Alternating Vector Addition Systems with States

Jean-Baptiste Courtois, Sylvain Schmitz

ALTERNATING VECTOR ADDITION SYSTEMS WITH STATES

JEAN-BAPTISTE COURTOIS AND SYLVAIN SCHMITZ

Abstract. Alternating vector addition systems are obtained by equipping vector addition systems with states (VASS) with 'fork' rules, and provide a natural setting for infinite-arena games played over a VASS. Initially introduced in the study of propositional linear logic, they have more recently gathered attention in the guise of *multi-dimensional energy* games for quantitative verification and synthesis.

We show that establishing who is the winner in such a game with a state reachability objective is 2-ExpTime-complete. As a further application, we show that the same complexity result applies to the problem of whether a VASS is simulated by a finite-state system.

Keywords. VASS, energy game, simulation game, Pareto frontier

1. Introduction

Vector addition systems with states (VASS) allow to model systems manipulating multiple discrete resources, for instance bank accounts balances or numbers of processes running concurrently. Extending their definition to two-players games is both a very natural endeavour and a tricky problem: the most immediate definition, where both players can freely update the vector values, leads to an undecidable game even with the simplest winning condition, namely control state reachability [1].

Facing this difficulty, one might expect to see a flurry of competing definitions for VASS games that would retain decidability through various restrictions. Surprisingly, this is not really the case: if there is indeed a large number of denominations (e.g. B-VASS games [19], Z-reachability games [5], multi-dimensional energy games [7]), Abdulla, Mayr, Sangnier, and Sproston [2] noted last year that they all defined essentially the same *asymmetric* class of games, where one player is restricted and cannot update the vector values.

Our contention in this paper is that so many different people coming up independently with the same model is not a coincidence, but a sure sign of a fundamental idea deserving investigation in its own right. We find further arguments in our own initial interest in such games, which comes from the study of simulation problems between Petri nets and finite-state systems [12, 15] where they arise naturally. Furthermore the model was already explicitly defined in the '90s in the study of substructural logics [16, 13, 22], and appears implicitly in recent proofs of complexity lower bounds in [9, 3].

Work funded in part by the ANR grant 11-BS02-001-01 ReacHard.

We show in this paper that determining the winner of an asymmetric VASS game with a state reachability objective is 2-ExpTime-complete. We extend for this well-known techniques by Rackoff [18] and Lipton [17] used to establish the complexity of VASS problems, see sections 3 and 4. We also provide refined bounds when the dimension of the problem is fixed, and show how to compute the *Pareto frontier* for such games.

Perhaps more importantly than those technical contributions, we propose a single, simple definition for alternation in VASS by way of 'fork' rules in Section 2, for which the complexity analyses of sections 3 and 4 are relatively easy, and establish it as a pivotal definition for VASS games. Indeed, we relate it to energy games in Section 5 (following [2]) and to regular simulation problems for VASS in Section 6. Our lower bound improves on all the published bounds for those problems, including the ExpSpace-hardness of simulations between *basic parallel processes* and finite-state processes due to Lasota [15]. Our upper bound applies to the simulation of Petri nets by finite-state systems, for which only decidability was known [12].

2. Alternating VASS

Alternating VASS were originally called 'and-branching' counter machines by Lincoln, Mitchell, Scedrov, and Shankar [16], and were introduced to prove the undecidability of propositional linear logic. Kanovich [13] later identified a fragment of linear logic, called the $(!, \oplus)$-Horn fragment, that captures exactly alternation in VASS, and adopted a game viewpoint. Alternating VASS were also instrumental in establishing the Ackermann-completeness of the conjunctive-implicational fragment of relevance logic [22]. As discussed in sections 5 and 6, this class of systems has since reappeared in other contexts, which motivates its study in earnest.

2.1. **Basic Definitions.** An *alternating vector addition system with states* (AVASS) is syntactically a tuple $\mathcal{A} = \langle Q, d, T_u, T_f \rangle$ where Q is a finite set of *states*, d is a *dimension* in $\mathbb{N}$, and $T_u \subseteq Q \times \mathbb{Z}^d \times Q$ and $T_f \subseteq Q^3$ are respectively finite sets of *unary* and *fork* rules. We denote unary rules $(q, \mathbf{u}, q_1)$ in T_u with $\mathbf{u}$ in $\mathbb{Z}^d$ by '$q \xrightarrow{\mathbf{u}} q_1$' and fork rules (q, q_1, q_2) in T_f by '$q \to q_1 \wedge q_2$.' A *vector addition system with states* (VASS) is an AVASS with $T_f = \emptyset$.

2.1.1. *Deduction Semantics.* Given an AVASS, its semantics is defined by a deduction system over *configurations* $(q, \mathbf{v})$ in $Q \times \mathbb{N}^d$: if $q \xrightarrow{\mathbf{u}} q_1$ and $q \to q_1 \wedge q_2$ are rules, then

$$\frac{q, \mathbf{v}}{q_1, \mathbf{v} + \mathbf{u}}\ \text{unary} \qquad \frac{q, \mathbf{v}}{q_1, \mathbf{v} \quad q_2, \mathbf{v}}\ \text{fork}$$

where '+' denotes component-wise addition in $\mathbb{N}^d$, and implicitly $\mathbf{v} + \mathbf{u}$ has no negative component, i.e. is in $\mathbb{N}^d$. Such a deduction system can be employed either *top-down* or *bottom-up* depending on the decision problems at hand (as with tree automata).

When working with finite deduction trees t, we define the *height* $h(t)$ of t as the maximal length among all its branches. A *(multi)-context* C is a finite tree with n distinguished leaves labelled with n distinct variables

$x_1, \ldots, x_n$; $C[t_1, \ldots, t_n]$ then denotes the tree obtained by substituting for each $1 \leq j \leq n$ the tree t_j for the variable x_j.

