, \eta, F_A)$ denotes the set of all (η, F_A) -paths starting from a state q_0 . As shown in [3], $\text{out}(q_0, \eta, F_A)$ is never empty. In the worst case, $\text{out}(q_0, \eta, F_A)$ contains a single r -path (q_0, η) .

Given an RBM M and a state q , the truth of pr-rf-RAL' formulas is defined inductively as follows (we omit the propositional cases):

- $M, q \models_{\text{ral}} \langle\langle A \rangle\rangle^? \bigcirc \phi$ iff $\exists F_A : \forall \lambda \in \text{out}(q, \eta, F_A) : |\lambda| \geq 2 \wedge M, \lambda[1] \models_{\text{ral}} \phi$;
- $M, q \models_{\text{ral}} \langle\langle A \rangle\rangle^? \square \phi$ iff $\exists F_A : \forall \lambda \in \text{out}(q, \eta, F_A) : |\lambda| = \infty \wedge \forall i \geq 0 : M, \lambda[i] \models_{\text{ral}} \phi$;
- $M, q \models_{\text{ral}} \langle\langle A \rangle\rangle^? \phi \mathcal{U} \psi$ iff $\exists F_A : \forall \lambda \in \text{out}(q, \eta, F_A) : \exists i \geq 0, i < |\lambda| : M, \lambda[i] \models_{\text{ral}} \psi \wedge \forall j \geq 0, j < i : M, \lambda[j] \models_{\text{ral}} \phi$.

The definition above gives *finite semantics* of pr-rf-RAL'. *Infinite semantics* is obtained if the condition “for all $\lambda \in \text{out}(q, \eta, F_A)$ ” above is replaced with “for all infinite $\lambda \in \text{out}(q, \eta, F_A)$ ”.

Theorem 4. [3,12] *The model-checking problem for pr-rf-RAL' with infinite semantics is undecidable.*

The problem whether model-checking for pr-rf-RAL' with finite semantics is decidable is left open in [3]. Below we show that it is in fact decidable by adapting the model-checking algorithm for $\text{RB}\pm\text{ATL}$. Before we do this, we investigate the differences between pr-rf-RAL' and $\text{RB}\pm\text{ATL}$ in more detail. In particular we consider whether we can obtain a logic equivalent to pr-rf-RAL' by simply removing the restriction that agents always have at least the *idle* action available from the semantics of $\text{RB}\pm\text{ATL}$.

6.2. The logic $\text{RB}\pm\text{ATL-nt}$

As models for pr-rf-RAL' are not total in general, we facilitate a comparison with $\text{RB}\pm\text{ATL}$ by introducing a variant $\text{RB}\pm\text{ATL-nt}$ of $\text{RB}\pm\text{ATL}$ where we remove the requirement of total transitions in Definition 1. In other words, $\text{RB}\pm\text{ATL-nt}$ has the same syntax as $\text{RB}\pm\text{ATL}$ yet a broader class, namely RB-CGS-nt , of models which do not need to be total. In particular, in Definition 1, *Act* does not need to include *idle* and $d : S \times \text{Agt} \rightarrow \wp(\text{Act})$ may be mapped to an empty set or to a set not containing *idle*.

Obviously, any RB-CGS model is an RB-CGS-nt but not vice versa. Since RB-CGS-nt models are not total in general, at a state s , the set $D_A(s)$ of possible joint actions by a coalition A and the set of possible outcomes of a joint action $\sigma_A \in D_A(s)$ may be empty.

