Parameterised Resource-Bounded ATL

Natasha Alechina
Utrecht University
Utrecht, The Netherlands
n.a.alechina@uu.nl

Stéphane Demri
LSV, CNRS, ENS Paris-Saclay,
University Paris-Saclay
Cachan, France
demri@lsv.fr

Brian Logan
University of Nottingham
Nottingham, UK
brian.logan@nottingham.ac.uk

Abstract

It is often advantageous to be able to extract resource requirements in resource logics of strategic ability, rather than to verify whether a fixed resource requirement is sufficient for achieving a goal. We study Parameterised Resource-Bounded Alternating Time Temporal Logic where parameter extraction is possible. We give a parameter extraction algorithm and prove that the model-checking problem is 2EXPTIME-complete.

Introduction

There has been a considerable amount of work on logics of strategic ability interpreted over structures where agents' actions consume resources, or both produce and consume resources. Examples include an extension of Coalition Logic where actions consume resources and coalitional modalities are annotated with resource bounds ('agents in coalition A have a strategy of cost at most b to achieve ϕ') (RBCL) (Alechina et al. 2009; 2011), a similar extension for Alternating Time Temporal Logic ATL, Resource-Bounded ATL (RB-ATL) (Alechina et al. 2010), extensions of Computation Tree Logic and Alternating Time Temporal Logic with both consumption and production of resources (RTL, RAL) (Bulling and Farwer 2010a; 2010b), a variant of Resource-Bounded ATL where all resources are convertible to money and the amount of money is bounded (PRB-ATL) (Della Monica, Napoli, and Parente 2011; 2013), an extension of PRB-ATL to μ-calculus (Della Monica and Lenzi 2012), a version of ATL with more general numerical constraints (QATL*) (Bulling and Goranko 2013), and a version of RB-ATL where unbounded production of resources is allowed (RB$\pm$ATL) (Alechina et al. 2017b). These logics can express properties such as: 'a coalition of agents with n units of resource 1 and m units of resource 2 can reach a state where they can maintain property ϕ forever without spending any resources'. Essentially, such logics enable formulating and solving multi-agent conditional planning problems with resources, where the planning problems may be nested and involve invariant properties.

However, with one exception, these logics assume that the amount of resources available to the agents is *fixed* and known in advance. Using these logics, it is possible to ask whether a strategy to achieve some goal exists if the agents start with e.g., 100 units of energy. It is not possible to ask what is the *minimal* amount of energy that makes some goal achievable (unless it is already known that the goal is achievable for some fixed cost). In other words, it is not possible to extract the values of resource parameters using model-checking procedures for these logics. The only exception is the logic ParRB$\pm$ATL* (Parameterised RB$\pm$ATL*, Resource Bounded Alternating Time Temporal Logic) introduced in (Alechina et al. 2018) that allows concrete values for resource parameters to be synthesised. Using ParRB$\pm$ATL*, we can ask what is the minimal amount of energy needed for achieving some goal, or for which values of x and y does it hold that coalition A has a strategy requiring at most x units of resource to enforce some temporal goal, and coalition A' does not have a strategy to enforce another temporal goal with y units of resource. The function form of the model-checking problem for ParRB$\pm$ATL* (return a set of constraints on the resource variables for which the formula is true) was shown to be decidable in (Alechina et al. 2018), but the upper bound on the complexity of the problem was left open.

In this paper we study the model-checking problem for the logic ParRB$\pm$ATL (Parameterised RB$\pm$ATL). ParRB$\pm$ATL is a fragment of ParRB$\pm$ATL* with the same syntax as RB$\pm$ATL (Resource Bounded ATL), but with variables instead of concrete values for resource amounts. We show that the complexity of the model-checking problem for ParRB$\pm$ATL is 2EXPTIME-complete, and give an explicit algorithm to compute the set of assignments satisfying the formula. For positive formulas, this algorithm computes the set of Pareto optimal (non-dominated) resource bounds satisfying the formula.

Our results build on results and techniques developed in games on vector addition systems with states (Courtois and Schmitz 2014) and energy games (Jurdzinski, Lazić, and Schmitz 2015). In particular, we use techniques for solving reachability problems with unknown initial credit for vector addition systems with states for model checking formulas

with the Until temporal operator, and techniques for solving non-termination problems with unknown initial credit for energy games for model checking invariant properties. The crux of establishing a 2EXPTIME upper bound for the ParRB±ATL model-checking problem is showing that there is a maximal value of the amount of resource needed to reach a particular state or to enter and execute a non-resource-consuming loop, and this bound depends only on the model and not on the property to be checked.

In the remainder of this section, we briefly recall other research in multi-weighted games related to our work. Juhl, Larsen and Raskin (2013) consider the problem of computing minimal winning vectors in multi-weighted games. They state three decision problems, but focus on the second and third (computing minimal upper bounds b such that on all infinite computations starting from 0, values stay below b), which are less relevant for us. Their first problem, computing minimal winning vectors for multiweighted games when all the vectors in the infinite runs have non-negative values, is the most related to model checking ParRB±ATL. At the time of (Juhl, Larsen, and Raskin 2013), the best known approach to this problem was that of Brázdil, Jančar and Kucera (2010), which has upper bound (k-1)-EXPTIME, where k is the number of resource types. Jurdzinski, Lazić and Schmitz (2015) significantly improved on the results in (Brázdil, Jančar, and Kucera 2010), which is why we use the results from (Jurdzinski, Lazić, and Schmitz 2015) in our paper.

The main contribution of our paper is to show that the maximal value given in (Jurdzinski, Lazić, and Schmitz 2015) transfers to ParRB±ATL, and to show how it can be used for nested temporal goals that include negations and a mixture of reachability and non-termination goals.