2.1.2. *Game Semantics.* The top-down direction of the deduction semantics allows for potentially infinite deduction trees, and defines in a natural way an *asymmetric* VASS game as defined by Kanovich [13] and later by Raskin et al. [19]. Two players, 'Controller' and 'Environment', play over the infinite arena $Q \times \mathbb{N}^d$. In a current configuration $(q, \mathbf{v})$, Controller chooses among the applicable rules in $T_u \cup T_f$. In case of a unary rule $q \xrightarrow{\mathbf{u}} q'$, the next configuration is $(q', \mathbf{v}+\mathbf{u})$, where by assumption $\mathbf{v}+\mathbf{u} \geq \mathbf{0}$ where '$\mathbf{0}$' denotes the null vector in $\mathbb{N}^d$. In case of a fork rule $q \to q_1 \wedge q_2$, Environment then chooses which branch of the deduction tree to explore, i.e. chooses between $(q_1, \mathbf{v})$ and $(q_2, \mathbf{v})$ as the next configuration. Various winning conditions on such plays $(q_0, \mathbf{v}_0), (q_1, \mathbf{v}_1), \ldots$ can then be envisioned, and correspond to conditions that must be satisfied by all the branches of a deduction tree. As shown by Abdulla et al. [2], such asymmetric games are closely related to *multi-dimensional energy games* [7, 5], see Section 5.

2.2. **Decision Problems and Complexity.** We assume when deriving complexity bounds a binary encoding of vectors in $\mathbb{Z}^d$. That is, letting $\|\mathbf{u}\|_\infty \stackrel{\text{def}}{=} \max_{1 \leq i \leq d} |\mathbf{u}(i)|$ denote the norm of the vector $\mathbf{u}$ and defining $\|T_u\|_\infty \stackrel{\text{def}}{=} \max_{(q,\mathbf{u},q') \in T_u} \|\mathbf{u}\|_\infty$the norm of a set of unary transitions, then the size of an AVASS $\langle Q, d, T_u, T_f \rangle$ depends polynomially on the *bitsize* $\log(\|T_u\|_\infty + 1)$. Note that we can reduce in logarithmic space by standard techniques all our decision problems to work with a set of unary rules T'_u with effects $\mathbf{u} = \mathbf{e}_i$ or $\mathbf{u} = -\mathbf{e}_i$—where '$\mathbf{e}_i$' is the unit vector with '1' in coordinate i and '0' everywhere else—, but this comes at the expense of an increase in the dimension by a factor of $\log(\|T_u\|_\infty + 1)$.

2.2.1. *Reachability.* The decision problem that originally motivated the definition of AVASS by Lincoln et al. [16] is *reachability*: given an AVASS $\langle Q, d, T_u, T_f \rangle$ and two states q_r and q_ℓ in Q, does there exist a deduction tree with root labelled by $(q_r, \mathbf{0})$ and every leaf labelled by $(q_\ell, \mathbf{0})$? Equivalently, does Controller have a strategy that ensures that a play starting in $(q_r, \mathbf{0})$ eventually visits $(q_\ell, \mathbf{0})$?

This problem is easily seen to be Σ^0_1-complete:

Fact 2.1 (Lincoln et al. [16])**.** *Reachability in AVASS is undecidable.*

Proof Idea. By a reduction from the halting problem in Minsky machines: note that a zero test $q \xrightarrow{c?=0} q'$ on a counter c can be emulated through a fork $q \to q' \wedge q_c$, where unary rules $q_c \xrightarrow{-\mathbf{e}_{c'}} q_c$ for all $c' \neq c$ allow to empty the counters different from c, and a last unary rule $q_c \xrightarrow{\mathbf{0}} q_\ell$ to the single target state allows to check that c was indeed equal to zero. □

2.2.2. *State Reachability.* Our main problem of interest in this paper is *state reachability* (aka *leaf coverability*): given as before an AVASS $\langle Q, d, T_u, T_f \rangle$ and two states q_r and q_ℓ in Q, we ask now whether there exists a deduction

tree with root labelled by $(q_r, \mathbf{0})$ and every leaf label in $\{q_\ell\} \times \mathbb{N}^d$. Equivalently, does Controller have a strategy that ensures that a play starting in $(q_r, \mathbf{0})$ eventually visits $(q_\ell, \mathbf{v})$ for some $\mathbf{v}$ in $\mathbb{N}^d$?

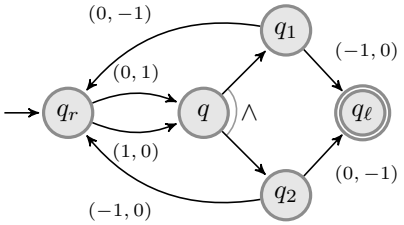
We prove in this paper that state reachability is 2-ExpTime-complete, see Theorem 3.1 and Proposition 4.1.

2.2.3. *Non-Termination.* A second problem of interest is *non-termination*: given an AVASS $\langle Q, d, T_u, T_f \rangle$ and an initial state q_r in Q, does there exist a deduction tree where every branch is infinite? Equivalently, does Controller have a strategy to ensure that a play starting in $(q_r, \mathbf{0})$ never stops?

Brázdil, Jančar, and Kučera [5] show in the context of energy games that this problem is ExpSpace-hard, and in $(d-1)$-ExpTime when the dimension d is fixed. It is still open whether this upper bound could be lowered, but our 2-ExpTime lower bound in Proposition 4.1 is the best one can hope for.

We discuss a few other decision problems related to energy games in Section 5 and to regular VASS simulations in Section 6.

2.3. **Example.** Figure 1a presents a 2-dimensional AVASS with state set $Q = \{q_r, q, q_1, q_2, q_\ell\}$ and a single fork rule $q \to q_1 \wedge q_2$.



(A) A 2-dimensional AVASS $\mathcal{A}$.

$$\dfrac{\dfrac{\dfrac{q_r, 1, 0}{q, 1, 1}}{\dfrac{q_1, 1, 1}{q_\ell, 0, 1}} \quad \dfrac{\dfrac{q_2, 1, 1}{q_\ell, 1, 0}}{}}{}$$

(B) A deduction tree in $\mathcal{A}$.

FIGURE 1. State reachability and non-termination in an AVASS.

For the state reachability problem, observe that Controller cannot ensure reaching state q_ℓ from the initial configuration $(q_r, 0, 0)$: in configuration $(q, 0, 1)$, Environment can send to $(q_1, 0, 1)$ from which only $(q_r, 0, 0)$ can be reached; in configuration $(q, 1, 0)$ Environment can send instead to $(q_2, 1, 0)$ with the same conclusion. However, starting instead from an initial configuration $(q_r, 1, 0)$ would give Controller a winning strategy described by the deduction tree in Figure 1b.