Given a RB-CGS-nt model M , a strategy F_A for a coalition $A \subseteq \text{Agt}$, a finite computation $\lambda \in S^+$ is consistent with F_A iff for all $i \in \{0, \dots, |\lambda| - 2\} : \lambda[i+1] \in \text{out}(\lambda[i], F_A(\lambda[0, i]))$ and $D_{\text{Agt}}(\lambda[|\lambda| - 1]) = \emptyset$, i.e., there is a deadlock at the last state of λ . We denote by $\text{out}_f(s, F_A)$ the set of all consistent finite computations of F_A starting from s . Then, the set of all consistent finite and infinite computations of F_A from s is defined as:

$$\text{out}_{\text{nt}}(s, F_A) = \text{out}(s, F_A) \cup \text{out}_f(s, F_A)$$

Under a resource bound $b \in B$, a computation $\lambda \in \text{out}_{\text{nt}}(s, F_A)$ can be only carried out until an index $i_{\text{max}} \in \mathbb{N}_{\infty}$ (see Fig. 7) iff:

$$\sum_{j=0}^i \text{cost}(\lambda[j], F_A(\lambda[0, j])) \leq b \text{ for all } i < i_{\text{max}}$$

and

$$\sum_{j=0}^{i_{\text{max}}} \text{cost}(\lambda[j], F_A(\lambda[0, j])) \not\leq b \text{ if } i_{\text{max}} \neq \infty$$

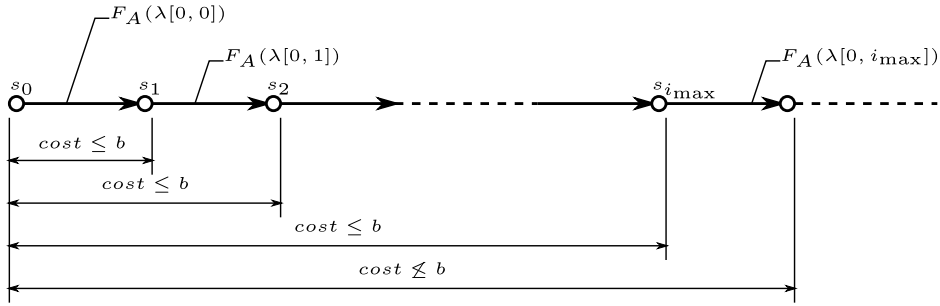


Fig. 7. λ is restricted by b .

Let us denote $\lambda(b) = \lambda[0, i_{\max}]$ and we call $\lambda(b)$ maximal with respect to b . Then, the set of all b -consistent (finite or infinite) computations of F_A starting from state s is defined as follows:

$$out_{nt}(s, F_A, b) = \{\lambda(b) \mid \lambda \in out_{nt}(s, F_A)\}$$

Note that this definition implies that the cost of every prefix of a b -consistent computation is below b and $out_{nt}(s, F_A, b)$ may contain finite computations. Furthermore, $out_{nt}(s, F_A, b)$ is always non-empty, as in the worst case, it contains a single computation s .

The semantics of RB \pm ATL-nt formulas is defined as follows (the atomic case and Boolean connectives are defined in the standard way):

- $M, s \models_{nt} \langle\langle A^b \rangle\rangle \bigcirc \phi$ iff \exists strategy F'_A such that for all $\lambda \in out_{nt}(s, F'_A, b)$: $|\lambda| \geq 2$ and $M, \lambda[1] \models \phi$;
- $M, s \models_{nt} \langle\langle A^b \rangle\rangle \square \phi$ iff \exists strategy F'_A such that for all $\lambda \in out_{nt}(s, F'_A, b)$ and $i \geq 0$: $|\lambda| = \infty$ and $M, \lambda[i] \models \phi$; and
- $M, s \models_{nt} \langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi$ iff \exists strategy F'_A such that for all $\lambda \in out_{nt}(s, F'_A, b)$, $\exists i \geq 0$: $i < |\lambda|$, $M, \lambda[i] \models \psi$ and $M, \lambda[j] \models \phi$ for all $j \in \{0, \dots, i-1\}$.