Syntax and Semantics of ParRB±ATL

In this section, we introduce the syntax and semantics of ParRB±ATL.

Let $Agt = \{a_1, \ldots, a_n\}$ be a set of $n \geq 1$ agents and $Res = \{res_1, \ldots, res_r\}$ be a set of $r \geq 1$ resource types, Π denote a set of propositions, and Var a countably infinite set of variables.

We write ParRB±ATL(n,r) for the logic ParRB±ATL with n agents and r resources, interpreted over models with the same numbers of agents and resources. When talking about all possible n and r, we write ParRB±ATL.

Formulas of ParRB±ATL(n,r) are defined by the following syntax

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \psi \mid \langle\!\langle A^{\mathbf{x}} \rangle\!\rangle \bigcirc \varphi \mid \langle\!\langle A^{\mathbf{x}} \rangle\!\rangle \Box \varphi \mid \langle\!\langle A^{\mathbf{x}} \rangle\!\rangle \varphi \, \mathcal{U} \, \psi$$

where $p \in \Pi$ is a proposition, $A \subseteq Agt$, and $\mathbf{x}$ is a sequence of variables $x_1, \ldots, x_r$ from Var of length r, intuitively placeholders for amounts of resource 1 to resource r.

Since formulas contain variables, formulas are evaluated relative to a model, a state and an assignment of values to variables. Informally, a state s and assignment $x_1 \mapsto b_1, \ldots, x_r \mapsto b_r$ where the b_i's are in $\mathbb{N}$ (which we will write as a mapping between vectors, $\mathbf{x} \mapsto \mathbf{b}$) satisfies $\langle\!\langle A^{\mathbf{x}} \rangle\!\rangle \bigcirc \varphi$

if A has a joint action with cost at most $\mathbf{b}$ such that all outcomes of this action satisfy φ. Similarly, $\langle\!\langle A^{\mathbf{x}} \rangle\!\rangle \Box \varphi$ is satisfied if A has a strategy to ensure that φ is always true, and the cost of any computation generated by this strategy is at most $\mathbf{b}$; $\langle\!\langle A^{\mathbf{x}} \rangle\!\rangle \varphi \, \mathcal{U} \, \psi$ is satisfied if A has a strategy to enforce ψ while maintaining the truth of φ, and the cost of this strategy is at most $\mathbf{b}$.

The models of the logic ParRB±ATL are resource-bounded concurrent game structures (RB-CGS) introduced in (Alechina et al. 2010), which are also the models of RB±ATL.

Definition 1. *A resource-bounded concurrent game structure 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 finite non-empty set of states.*

- *Π is a finite set of propositional variables and $\pi : \Pi \to \wp(S)$ is a truth assignment which associates each proposition in Π with a subset of states where it is true.*

- *Act is a non-empty set of actions which includes $idle$, and $d : S \times Agt \to \wp(Act)$ 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 $D(s) = d(s, a_1) \times \cdots \times d(s, a_n)$.*

- *$c : S \times Agt \times Act \to \mathbb{Z}^r$ is a partial function which maps a state s, and 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 $c(s, a, idle) = \mathbf{0}$ for all $s \in S$ and $a \in Agt$ where $\mathbf{0} = (0, \ldots, 0) \in \mathbb{N}^r$.*

- *$\delta : (s, \sigma) \mapsto S$ is a function that for every $s \in S$ and joint action $\sigma \in D(s)$ gives the state resulting from executing σ in s.*

Given an RB-CGS M, we denote the set of all infinite sequences of states (computations) by S^{ω} and the set of non-empty finite sequences of states by S^+. For a computation $\lambda = s_0 s_1 \ldots \in S^{\omega}$, we use the notation $\lambda[i] = s_i$ and $\lambda[i, j] = s_i \ldots s_j$ where $j \geq i \geq 0$.

Given an RB-CGS M and a state $s \in S$, a *joint action* by a coalition $A \subseteq Agt$ is a tuple $\sigma = (\sigma_a)_{a \in A}$ 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 denotes the joint action executed by A: $\sigma_A = (\sigma_a)_{a \in A}$. The set of all possible outcomes of a joint action $\sigma \in D_A(s)$ at state s is:

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

The cost of a joint action $\sigma \in D_A(s)$ is defined as:

$$cost(s, \sigma) = \sum_{a \in A} c(s, a, \sigma_a)$$

Given an RB-CGS M, a *strategy for a coalition* $A \subseteq Agt$ is a mapping $F_A : S^+ \to Act^{|A|}$ 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 consistent with F_A that start from s.

We use the usual point-wise notation for vector comparison and addition. In particular, $(b_1, \ldots, b_r) \leq (d_1, \ldots, d_r)$ iff $b_i \leq d_i$ for all $i \in \{1, \ldots, r\}$, and $(b_1, \ldots, b_r) + (d_1, \ldots, d_r) = (b_1 + d_1, \ldots, b_r + d_r)$.

For convenience, let us denote the set of possible resource bounds by $\mathbf{B} = \mathbb{N}^r$. Given a bound $\mathbf{b} \in \mathbf{B}$, a computation $\lambda \in out(s, F_A)$ is $\mathbf{b}$-consistent with F_A iff, for every $i \geq 0$,

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

Note that this definition implies that if agents start with resource allocation $\mathbf{b}$, then on every prefix of the computation the amount of resources they have is never below $\mathbf{0}$. Furthermore, the constraint on $\mathbf{b}$-consistent computations involves only action costs for agents from the coalition A (and not on the agents outside A).

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

Given an RB-CGS M, a state s of M, the truth of a ParRB$\pm$ATL formula φ with respect to M, s and assignment v from the set of variables to $\mathbb{N}$ is defined inductively on the structure of φ as follows:

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

Note that along the evaluation of formula, the assignment v does not vary, and the formulae are evaluated on states and not on pairs of the form $(s, \mathbf{b})$.