Regarding the non-termination problem, in a similar way Controller cannot ensure an infinite execution from $(q_r, 0, 0)$: this time Environment can force Controller to visit q_ℓ where the system deadlocks.

3. Complexity Upper Bounds

The state reachability problem asks about the existence of a deduction tree with root $(q_r, \mathbf{0})$ and leaves labels in $\{q_\ell\} \times \mathbb{N}^d$, which describes when using the game semantics a winning strategy for Controller. More generally, we are interested in deduction trees with root label $(q, \mathbf{v})$ and leaves in

$\{q_\ell\} \times \mathbb{N}^d$, which we call *witnesses* for $(q, \mathbf{v})$. Let us write $\mathcal{A}, q_\ell \triangleright q, \mathbf{v}$ if such a witness exists in an AVASS $\mathcal{A}$; then the state reachability problem asks whether $\mathcal{A}, q_\ell \triangleright q_r, \mathbf{0}$.

Following Rackoff [18], the main idea to prove a 2-ExpTime upper bound on the state reachability problem is to prove a doubly exponential upper bound on the height of witnesses, by induction on the dimension d; see Section 3.1. But let us first make a useful observation: if $\mathcal{A}, q_\ell \triangleright q, \mathbf{v}$ and $(q', \mathbf{v}') \geq (q, \mathbf{v})$ for the product ordering over $Q \times \mathbb{N}^d$, i.e. if $q = q'$ and $\mathbf{v}'(i) \geq \mathbf{v}(i)$ for all $1 \leq i \leq d$, then $\mathcal{A}, q_\ell \triangleright q', \mathbf{v}'$, and is moreover witnessed by a deduction tree of the same height—indeed, this is the deduction tree with $\mathbf{v}' - \mathbf{v}$ added to all the labels. This means that the set of root labels that ensure reaching q_ℓ is *upward-closed*, and since $(Q \times \mathbb{N}^d, \leq)$ is a well partial order, it has a finite set of minimal elements called its *Pareto frontier*:

$$\mathrm{Pareto}(\mathcal{A}, q_\ell) \stackrel{\text{def}}{=} \min\{(q, \mathbf{v}) \in Q \times \mathbb{N}^d \mid \mathcal{A}, q_\ell \triangleright q, \mathbf{v}\} \,. \tag{1}$$

For instance, with the example AVASS of Section 2.3,

$$\mathrm{Pareto}(\mathcal{A}, q_\ell) = \{(q_r, 1, 0), (q_r, 0, 1), (q, 1, 1), (q_1, 1, 0), (q_2, 0, 1), (q_\ell, 0, 0)\} \,.$$

We use in Section 3.2 the bounds on the size of witnesses to show that Pareto frontiers can be computed in doubly exponential time, which in turn proves:

Theorem 3.1. *State reachability in AVASS is in* 2-ExpTime. *It is in* ExpTime *when the dimension is fixed, and in* PTime *when furthermore the bitsize is fixed.*

Note that the PTime bound in the case of a fixed dimension and fixed bitsize is not trivial, since it still allows for infinite arenas. In essence it shows that we can add a fixed number of counters to a reachability game 'for free.'

3.1. **Small Witnesses.** Let us fix an instance $\langle \mathcal{A}, q_r, q_\ell \rangle$ of the state reachability problem with $\mathcal{A} = \langle Q, d, T_u, T_f \rangle$ and write $[d] \stackrel{\text{def}}{=} \{1, \ldots, d\}$ for its set of components. For a subset $I \subseteq [d]$ of the components of $\mathcal{A}$, we write $\mathbf{u}_{\restriction I}$ for the projection of a vector $\mathbf{u}$ on I, and define the *projection* $\mathcal{A}_{\restriction I} \stackrel{\text{def}}{=} \langle Q, |I|, T_{u\restriction I}, T_f \rangle$ of $\mathcal{A}$ on I as the AVASS with unary rules $T_{u\restriction I} \stackrel{\text{def}}{=} \{(q, \mathbf{u}_{\restriction I}, q') \mid (q, \mathbf{u}, q') \in T_u\}$. Let $W_I \stackrel{\text{def}}{=} \{(q, \mathbf{v}) \in Q \times \mathbb{N}^{|I|} \mid \mathcal{A}_{\restriction I}, q_\ell \triangleright q, \mathbf{v}\}$ be the set of witness roots in $\mathcal{A}_{\restriction I}$. We are interested in bounding the height $h(t)$ of *minimal* witnesses t in $\mathcal{A}_{\restriction I}$:

$$H_I \stackrel{\text{def}}{=} \sup_{(q,\mathbf{v}) \in W_I} \min\{h(t) \mid t \text{ witnesses } (q, \mathbf{v})\} \,, \tag{2}$$

where implicitly $H_I = 0$ if no witness exists.

A last remark before we proceed is that, if a label $(q, \mathbf{v})$ appears twice along a branch of a witness t, i.e. if $t = C[C'[t']]$ for some context C, some non-empty context C' with root label $(q, \mathbf{v})$, and tree t' with root label $(q, \mathbf{v})$, then the *shortening* $C[t']$ of t, obtained by replacing $C'[t']$ by t' in t, is also a witness.