If the condition “for all $\lambda \in out_{nt}(s, F'_A, b)$ ” is replaced with “for all infinite $\lambda \in out_{nt}(s, F'_A, b)$ ” in the truth definition of RB \pm ATL-nt, we obtain RB \pm ATL-nt with infinite semantics. Note that in a RB-CGS model M , if F_A is a b -strategy for a coalition A , we have that $out(s, F_A) = out(s, F_A, b) = out_{nt}(s, F_A) = out_{nt}(s, F_A, b)$. We have the following result:

Lemma 7. Given a RB-CGS model M , $M, s \models \phi'$ iff $M, s \models_{nt} \phi'$ under finite semantics.

Proof. (\Rightarrow) is obvious. For (\Leftarrow), the proof is by induction on the structure of ϕ' .

If $\phi' = \langle\langle A^b \rangle\rangle \square \phi$, we have that $out_{nt}(s, F'_A) = out_{nt}(s, F'_A, b)$ because $|\lambda| = \infty$ for all $\lambda \in out_{nt}(s, F'_A, b)$; thus, F'_A is a b -strategy.

If $\phi' = \langle\langle A^b \rangle\rangle \bigcirc \phi$, let us consider the following strategy for A :

$$F_A(\lambda) = \begin{cases} F'_A(\lambda) & \text{if } \exists \lambda' \in S^+ \cup S^\omega : \lambda\lambda' \in out(s, F'_A, b) \\ \text{idle} & \text{otherwise.} \end{cases}$$

It is straightforward that F_A is a b -strategy to satisfy $M, s \models \langle\langle A^b \rangle\rangle \bigcirc \phi$ at s .

If $\phi' = \langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi$, the proof is similar to the above case, hence it is omitted here. \square

The above result shows that over the class of RB-CGS models, RB \pm ATL and RB \pm ATL-nt with finite semantics are equivalent. Furthermore, we have the following result:

Theorem 5. The model-checking problem for RB \pm ATL-nt with finite semantics is decidable.

Proof. The model-checking algorithm for RB \pm ATL can be easily adapted to a model-checking algorithm for RB \pm ATL-nt. The only change required is in the case of $\langle\langle A^b \rangle\rangle \bigcirc \phi$, the function $Pre(A, \rho, b)$ is redefined as $Pre(A, \rho, b) = \{s \in S \mid \exists \sigma_A \in D_A(s) : cost(s, \sigma_A) \leq b \wedge out(s, \sigma_A) \subseteq \rho \wedge out(s, \sigma_A) \neq \emptyset\}$. Here, we additionally require that $out(s, \sigma_A) \neq \emptyset$. UNTIL-STRATEGY and BOX-STRATEGY do not change; observe that if the set Act_A of available actions is empty, the algorithms return false. \square

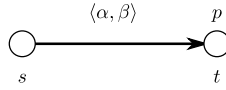


Fig. 8. Comparing resource endowments and bounds.

6.3. Comparing pr-rf-RAL' and RB±ATL-nt

At the semantical level, every RBM $M = (Agt, Q, \Pi, \pi, Act, d, o, Res, t)$ can be converted straightforwardly into an RB-CGS-nt $M' = (Agt, Res, Q, \Pi, \pi', Act, d, c, \delta)$ where:

- $\pi'(p) = \{q \in Q \mid p \in \pi(q)\}$ for all $p \in \Pi$;
- $c(q, a, \alpha) = (-t(\alpha, r))_{r \in Res}$ for all $q \in Q, a \in Agt, \alpha \in Act$; and
- $\delta = o$.