The two decision forms of the model-checking problem for ParRB$\pm$ATL are as follows.

Definition 2. *The following problem is the* unknown initial assignment model-checking problem *for ParRB$\pm$ATL.*

Input: $n, r \geq 1$ *(in unary), a ParRB$\pm$ATL(n,r) formula φ, a finite model M, and a state s.*

Question: *Is there an assignment v such that $M, s, v \models \varphi$?*

Definition 3. *The following problem is the* known initial assignment model-checking problem *for ParRB$\pm$ATL.*

Input: $n, r \geq 1$ *(in unary), a ParRB$\pm$ATL(n,r) formula φ, a finite model M, a state s, and an assignment v.*

Question: *Is it true that $M, s, v \models \varphi$?*

The function form of the model-checking problem is to *compute* the set of all satisfying assignments.

Definition 4. *The following problem is the* function form of the model-checking problem *for ParRB$\pm$ATL(n,r).*

Input: $n, r \geq 1$ *(in unary), a ParRB$\pm$ATL(n,r) formula φ, a finite model M, and a state s.*

Output: *Compute a formula γ describing the set of assignments v such that $M, s, v \models \varphi$, where γ is of the following form (for $b \in \mathbb{N}$):*

$$\gamma := x \geq b \mid \top \mid \bot \mid \neg\gamma \mid \gamma \wedge \gamma \mid \gamma \vee \gamma$$

Clearly, if the function form of the problem is solvable, then both model-checking problems for ParRB$\pm$ATL are decidable.

The following is an immediate consequence[1] of Theorem 8 in (Alechina et al. 2018):

Theorem 1. *The function form of the model-checking problem for ParRB$\pm$ATL is decidable.*

However, the upper bound on complexity for this problem is not known. The method for proving decidability of the function form of the model-checking problem for ParRB$\pm$ATL* is by reduction to parity games on single-sided VASS (Abdulla et al. 2013). While this can be used for ParRB$\pm$ATL model checking, it leads to a huge complexity bound (probably non primitive recursive, as structures close to Karp-Miller trees need to be constructed). In the next section, we give a direct algorithm for model checking ParRB$\pm$ATL.

Theorem 2. *The lower bound on the complexity of the known initial assignment model-checking problem for ParRB$\pm$ATL is 2EXPTIME.*

Proof. From the result that the model-checking problem for RB$\pm$ATL is 2EXPTIME-hard (Theorem 3, (Alechina et al. 2018)) when r is an input parameter.[2] Note that the proof of Theorem 3 in (Alechina et al. 2018) does not require infinite bounds, so the result also holds for RB$\pm$ATL without infinite bounds. It is straightforward to reduce the model-checking problem for RB$\pm$ATL to the known initial assignment model-checking problem for ParRB$\pm$ATL. $\square$

Upper Bound on the Complexity of the Function Form of the ParRB$\pm$ATL Model-Checking Problem

In this section, we first introduce the key concepts used in the model-checking algorithm for ParRB$\pm$ATL, and in the process show that Until and Box formulas of ParRB$\pm$ATL that do not contain nested modalities can be model-checked

---

[1] The set of assignments considered in (Alechina et al. 2018) is not of the form $Var \to \mathbb{N}$ but of the form $Var \to \mathbb{N} \cup \{\infty\}$, and the constraint formula may include $x = \infty$. This is motivated by the connection between ParRB$\pm$ATL* and RB$\pm$ATL where modalities have bounds which may contain ∞. That in turn was motivated by the desire to make RB$\pm$ATL a conservative extension of ATL. However, the proof of Theorem 8 in (Alechina et al. 2018) goes through for the case when assignments are of the form $Var \to \mathbb{N}$, which is what we are interested in this paper.

[2] It is EXPTIME-hard for fixed $r \geq 4$, see Corollary 1 in (Alechina et al. 2018).

in 2EXPTIME. We then state the model-checking algorithm and the general result on the upper bound on complexity of the function form of the model-checking problem for ParRB±ATL.

Preliminaries and Definitions

The proof of a 2EXPTIME upper bound for the function form of the ParRB±ATL model-checking problem builds on results for reachability games with unknown initial credit for alternating vector addition systems with states (Courtois and Schmitz 2014) and non-termination energy games with unknown initial credit (Jurdzinski, Lazić, and Schmitz 2015). These results roughly correspond to model checking formulas of the form $\langle\!\langle A^{\mathbf{x}} \rangle\!\rangle \phi \,\mathcal{U}\, \psi$ and $\langle\!\langle A^{\mathbf{x}} \rangle\!\rangle \Box \phi$, respectively, where ϕ and ψ do not in turn contain variables, i.e. these are propositional formulae.

First we introduce a notion of a *witness* for a successful A-strategy in enforcing a formula of the form $\langle\!\langle A^{\mathbf{x}} \rangle\!\rangle \phi \,\mathcal{U}\, \psi$. Informally, a witness is a finite tree with the root labelled $(s, \mathbf{b})$, representing the result of executing a successful A-strategy from s under assignment $\mathbf{x} \mapsto \mathbf{b}$ until ψ states are reached on all branches.