Assume that there exists some witness for some root label $(q, \mathbf{v})$. We are going to bound H_I by induction on $|I|$: for the base case where $I = \emptyset$, by

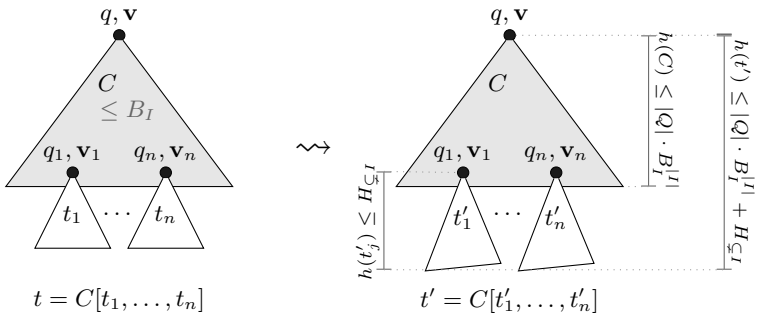


FIGURE 2. Bounding the height of minimal witnesses.

repeated shortenings we see that no branch of a minimal witness can have the same state twice, thus

$$H_\emptyset \leq |Q| \,. \tag{3}$$

Consider now some non-empty set I and a minimal witness t for $(q, \mathbf{v})$. We would like to bound H_I, assuming by induction hypothesis that we are able to bound H_J for all $J \subsetneq I$ by some value

$$H_{\subsetneq I} \geq \max_{J \subsetneq I} H_J \,.$$

Define for this a large value

$$B_I \stackrel{\text{def}}{=} \|T_u\|_\infty \cdot H_{\subsetneq I}$$

and consider along each branch of t the first occurrence (starting from the root) of a node with some vector value $\geq B_I$ if one exists. If there are n such occurrences then $t = C[t_1, \ldots, t_n]$ where C is a context where all the vector values are $< B_I$, and each t_j witnesses $\mathcal{A}_I, q_\ell \triangleright q_j, \mathbf{v}_j$ where $\mathbf{v}_j(i_j) \geq B_I$ for some i_j in I.

(1) By repeated shortenings, we can bound the height of C by $|Q| \cdot B_I^{|I|}$.
(2) For each j, let $I_j \stackrel{\text{def}}{=} I \setminus \{i_j\}$. Then t_j is also a witness for $\mathcal{A}_{\restriction I_j}, q_\ell \triangleright q_j, \mathbf{v}_{j\restriction I_j}$, and we can replace it by a witness t'_j of height at most H_{I_j}. Then t'_j also witnesses $\mathcal{A}_I, q_\ell \triangleright q_j, \mathbf{v}_j$ because B_I bounds the maximal total decrease that can occur along a branch of a deduction tree of height H_{I_j}.

See Figure 2 for an illustration. Hence $t' \stackrel{\text{def}}{=} C[t'_1, \ldots, t'_n]$ is a witness for $(q, \mathbf{v})$ and

$$H_I \leq h(t') \leq |Q| \cdot B_I^{|I|} + H_{\subsetneq I} = |Q| \cdot (\|T_u\|_\infty \cdot H_{\subsetneq I})^{|I|} + H_{\subsetneq I} \,. \tag{4}$$

Combining (3) with (4), we obtain:

$$H_{[d]} \leq (|Q| \cdot (\|T_u\|_\infty + 1) + 1)^{(3d)!} \,. \tag{5}$$

Observe that this bound is doubly exponential in d, but only exponential in the bitsize $\log(\|T_u\|_\infty + 1)$, and polynomial in the number of states $|Q|$.

3.2. **Pareto Frontier.** Equation (5) yields an algorithm in AExpSpace = 2-ExpTime to decide given $(q, \mathbf{v})$ in $Q \times \mathbb{N}^d$ whether $\mathcal{A}, q_\ell \triangleright q, \mathbf{v}$, since it suffices to look for a minimal witness of height at most $H_{[d]}$, and the vector values in such a witness are bounded by $H_{[d]} \cdot \|T_u\|_\infty$.

Furthermore, as observed by Yen and Chen [23], a bound like (5) that does not depend on the initial configuration $(q, \mathbf{v})$ can be exploited to compute the Pareto frontier: if $(q, \mathbf{v})$ belongs to $\mathrm{Pareto}(\mathcal{A}, q_\ell)$, then $\|\mathbf{v}\|_\infty \leq H_{[d]} \cdot \|T_u\|_\infty$. Thus the Pareto frontier can be computed by running the previous algorithm on at most $|Q| \cdot (1 + H_{[d]} \cdot \|T_u\|_\infty)^d$ candidates $(q, \mathbf{v})$:

Proposition 3.2. *Let $\mathcal{A} = \langle Q, d, T_u, T_f \rangle$ be an AVASS and q_ℓ be a state in Q. Then the Pareto frontier $\mathrm{Pareto}(\mathcal{A}, q_\ell)$ can be computed in doubly exponential time. If d is fixed it can be computed in exponential time, and if $\|T_u\|_\infty$ is also fixed it can be computed in polynomial time.*

4. Complexity Lower Bounds

In this section, we match the 2-ExpTime upper bound of Theorem 3.1 for state reachability in AVASS (Section 4.1). Regarding the fixed dimensional cases, we also show in Section 4.2 that our ExpTime upper bound is optimal—note that the case where both the dimension and the bitsize are fixed is trivially PTime-hard by reduction from the emptiness problem for tree automata. These lower bounds on decision problems also entail that our bounds in Proposition 3.2 for the complexity of computing Pareto frontiers are optimal.

4.1. **A General Lower Bound.** We extend the classical ExpSpace-hardness proof of Lipton [17] for state reachability in VASS to the AVASS case. Instead of reducing from the halting problem for Minsky machines with counter valuations bounded by 2^{2^n}, we reduce instead from the same problem for *alternating* Minsky machines.

More precisely, a Minsky machine can be defined as a VASS with additional *zero-test* rules T_z of the form $q \xrightarrow{i?=0} q'$ for $1 \leq i \leq d$ with deduction semantics

$$\frac{q, \mathbf{v} \quad \mathbf{v}(i) = 0}{q', \mathbf{v}} \text{ zero-test}$$

An *alternating* Minsky machine $\langle Q, d, T_u, T_f, T_z \rangle$ can similarly be defined by allowing fork rules. By adapting the usual encoding of Turing machines into Minsky machines [11] to the alternating case, the halting problem for alternating Minsky machines with counter values bounded by 2^{2^n} is hard for AExpSpace = 2-ExpTime. With this in mind, the necessary adaptations of Lipton's reduction are straightforward.

Proposition 4.1. *State reachability and non-termination in AVASS are hard for* 2-ExpTime.