At the syntactical level, pr-rf-RAL' and RB±ATL-nt are rather different. While pr-rf-RAL' enables specifying the ability of a coalition under a resource endowment, RB±ATL-nt allows one to specify the ability of a coalition within a resource bound. Let us consider an example, as depicted in Fig. 8, in order to clarify the difference between endowments and bounds. In this example, our model has two agents a and b and one resource. From state s , agents a and b can only perform α and β , respectively, which cost $-c$ and c (for some $c > 0$), respectively. As their joint action is cost-free, we have that $s \models_{nt} \langle \langle \{a, b\}^0 \rangle \rangle \circ p$. However, given an empty endowment $\eta_0 = \{a \mapsto 0, b \mapsto 0\}$, there is no possible share from this endowment to cover the cost c of action β ; i.e., $s \not\models_{ral} \langle \langle \{a, b\} \rangle \rangle^{\eta_0} \circ p$. This is because (s, η_0) is the only r -path from s under η_0 and it is shorter than the computation s, t under 0 . In general, we have the following result:

Lemma 8. Given a RBM model M , for any state q_0 , coalition $A \subseteq Agt$, strategy F_A , endowment η^0 and bound $b = (\sum_{a \in A} \eta_a^0(r))_{r \in Res}$, if $(q_0, \eta^0), (q_1, \eta^1), \dots, (q_k, \eta^k)$ is the prefix of some r -path in $out(q_0, \eta^0, F_A)$, then $q_0 q_1 \dots q_k$ is also the prefix of some computation in $out_{nt}(q_0, F_A, b)$.

Proof. The proof is done by induction on k ; additionally, we also show that $(\sum_{a \in A} \eta_a^k(r))_{r \in Res} = b - \sum_{j=0}^{k-1} cost(q_j, F_A(q_1 \dots q_j))$.

Base case $k = 0$: The proof is trivial.

Induction step: Assume that $(q_0, \eta^0), (q_1, \eta^1), \dots, (q_{k+1}, \eta^{k+1})$ is the prefix of some r -path in $out(q_0, \eta^0, F_A)$. Then, so is $(q_0, \eta^0), (q_1, \eta^1), \dots, (q_k, \eta^k)$. By induction hypothesis, we have that $q_0 \dots q_k$ is the prefix of some computation in $out_{nt}(q_0, F_A, b)$ and $(\sum_{a \in A} \eta_a^k(r))_{r \in Res} = b - \sum_{j=0}^{k-1} cost(q_j, F_A(q_1 \dots q_j))$.

As $(q_0, \eta^0), (q_1, \eta^1), \dots, (q_{k+1}, \eta^{k+1})$ is a prefix, $Share(A, \eta^k, (\sum_{a \in A} cons(F_A(q_0 \dots q_k)_a, r)))_{r \in Res} \neq \emptyset$, i.e., $\sum_{a \in A} \eta_a^k(r) \geq \sum_{a \in A} cons(F_A(q_0 \dots q_k)_a, r)$ for all $r \in Res$; hence $\sum_{a \in A} \eta_a^{k+1}(r) \geq \sum_{a \in A} prod(F_A(q_0 \dots q_k)_a, r) \geq 0$.

We also have $(\sum_{a \in A} \eta_a^{k+1}(r))_{r \in Res} = (\sum_{a \in A} (\eta_a^k(r) + prod(F_A(q_0 \dots q_k)_a, r) - sh_k(a, r)))_{r \in Res} = (\sum_{a \in A} (\eta_a^k(r) + prod(F_A(q_0 \dots q_k)_a, r) - cons(F_A(q_0 \dots q_k)_a, r)))_{r \in Res} = b - \sum_{j=0}^k cost(q_j, F_A(q_1 \dots q_j))$. As $\sum_{a \in A} \eta_a^{k+1}(r) \geq 0$ for all $r \in Res$, $b - \sum_{j=0}^k cost(q_j, F_A(q_1 \dots q_j)) \geq \bar{0}$, i.e., $\sum_{j=0}^k cost(q_j, F_A(q_1 \dots q_j)) \leq b$, hence $q_0 \dots q_{k+1}$ is also a prefix of some computation in $out_{nt}(q_0, F_A, b)$. \square