Definition 5. *Given a formula of the form $\langle\!\langle A^{\mathbf{x}} \rangle\!\rangle \phi \,\mathcal{U}\, \psi$, a state s in an RB-CGS M, and a vector $\mathbf{b} \in \mathbb{N}^r$, a finite tree with root $(s, \mathbf{b})$ is a* witness *for $M, s, \mathbf{x} \mapsto \mathbf{b} \models \langle\!\langle A^{\mathbf{x}} \rangle\!\rangle \phi \,\mathcal{U}\, \psi$ if*

- *all nodes of the tree are of the form $(s', \mathbf{b}')$ where s' is a state (resulting from executing the joint action in the state of the parent node) and $\mathbf{b}'$ the vector of resource amounts A have in s' as a result, and $\mathbf{b}' \geq 0$,*
- *in all leaf nodes $(s', \mathbf{b}')$ of the tree, s' satisfies ψ (since ψ is propositional, this can be determined using only the truth assignment π),*
- *in all non-leaf nodes $(s', \mathbf{b}')$ of the tree, s' satisfies ϕ (again, only the truth assignment π is needed here).*

In the first point of the definition, an edge $(s, \mathbf{b}) \mapsto (s', \mathbf{b}')$ is present in the tree because of some joint action $\sigma \in D(s)$, such that $\delta(s, \sigma) = s'$, and for all $i \in [1, r]$, we have $b'[i] = b[i] - (\Sigma_{a \in A}\, c(s, a, \sigma(a))[i])$. Note that in the expression above, the value $c(s, a, \sigma(a))[i]$ occurs negatively.

Observe that a witness corresponds to a perfect recall $\mathbf{b}$-strategy for A, because after reaching the leaf nodes satisfying ψ, the coalition can forever execute the *idle* action that consumes no resources.

A witness for $M, s, \mathbf{x} \mapsto \mathbf{b} \models \langle\!\langle A^{\mathbf{x}} \rangle\!\rangle \Box \phi$ is defined similarly.

Definition 6. *A* witness *for $M, s, \mathbf{x} \mapsto \mathbf{b} \models \langle\!\langle A^{\mathbf{x}} \rangle\!\rangle \Box \phi$ is a finite tree with root $(s, \mathbf{b})$ where in every node the state satisfies ϕ, all resource allocations in the nodes are non-negative, and for each leaf node $(s', \mathbf{b}')$ there is exactly one other node $(s', \mathbf{b}'')$ on the same branch with the same state s' and $\mathbf{b}'' \leq \mathbf{b}'$.*

In other words, a witness for $\langle\!\langle A^{\mathbf{x}} \rangle\!\rangle \Box \phi$ describes the initial prefix of a successful strategy for satisfying $\langle\!\langle A^{\mathbf{x}} \rangle\!\rangle \Box \phi$. It can be extended to a perfect recall $\mathbf{b}$-strategy for A since it describes how to enforce a non-consuming loop on any computation.

Clearly, if there is a witness for $M, s, \mathbf{x} \mapsto \mathbf{b} \models \phi$, then there is also a witness for $M, s, \mathbf{x} \mapsto \mathbf{b}' \models \phi$, where $\mathbf{b} \leq \mathbf{b}'$ (the same strategy will work for a greater initial resource allocation). We will also talk about w being a witness for ϕ being true in M and s if there is some assignment v such that w is a witness for $M, s, v \models \phi$, and about w being a witness for ϕ being true in M if there is some state s and assignment v such that w is a witness for $M, s, v \models \phi$.

A norm of a vector $\mathbf{b}$ is the largest component of the vector (by absolute value), $max(|\mathbf{b}[i]|)$. This notion can be extended to the norm of a finite set of vectors. The largest value occurring in a description of an RB-CGS M is the norm of the set of costs of actions. We will denote this number by max_M. Similarly, we can talk about the norm of a witness w, $\|w\|$. Let us call a witness w *norm-minimal* for some model-checking instance if there is no witness w' for the same model-checking instance with $\|w'\| < \|w\|$.

Proposition 1. *If a witness for $\langle\!\langle A^{\mathbf{x}} \rangle\!\rangle \Box \phi$ being true in M, s exists, then there exists a witness w with $\|w\| \leq (4|S| \cdot max_M)^{2(r+2)^3}$.*

Proof. The proof uses two results from (Jurdzinski, Lazić, and Schmitz 2015). The first result (Lemma 3.1) is that if Player 1 has a winning strategy in a bounded energy game (ensuring non-termination, with counters remaining non-negative *and* below a certain bound), then Player 1 has a strategy where the norm of any computation is bounded by $(4|S| \cdot max_M)^{2(r+2)^3}$. The second result (Proposition 2.4) is that Player 1 has a winning strategy in a non-termination energy game if and only if Player 1 has a winning strategy in a bounded energy game on the same graph with resource-decrementing loops added to every state. To construct the winning strategy for the bounded game, resource-decrementing loops can be used to keep the resource amounts below the upper bound, while the transitions in the non-termination energy game simply ignore decrementing loops (in particular, they may involve incrementing some resource forever in a loop). If all 'useful' (non-resource-decrementing-loop transitions) in the bounded game can be accomplished without more than $(4|S| \cdot max_M)^{2(r+2)^3}$ resources on any counter up to the point the 'final' loop is repeatedly executed, then the same applies to the original energy game, and hence to a witness for $\langle\!\langle A^{\mathbf{x}} \rangle\!\rangle \Box \phi$ being true in M, s. $\qquad\square$

Proposition 2. *If a witness for $\langle\!\langle A^{\mathbf{x}} \rangle\!\rangle \psi_1 \,\mathcal{U}\, \psi_2$ being true in M, s exists, then there exists a witness w with $\|w\| \leq (4(|S| + 1) \cdot max_M)^{2(r+3)^3}$.*