Proof Idea. Consider an alternating Minsky machine $\mathcal{M} = \langle Q, d, T_u, T_f, T_z \rangle$ with vector components bounded by 2^{2^n} for $n \stackrel{\text{def}}{=} |\mathcal{M}|$ and a target state q_ℓ with no applicable rule. Note that we can assume that $\mathcal{M}$ always terminates: we can otherwise reduce to this case by constructing a padded $\mathcal{M}'$

that first initialises an additional counter to $|Q| \cdot 2^{m2^n}$ the number of distinct configurations of $\mathcal{M}$ and then decrements it at every step. Then state reachability and non-termination are essentially the same, as it suffices to add a self-loop on q_ℓ.

The main issue is to handle zero-tests using only unary rules (and forks). Lipton's idea to this end is to introduce *complementary* coordinates $\bar{i}$ for each coordinate i of the original machine, such that after an initialisation phase $\mathbf{v}(i)+\mathbf{v}(\bar{i}) = 2^{2^n}$ in any configuration $(q, \mathbf{v})$ of the constructed AVASS. Maintaining such an invariant is easy by encoding unary rules $q \xrightarrow{\mathbf{u}} q'$ by $q \xrightarrow{\mathbf{u}+\bar{\mathbf{u}}} q'$ where $\bar{\mathbf{u}}$ applies $-\mathbf{u}$ to the complement coordinates. Then a zero-test $q \xrightarrow{i?=0} q'$ can be replaced by a fork $q \to q' \wedge q_i$ where q_i is a new state such that a computation starting from $(q_i, \mathbf{v})$ eventually reaches q_ℓ only if $\mathbf{v}(\bar{i}) = 2^{2^n}$ and deadlocks otherwise. The subsystem reachable from q_i is constructed by Lipton using only unary rules, just like the initialisation subsystem, which guarantees that $(q_r, \mathbf{v})$ can be reached from the new initial configuration $(q'_r, \mathbf{0})$ only if $\mathbf{v}(i) = 0$ and $\mathbf{v}(\bar{i}) = 2^{2^n}$ for all $1 \leq i \leq d$ and deadlocks otherwise.

As can be noticed in this proof sketch, some aspects of Lipton's construction could be simplified by the use of forks, namely the somewhat delicate handling of 'return states' in the subsystems. However Theorem 3.1 shows that forks offer only limited additional computational power. □

Proposition 4.1 was implicit in the 2-ExpTime lower bound proofs of [9, 3] for similar questions. Reducing instead from AVASS would simplify these proofs by separating the extension of Lipton's arguments from the actual reduction.

4.2. **Fixed Dimension.** Similarly to Proposition 4.1, proving an ExpTime lower bound in the case where the dimension d is fixed is rather easy: Rosier and Yen [20, Theorem 3.1] show indeed that the *boundedness* problem for VASS of dimension $d \geq 4$ is PSpace-hard by reducing from the acceptance problem in linear bounded automata (LBA). Their proof easily extends to the state reachability and non-termination problems for VASS, and for AVASS by reducing instead from alternating LBA.

Proposition 4.2. *State reachability and non-termination in AVASS of fixed dimension $d \geq 4$ are* ExpTime*-hard.*

Proof Idea. Let us first fix some notation. An *alternating linear bounded automaton* is an alternating Turing machine $\mathcal{A} = \langle Q_\Diamond, Q_\Box, \Sigma, \Gamma, \delta, q_0, \vdash, \dashv, F\rangle$ where $Q_\Diamond$ and $Q_\Box$ are two disjoint finite sets of states with union $Q \stackrel{\text{def}}{=} Q_\Diamond \uplus Q_\Box$, $\Sigma \subseteq \Gamma$ are finite input and tape alphabets, $q_0 \in Q$ and $F \subseteq Q$ are an initial state and a set of final states, $\vdash$ and $\dashv$ are the left and right endmarkers in Σ, and δ is a transition relation in $Q \times \Gamma \times Q \times \Gamma \times \{-1, 1\}$.

A configuration of $\mathcal{A}$ is a triple $(\vdash a_1 \cdots a_n \dashv, q, i)$ where $a_1 \cdots a_n$ is a sequence in Γ^*, q is the current state, and $0 \leq i \leq n+1$ is the current position of the head. A transition (q, a_i, q', b, m) in δ updates such a configuration to $(\vdash a_1 \cdots a_{i-1} b a_{i+1} \cdots a_n \dashv, q', i + m)$. Because $\mathcal{A}$ is linearly bounded, it can never move left of $\vdash$ nor right of $\dashv$ nor overwrite them. A configuration with q in $Q_\Diamond$ is existential, and universal otherwise. We assume without

loss of generality that $\Gamma = \{0, 1\}$ and that there are exactly two transitions $(q_\Box, c_1, q_1, b_1, m_1)$ and $(q_\Box, c_2, q_2, b_2, m_2)$ for each state $q_\Box$ in $Q_\Box$.

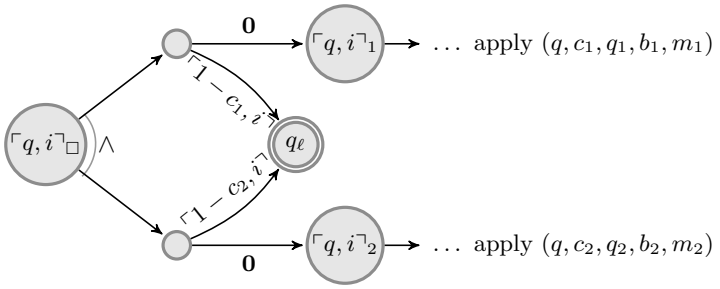
Alternation is handled as usual by seeing a computation as a finite tree with a single successor for existential configurations and all the successors for universal ones, and a computation is accepting if all its leaves are in accepting states. The acceptance problem for an alternating LBA $\mathcal{A}$ and an input x_0 in Σ^* asks whether there exists an accepting computation rooted by $(\vdash x_0 \dashv, q_0, 0)$, and is complete for APSpace = ExpTime.