As suggested by the function η^b which translates resource bounds into endowments (introduced in [12] by Bulling and Farwer to relate their framework to RBCL [13]), pr-rf-RAL' formulas can also be converted into RB±ATL-nt formulas by a translation function tr where an endowment η is converted to a bound $(\sum_{a \in A} \eta_a(r))_{r \in Res}$. The function tr is defined inductively as follows (propositional cases are omitted):

- $tr(\langle \langle A \rangle \rangle^\eta \circ \phi) = \langle \langle A \rangle \rangle^{(\sum_{a \in A} \eta_a(r))_{r \in Res}} \circ tr(\phi)$;
- $tr(\langle \langle A \rangle \rangle^\eta \square \phi) = \langle \langle A \rangle \rangle^{(\sum_{a \in A} \eta_a(r))_{r \in Res}} \square tr(\phi)$; and
- $tr(\langle \langle A \rangle \rangle^\eta \phi \mathcal{U} \psi) = \langle \langle A \rangle \rangle^{(\sum_{a \in A} \eta_a(r))_{r \in Res}} tr(\phi) \mathcal{U} tr(\psi)$.

Here, resource bounds are sums of individual endowments for each resource. The example in Fig. 8 and Lemma 8 show that satisfiability is not preserved by the translation function tr . In order to obtain preservation of satisfiability, it is necessary to relax the requirement in the definition of computations in RBM models. In particular, the last condition is relaxed as follows:

- $\exists sh_i \in Share(A, \eta, \rho) : \forall a \in A, r \in Res : \eta_a^{i+1}(r) = \eta_a^i(r) + prod(\sigma_a, r) - sh_i(a, r)$ where $\sigma_A = F_A(q_0 \dots q_i)$ and $\rho(r) = \sum_{a \in A} (cons(\sigma_a, r) - prod(\sigma_a, r))$.

Comparing the original condition, the production of resource in a step is also considered to cover for the consumption in the same step by adding it into the share function. Let us call RBM models with this relaxed condition relaxed RBM models. We have the following result:

Lemma 9. *Given a relaxed RBM model M , for any state q_0 , strategy F_A , endowment η^0 and bound $b = (\sum_{a \in A} \eta_a(r))_{r \in Res}$, then:*

- if $(q_0, \eta^0), (q_1, \eta^1), \dots \in out(q_0, \eta^0, F_A)$, then $q_0 q_1 \dots \in out(q_0, F_A, b)$;
- conversely, if $q_0 q_1 \dots \in out_{nt}(q_0, F_A, b)$; then $\exists \eta^1, \eta^2 \dots$ such that $(q_0, \eta^0), (q_1, \eta^1), \dots \in out(q_0, \eta^0, F_A)$.

Proof. The claim directly follows from Lemma 8. \square

Let pr-rf-RAL" be pr-rf-RAL' interpreted over relaxed RBM models. We have the following result:

Lemma 10. *Given a relaxed RBM model M , $M, s \models_{pr-rf-RAL''} \phi'$ iff $M', s \models_{nt} tr(\phi')$.*

Proof. Let us prove the direction from left to right. The other direction is similar. The proof is done by induction on the structure of ϕ' . The base case is trivial, hence omitted here.

In the induction step, the cases of propositional connectives are trivial, hence they are also omitted. Let us consider the following three cases.

$\phi' = \langle\langle A \rangle\rangle^{\eta} \bigcirc \phi$: Let $b = (\sum_{a \in A} \eta_a(r))_{r \in Res}$ and F_A be the strategy to satisfy ϕ' at s . For every $q_0 q_1 \dots \in out_{nt}(s, F_A, b)$ where $s = q_0$, by Lemma 9, there are η^1, η^2 such that $(q_0, \eta^0)(q_1, \eta^1) \dots \in out(s, \eta, F_A)$. As $M, s \models_{ral} \langle\langle A \rangle\rangle^{\eta} \bigcirc \phi$, we have that $M, q_1 \models_{ral} \phi$. By induction hypothesis, $M', q_1 \models_{nt} tr(\phi)$. Hence, $M', s \models \langle\langle A^b \rangle\rangle \bigcirc tr(\phi)$.