Proof. By contradiction, assume that in some model M, a norm-minimal witness w for $\langle\!\langle A^{\mathbf{x}} \rangle\!\rangle \psi_1 \,\mathcal{U}\, \psi_2$ being true in M, s has norm $\|w\| > (4(|S| + 1) \cdot max_M)^{2(r+3)^3}$. Produce a model M' from M as follows. If any joint action by A which is involved in w is composed of only *idle* actions by all agents in A, add to M another 0 cost action α and replace *idle* with α in w. Add an extra resource type $r+1$, and let all

actions by agents in A involved in w (apart from the *idle* action) consume one unit of $r+1$. Redirect all other transitions (not involved in w) to an additional new state s' not satisfying ψ_2. Note that M' contains a witness w' corresponding to w but with extra costs on the $(r+1)$th resource, and w' is still a minimal (and the only) witness for $\langle\!\langle A^{\mathbf{x}}\rangle\!\rangle\psi_1\,\mathcal{U}\,\psi_2$. This is because the only state in M' that is not part of w' is s', and a ψ_2 state is not accessible from s'. Note also that w' does not contain any non-resource-consuming loops on the branches leading to ψ_2 states because of the costs on the $(r+1)$th resource. Now to complete the transformation of M into M', we add a new propositional variable p that holds in all w' states but not in s', and add $\mathbf{0}$ cost self-loops by the *idle* action to all ψ_2 states. Consider the formula $\langle\!\langle A^{\mathbf{x}}\rangle\!\rangle\Box p$. The only way to satisfy it in M', s is to use a strategy corresponding to w'. (This is because the only way to reach a non-resource-consuming loop satisfying p is to reach a ψ_2 state, and w' is a minimal witness to reaching a ψ_2 state while satisfying ψ_1, hense while satisfying p). The norm of w' is still the same as the norm of w (without loss of generality, we assume that the costs on $r+1$ are less than $\|w\|$), so the norm-minimal witness for $\langle\!\langle A^{\mathbf{x}}\rangle\!\rangle\Box p$ has the norm strictly greater than $(4(|S|+1)\cdot max_M)^{2(r+3)^3}$, that is, strictly greater than $(4|S'|)\cdot max_{M'})^{2(r'+2)^3}$, where $|S'| = |S| + 1$ is the number of states in M' and $r' = r+1$ is the number of resource types in M'. This contradicts Proposition 1. $\qquad\square$

Given that the norm of a norm-minimal witness is bounded by $B_M = (4(|S| + 1) \cdot max_M)^{2(r+3)^3}$, the height of a witness is bounded by an exponential in r and max_M (since the model is encoded in binary). This means that all such potential witnesses can be generated and checked by an alternating Turing Machine in EXPSPACE. As pointed out in (Courtois and Schmitz 2014), this gives a naive algorithm for computing the Pareto frontier (all non-dominated initial resource allocations): all potential witnesses with resource allocations at the root starting from $(0, \ldots, 0)$ and up to $(B_M, \ldots, B_M)$ can be explored in turn in AEXPSPACE, and the non-dominated values $\mathbf{b}$ saved as constraints $\mathbf{x} \geq \mathbf{b}$. Since ASPACE$(f(n))$ = DTIME$(2^{O(f(n))})$, AEXPSPACE=2EXPTIME, and this algorithm produces a description of satisfying assignments for non-nested Box and Until formulas in 2EXPTIME. In the next section, we show that this upper bound applies to the whole language.

Model-Checking Algorithm

In this section, we present a 2EXPTIME algorithm to solve the function form of the ParRB±ATL model-checking problem. Given a model M and a ParRB±ATL formula ϕ with variables $x_1, \ldots, x_n$, the algorithm returns a set of pairs $\|\phi\|_M$ of the form (s, γ) where s is a state and γ a constraint formula defining the set of assignments v to $x_1, \ldots, x_n$ such that $M, s, v \models \phi$.

Note that γ could be equivalent to $\bot$ (unsatisfiable), in which case there is no assignment v such that $M, s, v \models \phi$. We will say that γ is satisfiable iff there is at least one assignment v satisfying the constraints in γ.

To formulate the cases for $\langle\!\langle A^{\mathbf{x}}\rangle\!\rangle\psi_1\,\mathcal{U}\,\psi_2$ and $\langle\!\langle A^{\mathbf{x}}\rangle\!\rangle\Box\psi$, we need to redefine the notion of a witness relative to

$\|\psi_1\|_M, \|\psi_2\|_M$, and $\|\psi\|_M$. A *constrained witness* for $\langle\!\langle A^{\mathbf{x}}\rangle\!\rangle\psi_1\,\mathcal{U}\,\psi_2$ being true in M, s is a finite tree where each leaf is a node of the form $(s', \mathbf{b}')$ (for some $\mathbf{b}'$) where s' is such that $(s', \gamma') \in \|\psi_2\|_M$ and γ' is satisfiable. All other nodes are of the form $(s', \mathbf{b}')$ where $(s', \gamma') \in \|\psi_1\|_M$ and γ' is satisfiable. A constrained witness w for $\langle\!\langle A^{\mathbf{x}}\rangle\!\rangle\psi_1\,\mathcal{U}\,\psi_2$ being true in M, s is exactly like a (non-constrained) witness for $\langle\!\langle A^{\mathbf{x}}\rangle\!\rangle\psi_1\,\mathcal{U}\,\psi_2$ being true in M, s, but instead of requiring states in the nodes to satisfy ψ_1 or ψ_2, we require them to be in $\|\psi_i\|_M$ with a satisfiable constraint formula γ'. The formula γ' describes constraints on resources required to satisfy ψ_1 or ψ_2. If ψ_i is a propositional formula, then γ' can only be of the form $\top$, otherwise it may be any satisfiable constraint, for example, $\mathbf{y} < \mathbf{1}$. These γ' constraints are going to be used to construct the formula $\gamma(w)$ defined below.[3]

Similarly, a constrained witness for $\langle\!\langle A^{\mathbf{x}}\rangle\!\rangle\Box\psi$ is obtained from a witness by replacing the notion of a state s' satisfying ψ with a requirement that $(s', \gamma') \in \|\psi\|_M$ and γ' is satisfiable.