Consider an instance $\langle \mathcal{A}, x_0 \rangle$ of the alternating LBA acceptance problem with $n \stackrel{\text{def}}{=} |x_0|$. The idea behind Rosier and Yen's reduction in the non-alternating case is to encode a $(\vdash a_1 \cdots a_n \dashv, q, i)$ of $\mathcal{A}$ using a 4-dimensional vector $\mathbf{v}$ such that $a_1 \cdots a_{i-1}$ is a binary representation of $\mathbf{v}(1)$, $a_i \cdots a_n$ one of $\mathbf{v}(2)$, and the complement sequences $(1 - a_1) \cdots (1 - a_{i-1})$ and $(1 - a_i) \cdots (1 - a_n)$ binary representations of $\mathbf{v}(3)$ and $\mathbf{v}(4)$. The current state q and head position i are encoded in the states of the AVASS. Testing and rewriting the current symbol and moving the head left or right can then be performed using unary rules, relying for this on the fact that update values exponential in n can be succinctly represented in an AVASS of polynomial size.

It thus remains to see how LBA alternation, i.e. a pair of universal transitions (q, c_1, q_1, b_1, m_1) and (q, c_2, q_2, b_2, m_2) with q in $Q_\Box$ can be implemented. Without entering the details of Rosier and Yen's encoding, we simply need the fact that there exist

- vectors $\ulcorner 0, i \urcorner$ and $\ulcorner 1, i \urcorner$ in $\mathbb{N}^4$ such that a rule $q' \xrightarrow{\ulcorner 0, i \urcorner} q''$ (resp. $q' \xrightarrow{\ulcorner 1, i \urcorner} q''$) can be applied in an AVASS configuration $(q', \mathbf{v})$ if and only if $\mathbf{v}$ encodes an LBA configuration with $a_i = 0$ (resp. $a_i = 1$), and
- states $\ulcorner q, i \urcorner_j$ for $j \in \{1, 2\}$ whose applicable rules implement transition (q, c_j, q_j, b_j, m_j) assuming head position i.

We construct the following gadget:



Observe that Environment cannot 'cheat' by attempting to force Controller into a deadlock by forcing the application of a transition (q, c_j, q_j, b_j, m_j) where c_j does not match the current symbol a_i under the head: Controller would punish such a move by going directly to q_ℓ the target state using the $\xrightarrow{\ulcorner 1 - c_j, i \urcorner}$ unary rule. □

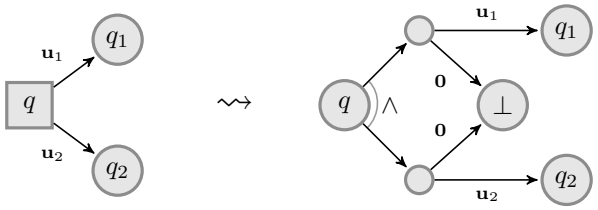


FIGURE 3. Translation of Environment rules from energy games to AVASS games [2].

5. Energy Games

The asymmetric game semantics described in § 2.1.2 is easily seen to be equivalent to one-sided VASS games as defined in [19, 2]. Such a game is played on a VASS with a partitioned state space $Q = Q_\Diamond \uplus Q_\Box$, where Controller owns the states in $Q_\Diamond$ and can freely manipulate the current vector value, while Environment owns the states in $Q_\Box$ and can only change the current state: if $q_\Box \xrightarrow{\mathbf{u}} q'$ is a rule in T_u and $q_\Box \in Q_\Box$, then $\mathbf{u} = \mathbf{0}$; these restricted Environment rules correspond to AVASS fork rules.

5.1. **Multi-dimensional Energy Games.** Abdulla et al. [2] have shown the equivalence of AVASS games with the *(multi-dimensional) energy games* of Brázdil et al. [5] and Chatterjee et al. [7], where the asymmetry between Controller and Environment is not enforced in the structure of the AVASS or in restricted unary rules for Environment: in such a game, Environment can use arbitrary unary rules. This would lead to an undecidable state reachability game when played on the $Q \times \mathbb{N}^d$ arena [1], but energy games are played instead over $Q \times \mathbb{Z}^d$—which means that unary rules can be applied even if they yield some negative vector components.

Asymmetry appears instead in the victory conditions for Controller. In addition to a winning condition $\mathit{Win} \subseteq Q^\omega \cup Q^*$ on the sequence of states $q_0, q_1, \ldots$ appearing during the play, Controller must also ensure that all the components of the vectors $\mathbf{v}_0, \mathbf{v}_1, \ldots$ remain non-negative. Such games are motivated by the synthesis of controllers that ensure that multiple quantitative values (represented by the integer vectors) are maintained above some critical values. Figure 3 illustrates Abdulla et al.'s translation from energy games to AVASS: state '$\bot$' denotes a *losing* state ensuring that no play that visits it can satisfy *Win*—typically a deadlock state.

Various regular winning conditions *Win* can be employed in this setting: the simplest one is (state) reachability, i.e. $\mathit{Win} = Q^*\{q_\ell\}$, which is in 2-ExpTime by Theorem 3.1. Non-termination, i.e. $\mathit{Win} = Q^\omega$, is shown to be in Tower, i.e. iterated exponential time, by Brázdil et al. [5]. Finally, parity is shown decidable by Abdulla et al. [2]. Theorem 4.1 furthermore entails that state reachability and non-termination (and thus parity) multi-dimensional energy games are 2-ExpTime-hard.

5.2. **Unknown Initial Credit.** Chatterjee et al. [7] focus on the case where the initial credit is unknown in these decision problems: given an AVASS $\langle Q, d, T_u, T_f \rangle$ a state q_r and a winning condition *Win*, does there exist $\mathbf{v}_r$

in $\mathbb{N}^d$ such that Controller has a winning strategy from the configuration $(q_r, \mathbf{v}_r)$? This can be applied to decide whether Controller has a winning finite-memory strategy in multi-dimensional mean-payoff games with unknown initial credit [7].

This question is trivial in the case of state reachability, because it does not depend on the update values $\mathbf{u}$ of unary rules:

Fact 5.1. *State reachability with unknown initial credit in AVASS is* PTime*-complete.*

Chatterjee, Randour, and Raskin [8] show that the unknown initial credit problem is co-NPTime-complete with parity winning conditions. In dimension $d = 2$ with fixed bitsize, Chaloupka [6] shows the problem to be PTime-complete for the non-termination objective.