$\phi' = \langle\langle A \rangle\rangle^{\eta} \phi \mathcal{U} \psi$: Let $b = (\sum_{a \in A} \eta_a(r))_{r \in Res}$ and F_A be the strategy to satisfy ϕ' at s . For every $q_0 q_1 \dots \in out_{nt}(s, F_A, b)$ where $s = q_0$, by Lemma 9, there are η^1, η^2 such that $(q_0, \eta^0)(q_1, \eta^1) \dots \in out(s, \eta, F_A)$. As $M, s \models_{ral} \langle\langle A \rangle\rangle^{\eta} \phi \mathcal{U} \psi$, we have that $\exists i \geq 0$ such that $M, q_j \models_{ral} \phi$ for all $j < i$ and $M, q_i \models_{ral} \psi$. By induction hypothesis, $M', q_j \models_{nt} tr(\phi)$ for all $j < i$ and $M, q_i \models_{ral} \psi$. Hence, $M', s \models \langle\langle A^b \rangle\rangle tr(\phi) \mathcal{U} tr(\psi)$.

$\phi' = \langle\langle A \rangle\rangle^{\eta} \square \phi$: Let $b = (\sum_{a \in A} \eta_a(r))_{r \in Res}$ and F_A be the strategy to satisfy ϕ' at s . For every $q_0 q_1 \dots \in out_{nt}(s, F_A, b)$ where $s = q_0$, by Lemma 9, there are η^1, η^2 such that $(q_0, \eta^0)(q_1, \eta^1) \dots \in out(s, \eta, F_A)$. As $M, s \models_{ral} \langle\langle A \rangle\rangle^{\eta} \phi \mathcal{U} \psi$, we have that $(q_0, \eta^0)(q_1, \eta^1) \dots$ is infinite and $M, q_j \models_{ral} \phi$ for all $j \geq 0$. By induction hypothesis, $M', q_j \models_{nt} tr(\phi)$ for all $j \geq 0$. Hence, $M', s \models \langle\langle A^b \rangle\rangle \square tr(\phi)$. \square

The above lemma shows that over the class of relaxed RBM models, RB \pm ATL-nt and pr-rf-RAL" with finite semantics are equivalent. Similar to the above result, it is also straightforward that RB \pm ATL-nt with infinite semantics is equivalent to pr-rf-RAL" with infinite semantics:

Lemma 11. *Given a relaxed RBM model M , under the infinite semantics, $M, s \models_{pr-rf-RAL''} \phi'$ iff $M', s \models_{nt} tr(\phi')$.*

Proof. The proof is the same as the proof of Lemma 10 except we only consider infinite computations. \square

Note that the proof for the undecidability of pr-rf-RAL' in [3] with infinite semantics can be applied for pr-rf-RAL" with infinite semantics. Hence, we have the following result:

Lemma 12. *Model-checking pr-rf-RAL" with infinite semantics is undecidable.*

Then, we have the following consequences:

Corollary 5. *Model-checking RB \pm ATL-nt with infinite semantics is undecidable.*

Since model-checking RB \pm ATL-nt with finite semantics is decidable, we have:

Lemma 13. *Model-checking pr-rf-RAL" with finite semantics is decidable.*

Furthermore, the same result can also be established for pr-rf-RAL':