We call a constrained witness w with the root $(s, \mathbf{b})$ *non-dominated* if it is norm-minimal and there is no constrained witness w' with root $(s, \mathbf{b}')$ such that $\mathbf{b}' < \mathbf{b}$.

We denote the set of non-dominated constrained witnesses for (M, s, ψ) by $W(M, s, \psi)$. This set is finite and each witness has height bounded by $|S| \cdot (B_M + 1)^r$ by Propositions 1 and 2 (because no two nodes on the same branch have the same state and resource vector, and the maximal value in resource vector is B_M).

Given a constrained witness w, we define a formula $\gamma(w)$ describing constraints on the nodes in w. Intuitively, for formulas of the form $\langle\!\langle A^{\mathbf{x}}\rangle\!\rangle\psi_1\,\mathcal{U}\,\psi_2$ and $\langle\!\langle A^{\mathbf{x}}\rangle\!\rangle\Box\psi$, the constraint on $\mathbf{x}$ is $\mathbf{x} \geq \mathbf{b}$ if the root of the witness is $(s, \mathbf{b})$. There are however additional provisos. If some of the variables in $\mathbf{x}$ occur in proper subformulas of that formula, then there will be some more constraints on $\mathbf{x}$ arising from the constraints on the occurrences in proper subformulas. Also, constraint $\mathbf{x} \geq \mathbf{b}$ is only tight (in the sense of $M, s, v \models \langle\!\langle A^{\mathbf{x}}\rangle\!\rangle\psi_1\,\mathcal{U}\,\psi_2$ iff $v \models \mathbf{x} \geq \mathbf{b}$), if the witness is non-dominated. In addition to $\mathbf{x}$, ψ_1, ψ_2 and ψ may have other strategic modalities with variables $\mathbf{y}$. The purpose of the formula $\gamma(w)$ is to extract the constraint on $\mathbf{y}$ from the witness, by combining constraint formulas for the states in nodes of the witness.

For a constrained witness w for $\langle\!\langle A^{\mathbf{x}}\rangle\!\rangle\psi_1\,\mathcal{U}\,\psi_2$ with a root $(s, \mathbf{b})$,

$$\gamma(w) = \mathbf{x} \geq \mathbf{b} \wedge$$

$$\bigwedge_{s' \in leaves(w), (s', \gamma') \in \|\psi_2\|_M} \gamma' \wedge$$

$$\bigwedge_{s' \in -leaves(w), (s', \gamma') \in \|\psi_1\|_M} \gamma'$$

where $s' \in leaves(w)$ if $(s', \mathbf{b}')$ is a leaf node of w and $s' \in -leaves(w)$ if $(s', \mathbf{b}')$ is a non-leaf node of w.

---

[3]Note that there is no connection between γ' and $\mathbf{b}'$. As in the (non-constrained) witness, $\mathbf{b}'$ corresponds to values remaining on the original resource assignement to $\mathbf{x}$, that is, the resources used by the A-strategy to satisfy the formula $\langle\!\langle A^{\mathbf{x}}\rangle\!\rangle\psi_1\,\mathcal{U}\,\psi_2$.

Algorithm 1 Computing $\|\phi_0\|$

1: **function** PARRB$\pm$ATL-MC(M, ϕ_0)
2:     **for** $\phi' \in Sub(\phi_0)$ **do**
3:         **case** $\phi' = p$
4:             $\|p\|_M \leftarrow \{(s, \top) : s \in \pi(p)\} \cup \{(s, \bot) : s \notin \pi(p)\}$
5:         **case** $\phi' = \neg\psi$
6:             $\|\neg\psi\|_M \leftarrow \{(s, \neg\gamma) : (s, \gamma) \in \|\psi\|_M\}$
7:         **case** $\phi' = \psi_1 \wedge \psi_2$
8:             $\|\psi_1 \wedge \psi_2\|_M \leftarrow \{(s, \gamma_1 \wedge \gamma_2) : (s, \gamma_1) \in \|\psi_1\|_M \wedge (s, \gamma_2) \in \|\psi_2\|_M\}$
9:         **case** $\phi' = \psi_1 \vee \psi_2$
10:            $\|\psi_1 \vee \psi_2\|_M \leftarrow \{(s, \gamma_1 \vee \gamma_2) : (s, \gamma_1) \in \|\psi_1\|_M \wedge (s, \gamma_2) \in \|\psi_2\|_M\}$
11:         **case** $\phi' = \langle\!\langle A^{\mathbf{x}} \rangle\!\rangle \bigcirc \psi$
12:            $\|\langle\!\langle A^{\mathbf{x}} \rangle\!\rangle \bigcirc \psi\|_M \leftarrow \{(s, \gamma) : \gamma = \bigvee_{\sigma \in D_A(s)} (\mathbf{x} \geq cost(\sigma, s) \wedge \bigwedge_{s' \in out(s, \sigma), (s', \gamma') \in \|\psi\|_M} \gamma')\}$
13:         **case** $\phi' = \langle\!\langle A^{\mathbf{x}} \rangle\!\rangle \psi_1 \,\mathcal{U}\, \psi_2$
14:            $\|\langle\!\langle A^{\mathbf{x}} \rangle\!\rangle \psi_1 \,\mathcal{U}\, \psi_2\| \leftarrow \{(s, \gamma) : \gamma = \bigvee_{w \in W(M, s, \langle\!\langle A^{\mathbf{x}} \rangle\!\rangle \psi_1 \,\mathcal{U}\, \psi_2)} \gamma(w)\}$
15:         **case** $\phi' = \langle\!\langle A^{\mathbf{x}} \rangle\!\rangle \Box \psi$
16:            $\|\langle\!\langle A^{\mathbf{x}} \rangle\!\rangle \Box \psi\| \leftarrow \{(s, \gamma) : \gamma = \bigvee_{w \in W(M, s, \langle\!\langle A^{\mathbf{x}} \rangle\!\rangle \Box \psi)} \gamma(w)\}$
17:     **return** $\|\phi_0\|_M$

For a constrained witness w for $\langle\!\langle A^{\mathbf{x}} \rangle\!\rangle \Box \psi$ with a root $(s, \mathbf{b})$,

$$\gamma(w) = \mathbf{x} \geq \mathbf{b} \wedge \bigwedge_{s' \in nodes(w), (s', \gamma') \in \|\psi\|_M} \gamma'$$

where $s' \in nodes(w)$ if $(s', \mathbf{b}')$ for some $\mathbf{b}'$ is a node of w.