6. Regular Simulations

Jančar and Moller [12] proved in 1995 that the two *regular VASS simulation problems* VASS $\preceq$ FS and FS $\preceq$ VASS, which ask whether a VASS is simulated by a finite-state system and vice versa, are decidable. They relied however on well quasi orders in their proofs and no complexity upper bounds have been published since—although Ackermann upper bounds are derivable from [10]. Regarding lower bounds, no improvement has appeared in the general case over the easy ExpSpace-hardness one can derive by reductions from the state reachability and non-termination problems for VASS and the lower bounds of Lipton [17] for these. However, in the particular case where we restrict ourselves to *basic parallel processes* (BPP) instead of VASS, Kučera and Mayr [14] proved that FS $\preceq$ BPP is PSpace-hard and BPP $\preceq$ FS is co-NPTime-hard, and both bounds were later improved to ExpSpace-hardness by Lasota [15].

By presenting reductions to and from the state reachability and non-termination problems in AVASS, we improve on all these results:

- BPP $\preceq$ FS and VASS $\preceq$ FS are both 2-ExpTime-complete by Proposition 4.1 and Theorem 3.1, and
- FS $\preceq$ BPP and FS $\preceq$ VASS are both 2-ExpTime-hard by Proposition 4.1 and in Tower by the results of Brázdil et al. [5].

We conclude the section by discussing some related problems: in Section 6.4, we show that the simulation equivalence problem FS $\simeq$ BPP is also 2-ExpTime-hard, and in Section 6.5, we show match the ExpSpace-hardness proof of Lasota [15] for the trace inclusion problem BPP $\subseteq$ FS by an ExpSpace upper bound for VASS $\subseteq$ FS.

6.1. Transition Systems and Simulations.

6.1.1. *Labelled Transition Systems.* Operational semantics are typically defined through *labelled transition systems* (LTS) $\mathcal{S} = \langle S, \Sigma, \to \rangle$ where S is a set of states, Σ is a set of actions, and $\to \subseteq S \times \Sigma \times S$ is a labelled transition relation, whose elements (s_1, a, s_2) are denoted by '$s_1 \xrightarrow{a} s_2$.' When S is finite we call $\mathcal{S}$ a *finite-state system* (FS).

For instance, the operational semantics of a VASS $\mathcal{V} = \langle Q, d, T_u \rangle$ along with a labelling $\sigma\colon T_u \to \Sigma$ using a set of actions Σ is the LTS $\mathcal{S}_\mathcal{V} \stackrel{\text{def}}{=} \langle Q \times$

$\mathbb{N}^d, \Sigma, \to\rangle$ with transitions $(q, \mathbf{v}) \xrightarrow{a} (q', \mathbf{v} + \mathbf{u})$ whenever $r = q \xrightarrow{u} q'$ is a unary rule in T_u with label $\sigma(r) = a$ (which we write more simply $q \xrightarrow{u,a} q'$ in the following).

6.1.2. *Simulations.* Given two LTS $\langle S_1, \Sigma, \to_1\rangle$ and $\langle S_2, \Sigma, \to_2\rangle$, a *simulation* is a relation $R \subseteq S_1 \times S_2$ such that, whenever (s_1, s_2) belongs to R then for each action a in Σ, if there exists s_1' in S_1 with $s_1 \xrightarrow{a}_1 s_1'$, then there also exists s_2' in S_2 such that $s_2 \xrightarrow{a}_2 s_2'$ and (s_1', s_2') is also in R. A state s_1 is *simulated* by a state s_2, written $s_1 \preceq s_2$, if there exists a simulation R such that (s_1, s_2) is in R.

Simulations can also be characterised by two-players turn-based *simulation games* between 'Spoiler', who wishes to disprove simulation, and 'Duplicator', who aims to establish its existence, played over the arena $S_1 \times S_2$. In a position (s_1, s_2), Spoiler first chooses a transition $s_1 \xrightarrow{a}_1 s_1'$ in $\mathcal{S}_1$, and Duplicator must answer with a transition $s_2 \xrightarrow{a}_2 s_2'$ with the same label a, and the game then proceeds from (s_1', s_2'). A player loses if during one of its turns no suitable transition can be found, otherwise the play is infinite and Duplicator wins. Then $s_1 \preceq s_2$ if and only if Duplicator has a winning strategy starting from (s_1, s_2).

Given two classes of (finitely-presented) systems $\mathbf{A}$ and $\mathbf{B}$, the *simulation problem* $\mathbf{A} \preceq \mathbf{B}$ takes as input two systems A in $\mathbf{A}$ and B in $\mathbf{B}$ with operational semantics $\mathcal{S}_A$ and $\mathcal{S}_B$, and two initial states s_A from $\mathcal{S}_A$ and s_B from $\mathcal{S}_B$, and asks whether $s_A \preceq s_B$. In the following we focus on *regular VASS simulations*, where one of $\mathbf{A}$ and $\mathbf{B}$ is the class of labelled VASS and the other the class FS.

6.2. **From Regular VASS Simulations to AVASS.** Our two reductions from regular VASS simulations essentially implement the simulation game as an AVASS game. Given a finite set of action Σ, a labelled VASS defined by $\mathcal{V} = \langle Q, d, T_u\rangle$ and $\sigma\colon T_u \to \Sigma$, a finite-state system $\mathcal{A} = \langle S, \Sigma, \to_{\mathcal{A}}\rangle$, and a pair of states (q_0, s_0) from $Q \times S$, we construct in both cases a state space

$$Q' \stackrel{\text{def}}{=} (Q \times S) \uplus (Q \times S \times \Sigma) \uplus \{q_\ell\}$$

for our AVASS. For convenience we allow forks of arbitrary finite arity $q \to q_1 \wedge \cdots \wedge q_r$.