Theorem 6. *Model-checking pr-rf-RAL' with finite semantics is decidable.*

Algorithm 5 Labelling $\langle\langle A \rangle\rangle^{\eta} \phi \mathcal{U} \psi$.

```

1: function UNTIL-STRATEGY( $n, \langle\langle A \rangle\rangle^{\eta} \phi \mathcal{U} \psi$ )
2:   if  $s(n) \not\models \langle\langle A \rangle\rangle^{\infty} \phi \mathcal{U} \psi$  then
3:     return false
4:   if  $\exists n' \in p(n) : s(n') = s(n) \wedge (\forall a \in A, r \in Res : \eta(n')(a, r) \geq \eta(n)(a, r))$  then
5:     return false
6:   for  $(a, i) \in \{(a, i) \in A \times Res \mid \exists n' \in p(n) : s(n') = s(n) \wedge$ 
       $(\forall b \in A, j \in Res : \eta(n')(b, j) \leq \eta(n)(b, j)) \wedge$ 
       $\eta(n')(a, i) < \eta(n)(a, i)\}$  do
7:      $\eta(n)(a, i) \leftarrow \infty$ 
8:   if  $s(n) \models \psi$  then
9:     return true
10:  if  $\eta(n) = \infty$  then
11:    return true
12:   $ActA \leftarrow \{\sigma \in D_A(s(n)) \mid \sum_{a \in A} \text{cons}(s(n), \sigma_a) \leq \sum_a \eta(n)(a)\}$ 
13:  for  $\sigma \in ActA$  do
14:     $O \leftarrow \text{out}(s(n), \sigma)$ 
15:     $strat \leftarrow true$ 
16:    for  $s' \in O$  do
17:       $shstrat \leftarrow false$ 
18:      for  $sh \in \text{Share}(A, \eta(n), \sum_{a \in A} \text{cons}(s(n), \sigma_a))$  do
19:         $shstrat \leftarrow shstrat \vee$ 
20:          UNTIL-STRATEGY( $node(n, \sigma, sh, s'), \langle\langle A \rangle\rangle^{\eta} \phi \mathcal{U} \psi$ )
21:        if  $shstrat$  then
22:          break
23:       $strat \leftarrow strat \wedge shstrat$ 
24:    if  $strat$  then
25:      return true
26:  return false

```

| Semantics | RB \pm ATL | RB \pm ATL-nt | pr-rf-RAL' | pr-rf-RAL' |
|-----------|--------------|-----------------|------------|------------|
| Finite | D | D | D | D |
| Infinite | D | U | U | U [3] |

Fig. 9. Decidability and undecidability results.

Proof. We adapt further the model-checking algorithm for RB \pm ATL-nt (which in turn is an adaptation of the one for RB \pm ATL) by replacing the role of resource bounds for a coalition with endowments for each agent in the coalition. In particular, each node in a search tree consists of a state, an endowment (where $\eta(n)$ returns the endowment of a node n) and a finite path of nodes from the root leading to this node. We also adapt the function $node$ to take into account shares where $node(n, \sigma_A, sh, s')$ returns a node n' where $s(n') = s'$, $p(n') = [p(n) \cdot n]$ and for all $a \in A$ and $r \in Res$: $\eta(n')(a, r) = \eta(n)(a, r) + \text{prod}(\sigma_a, r) - \text{sh}(a, r)$. The adaptation of Algorithm 2 is shown in Algorithm 5. In this algorithm, apart from the exchange of bounds with endowments, we also add an or-search for shares in lines 18–22. A similar adaption can be applied to Algorithm 3 and the detail is omitted here. \square

Fig. 9 summarizes the above decidability and undecidability results for the model-checking problems for RB \pm ATL, RB \pm ATL-nt, pr-rf-RAL' and pr-rf-RAL' where D stands for decidable and U for undecidable. Note that RB \pm ATL is decidable in both semantics due to the fact that both semantics are indistinguishable thanks to *idle*.

7. Conclusion

We have presented a model-checking algorithm for RB \pm ATL, a logic with *resource production*, which makes RB \pm ATL exceptional in the landscape of resource logics, for most of which the model-checking problem is undecidable [3,6]. We compared RB \pm ATL with a similar logic (a variant of RAL, [3]) to understand the differences between the two logics and why the model-checking problem for RB \pm ATL is decidable while the model-checking problem for pr-rf-RAL' with infinite semantics is undecidable. As a by-product of this comparison, we show that the model-checking problem for pr-rf-RAL' with finite semantics is decidable, solving a problem left open in [3].