For each subformula ϕ' of ϕ_0 in increasing order of complexity, Algorithm 1 computes $\|\phi'\|_M$.

Theorem 3. *On the input of M, ϕ_0, Algorithm 1 returns a set of pairs $\|\phi_0\|_M$ such that $(s, \gamma) \in \|\phi_0\|_M$ if, and only if: for all v $M, s, v \models \phi_0$ iff $v \models \gamma$.*

Proof. The proof is by induction on the complexity of ϕ_0. The inductive hypothesis is that $(s, \gamma) \in \|\phi'\|_M$ iff $(M, s, v \models \phi'$ iff $v \models \gamma)$.

Clearly this holds for atomic propositions: $(s, \top) \in \|p\|_M$ iff $M, s \models p$, by line 4 of Algorithm 1. For booleans, the conditions on assignments are also straightforward (lines 6, 8, 10).

Consider $\langle\!\langle A^{\mathbf{x}} \rangle\!\rangle \bigcirc \psi$. $M, s, v \models \langle\!\langle A^{\mathbf{x}} \rangle\!\rangle \bigcirc \psi$ iff v assigns to $\mathbf{x}$ the amount of resources sufficient to execute an action σ such that all outcomes of that action satisfy ψ under v. (This σ is the first action in a successful strategy, but just as in the case of Until, all subsequent actions in the strategy can be chosen to be $idle$, which means the strategy will not require more than $v(\mathbf{x})$ resources.) By the inductive hypothesis, all outcomes of σ satisfy ψ under v iff for every state s' in $out(s, \sigma)$ such that $(s', \gamma') \in \|\psi\|_M$, $v \models \gamma'$, that is, iff $v \models \bigwedge_{s' \in out(s, \sigma), (s', \gamma') \in \|\psi\|_M} \gamma'$. Hence $M, s, v \models \langle\!\langle A^{\mathbf{x}} \rangle\!\rangle \bigcirc \psi$ iff for some executable σ, $v \models \bigwedge_{s' \in out(s, \sigma), (s', \gamma') \in \|\psi\|_M} \gamma'$. This is exactly what line 12 of Algorithm 1 requires for $\|\langle\!\langle A^{\mathbf{x}} \rangle\!\rangle \bigcirc \psi\|_M$.

$M, s, v \models \langle\!\langle A^{\mathbf{x}} \rangle\!\rangle \psi_1 \,\mathcal{U}\, \psi_2$ iff there is a strategy such that each computation on this strategy satisfies ψ_1 under v until it satisfies ψ_2 under v. This holds iff there is a

witness w such that initial amount of resources is sufficient to execute the corresponding strategy and all nodes in the witness satisfy the corresponding formulas under v. By the inductive hypothesis, this holds iff for some non-dominated witness rooted in $(s, \mathbf{b})$ with $\mathbf{b} \leq v(\mathbf{x})$, $v \models \bigvee_{w \in W(M, s, \langle\!\langle A^{\mathbf{x}} \rangle\!\rangle \psi_1 \,\mathcal{U}\, \psi_2)} \gamma(w)$, which is the constraint formula on line 14 of Algorithm 1.

$M, s, v \models \langle\!\langle A^{\mathbf{x}} \rangle\!\rangle \Box \psi$ holds iff there is a strategy such that each computation on this strategy reaches a loop where no resources are consumed and every state on this computation satisfies ψ under v. By the inductive hypothesis, this holds iff for some non-dominated witness w rooted in $(s, \mathbf{b})$ with $\mathbf{b} \leq v(\mathbf{x})$, v satisfies $\gamma(w)$, iff $v \models \bigvee_{w \in W(M, s, \langle\!\langle A^{\mathbf{x}} \rangle\!\rangle \Box \psi_1)} \bigwedge_{s' \in nodes(w), (s', \gamma') \in \|\psi\|_M} \gamma'$, which is the constraint formula on line 16 of Algorithm 1. $\square$

Observe that if ϕ is a positive formula (does not contain modalities in the scope of negations), then the corresponding constraint formula γ only has subformulas of the form $\mathbf{x} \geq \mathbf{b}$. Since the constraint formula describes non-dominated witnesses, it specifies the *optimal* values of resource parameters (the Pareto frontier).

Theorem 4. *The upper bound on the complexity of the function form of the model-checking problem for ParRB$\pm$ATL is 2EXPTIME.*

Proof. The set of non-dominated witnesses $W(M, s, \phi')$ for each subformula ϕ' of ϕ_0 can be computed in AEXSPACE. Hence the whole algorithm can be executed in AEXSPACE which is equivalent to 2EXPTIME. $\square$

Conclusions and Future Work

Algorithm 1 returns a set of states satisfying a ParRB$\pm$ATL formula together with a constraint formula describing the set of assignments to resource variables for which the

ParRB±ATL formula is true in that state. The algorithm can be easily modified to return the witness (finite representation of the strategy). The complexity of the algorithm is very high (2EXPTIME). However, only the number of resource types and the log of the largest cost in the model ($log\ max_M$) are in the exponent; the running time is polynomial in the number of states and transitions in the model.

In future work we plan to implement the algorithm. Another direction of future work is a parametrised version of the fragment of Resource Agent Logic (RAL) with a decidable model-checking problem that was described in (Alechina et al. 2017a). In ParRB±ATL, the resources needed for satisfying each strategic modality are independent, even if nested subformulas share variables. For example, $\langle\!\langle A^x \rangle\!\rangle \top \,\mathcal{U}\,(q \wedge \langle\!\langle A^x \rangle\!\rangle \top \,\mathcal{U}\, p)$ means that there is a value for x, e.g., 5, such that given 5 units of the resource, A can enforce a state where q holds, and from where, given 5 units of the resource, A can enforce p. The second, nested, strategy does not use the amount of resource remaining from the execution of the first strategy, but starts from scratch with x resources, whatever value x happens to have. On the other hand, in RAL, there is an extra annotation for strategic modalities that means 'with the remaining resources'. For example, $\langle\!\langle A^x \rangle\!\rangle \top \,\mathcal{U}\,(q \wedge \langle\!\langle A^\downarrow \rangle\!\rangle \top \,\mathcal{U}\, p)$ means that there is a value for x, such that if A have this amount of resources, then they could enforce a state where q holds, and *with remaining resources*, they could enforce p. This logic can be used to solve model-checking problems that involve nested goals that share a single allocation of resources.

References

Abdulla, P.; Mayr, R.; Sangnier, A.; and Sproston, J. 2013. Solving Parity Games on Integer Vectors. In *CONCUR'13*, volume 8052 of *Lecture Notes in Computer Science*, 106–120. Springer.

Alechina, N.; Logan, B.; Nguyen, H. N.; and Rakib, A. 2009. A logic for coalitions with bounded resources. In Boutilier, C., ed., *Proceedings of the Twenty First International Joint Conference on Artificial Intelligence (IJCAI 2009)*, volume 2, 659–664. AAAI Press.

Alechina, N.; Logan, B.; Nga, N. H.; and Rakib, A. 2010. Resource-bounded alternating-time temporal logic. In van der Hoek, W.; Kaminka, G.; Lespérance, Y.; Luck, M.; and Sen, S., eds., *Proceedings of the Ninth International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, 481–488. IFAAMAS.

Alechina, N.; Logan, B.; Nguyen, H. N.; and Rakib, A. 2011. Logic for coalitions with bounded resources. *Journal of Logic and Computation* 21(6):907–937.

Alechina, N.; Bulling, N.; Logan, B.; and Nguyen, H. N. 2017a. The virtues of idleness: A decidable fragment of resource agent logic. *Artificial Intelligence* 245:56–85.

Alechina, N.; Logan, B.; Nguyen, H. N.; and Raimondi, F. 2017b. Model-checking for resource-bounded ATL with production and consumption of resources. *Journal of Computer and System Sciences* 88:126–144.

Alechina, N.; Bulling, N.; Demri, S.; and Logan, B. 2018. On the complexity of resource-bounded logics. *Theoretical Computer Science* 750:69–100.

Brázdil, T.; Jančar, P.; and Kucera, A. 2010. Reachability games on extended vector addition systems with states. In *ICALP'10*, volume 6199 of *Lecture Notes in Computer Science*, 478–489. Springer.

Bulling, N., and Farwer, B. 2010a. Expressing properties of resource-bounded systems: The logics RBTL and RBTL*. In *Computational Logic in Multi-Agent Systems - 10th International Workshop, CLIMA X, Revised Selected and Invited Papers*, volume 6214 of *Lecture Notes in Computer Science*, 22–45.

Bulling, N., and Farwer, B. 2010b. On the (un-)decidability of model checking resource-bounded agents. In Coelho, H.; Studer, R.; and Wooldridge, M., eds., *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI 2010)*, 567–572. IOS Press.

Bulling, N., and Goranko, V. 2013. How to be both rich and happy: Combining quantitative and qualitative strategic reasoning about multi-player games (extended abstract). In Mogavero, F.; Murano, A.; and Vardi, M. Y., eds., *Proceedings of the 1st International Workshop on Strategic Reasoning (SR 2013)*, volume 112 of *Electronic Proceedings in Theoretical Computer Science*, 33–41.

Courtois, J., and Schmitz, S. 2014. Alternating vector addition systems with states. In *39th International Symposium on Mathematical Foundations of Computer Science (MFCS'14)*, volume 8634 of *Lecture Notes in Computer Science*, 220–231. Springer.

Della Monica, D., and Lenzi, G. 2012. On a priced resource-bounded alternating μ-calculus. In Filipe, J., and Fred, A. L. N., eds., *Proceedings of the 4th International Conference on Agents and Artificial Intelligence (ICAART 2012)*, 222–227. SciTePress.

Della Monica, D.; Napoli, M.; and Parente, M. 2011. On a logic for coalitional games with priced-resource agents. *Electronic Notes in Theoretical Computer Science* 278:215–228.

Della Monica, D.; Napoli, M.; and Parente, M. 2013. Model checking coalitional games in shortage resource scenarios. In Puppis, G., and Villa, T., eds., *Proceedings of the Fourth International Symposium on Games, Automata, Logics and Formal Verification (GandALF 2013)*, volume 119 of *Electronic Proceedings in Theoretical Computer Science*, 240–255.

Juhl, L.; Larsen, K.; and Raskin, J.-F. 2013. Optimal bounds for multiweighted and parametrised energy games. In *Theories of Programming and Formal Methods - Essays Dedicated to Jifeng He on the Occasion of His 70th Birthday*, volume 8051 of *Lecture Notes in Computer Science*, 244–255. Springer.

Jurdzinski, M.; Lazić, R.; and Schmitz, S. 2015. Fixed-dimensional energy games are in pseudo-polynomial time. In *Proceedings of ICALP 2015*, 260–272.