Although the model-checking problem for RB \pm ATL is decidable, it is EXPSPACE-hard. In future work, we plan to implement model-checking algorithms for feasible fragments of RB \pm ATL in the model-checker MCMAS [14].

Acknowledgment

We thank the anonymous reviewer for their very thorough, detailed and constructive feedback.

References

- [1] R. Alur, T. Henzinger, O. Kupferman, Alternating-time temporal logic, *J. ACM* 49 (5) (2002) 672–713.
- [2] N. Bulling, J. Dix, W. Jamroga, Model checking logics of strategic ability: complexity, in: *Specification and Verification of Multi-agent Systems*, Springer, 2010, pp. 125–159.
- [3] N. Bulling, B. Farwer, On the (un-)decidability of model checking resource-bounded agents, in: *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI 2010)*, in: *Frontiers in Artificial Intelligence and Applications*, vol. 215, IOS Press, 2010, pp. 567–572.
- [4] D. Della Monica, M. Napoli, M. Parente, On a logic for coalitional games with priced-resource agents, *Electron. Notes Theor. Comput. Sci.* 278 (2011) 215–228.
- [5] D. Della Monica, M. Napoli, M. Parente, Model checking coalitional games in shortage resource scenarios, in: *Proceedings of the 4th International Symposium on Games, Automata, Logics and Formal Verification (GandALF 2013)*, in: *EPTCS*, vol. 119, 2013, pp. 240–255.
- [6] N. Bulling, V. Goranko, How to be both rich and happy: combining quantitative and qualitative strategic reasoning about multi-player games (extended abstract), in: F. Mogavero, A. Murano, M.Y. Vardi (Eds.), *Proceedings of the 1st International Workshop on Strategic Reasoning, SR 2013*, in: *EPTCS*, vol. 112, 2013, pp. 33–41.
- [7] E.M. Clarke, E.A. Emerson, A.P. Sistla, Automatic verification of finite-state concurrent systems using temporal logic specifications, *ACM Trans. Program. Lang. Syst.* 8 (2) (1986) 244–263.
- [8] N. Alechina, B. Logan, H.N. Nguyen, F. Raimondi, Decidable model-checking for a resource logic with production of resources, in: *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI 2014)*, IOS Press, 2014, pp. 9–14.
- [9] N. Alechina, B. Logan, H.N. Nguyen, A. Rakib, Resource-bounded alternating-time temporal logic, in: *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, IFAAMAS, 2010, pp. 481–488.
- [10] W. Reisig, *Petri Nets: An Introduction*, EATCS Monographs on Theoretical Computer Science, vol. 4, Springer, 1985.
- [11] J. Leroux, Acceleration for Petri nets, in: D.V. Hung, M. Ogawa (Eds.), *Proceedings of the 11th International Symposium on Automated Technology for Verification and Analysis, ATVA 2013*, in: *Lecture Notes in Computer Science*, vol. 8172, Springer, 2013, pp. 1–4.
- [12] N. Bulling, B. Farwer, On the (Un-)Decidability of Model Checking Resource-Bounded Agents, *Tech. Rep. IfI-10-05*, Clausthal University of Technology, 2010.
- [13] N. Alechina, B. Logan, H.N. Nguyen, A. Rakib, A logic for coalitions with bounded resources, in: *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI 2009)*, vol. 2, IJCAI/AAAI, AAAI Press, 2009, pp. 659–664.
- [14] A. Lomuscio, H. Qu, F. Raimondi, MCMAS: a model checker for the verification of multi-agent systems, in: A. Bouajjani, O. Maler (Eds.), *Proceedings of the 21st International Conference on Computer Aided Verification (CAV 2009)*, in: *Lecture Notes in Computer Science*, vol. 5643, Springer, 2009, pp. 682–688.