6.2.1. **VASS $\preceq$ FS**. We actually reduce in this case from the complement problem VASS $\not\preceq$ FS to AVASS state reachability from (q_0, s_0). Controller plays the role of Spoiler, owns the states in $Q \times S$, and tries to reach the distinguished state q_ℓ. Environment plays the role of Duplicator and owns the states in $Q \times S \times \Sigma$. The rules of the AVASS are then:

$$(q, s) \xrightarrow{\mathbf{u}} (q', s, a) \qquad \text{whenever } q \xrightarrow{\mathbf{u},a} q' \in T_u \tag{6}$$

$$(q', s, a) \to q_\ell \wedge \bigwedge_{s \xrightarrow{a}_{\mathcal{A}} s'} (q', s') \,. \tag{7}$$

Observe that Spoiler has a winning strategy from (q_0, s_0) in the simulation game if and only if it can force Duplicator into a deadlock, i.e. a state s and an action a where no transition $s \xrightarrow{a}_{\mathcal{A}} s'$ exists. This occurs if and only

if Environment can be forced into going to q_ℓ in (7) in the AVASS game starting from (q_0, s_0).

Proposition 6.1. *There is a logarithmic space reduction from* VASS $\not\preceq$ FS *to AVASS state reachability.*

6.2.2. **FS $\preceq$ VASS**. This direction is actually a particular case of [2, Theorem 5], who show the decidability of *weak simulation* by reducing it to a parity energy game. Controller now plays the role of Duplicator, owns the states in $Q \times S \times \Sigma$, and attempts to force an infinite play. Environment plays the role of Spoiler and owns the states in $Q \times S$. The rules of the AVASS are then:

$$(q, s) \to \bigwedge_{s \xrightarrow{a}_{\mathcal{A}} s'} (q, s', a) \ , \tag{8}$$

$$(q, s', a) \xrightarrow{\mathbf{u}} (q', s') \qquad \text{whenever } q \xrightarrow{\mathbf{u}, a} q' \in T_u \ . \tag{9}$$

Then, Duplicator has a winning strategy in the simulation game from (q_0, s_0) if and only if Controller has a winning strategy for non-termination in the AVASS game starting in (q_0, s_0):

Proposition 6.2. *There is a logarithmic space reduction from* FS $\preceq$ VASS *to AVASS non-termination.*

6.3. **From AVASS to Regular VASS Simulations.**

6.3.1. *Basic Parallel Processes.* As announced at the beginning of the section, we prove our lower bounds on the more restricted BPP rather than VASS. Formally, a *BPP net* is a Petri net $\mathcal{N} = \langle P, T, W \rangle$ where P and T are finite sets of places and transitions and $W : (P \times T) \cup (T \times P) \to \mathbb{N}$ is the weighted flow, where additionally for all transitions t in T there is exactly one place p in P with $W(p, t) = 1$ and for all $p' \neq p$, $W(p', t) = 0$. Given a labelling function $\sigma : T \to \Sigma$, its semantics is defined by the LTS $\mathcal{S}_\mathcal{N} \stackrel{\text{def}}{=} \langle \mathbb{N}^{|P|}, \Sigma, \to_\mathcal{N} \rangle$ where $m \xrightarrow{a}_\mathcal{N} m - W(P, t) + W(t, P)$ if and only if there exists t with $\sigma(t) = a$ and $m \geq W(P, t)$. In figures we represent places as circles, transitions as rectangles, and positive flows as arrows.

In both our reductions, we want to implement an AVASS game as a simulation game where the FS is in charge of maintaining the state information and the BPP is in charge of maintaining the vector values. We assume we are given an AVASS $\langle Q, d, T_u, T_f \rangle$ in *ordinary form*, i.e. where the only updates vectors in T_u are unit vectors, and in *binary form*, i.e. for each state q of Q, either

- there is a fork $q \to q_1 \wedge q_2$ (and we call q an *universal state*), or
- there are exactly two unary rules $q \xrightarrow{\mathbf{u}_1} q_1$ and $q \xrightarrow{\mathbf{u}_2} q_2$ with origin q (and we call it an *existential state*), or
- there are no applicable rules at all (and we call it a *deadlock state*).

We can ensure these two conditions through logarithmic space reductions. Our action alphabet is then defined as

$$\Sigma \stackrel{\text{def}}{=} \{\forall, \exists, 1, 2\} \cup \{inc_i, dec_i \mid 1 \leq i \leq d\} \ .$$

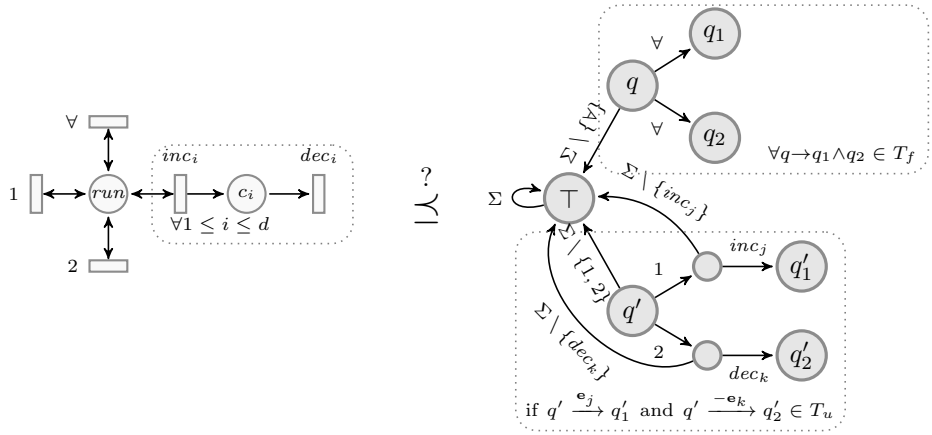


FIGURE 4. Reducing AVASS state reachability to a simulation BPP $\not\preceq$ FS.

6.3.2. **BPP $\preceq$ FS**. We actually reduce AVASS state reachability to BPP $\not\preceq$ FS and assume wlog. that the target state q_ℓ is a deadlock state. We construct a BPP net for Spoiler with places

$$P \stackrel{\text{def}}{=} \{run\} \cup \{c_i \mid 1 \leq i \leq d\}$$

where *run* contains a single token at all times and the c_i's encode the current vector value of the AVASS. Its transitions, labels and flows are depicted on the left of Figure 5. Its purpose is to force Duplicator, which is playing on the FS depicted on the right of Figure 5, into state q_ℓ. Because q_ℓ is a deadlock state and Spoiler can always fire transitions (e.g. $\forall$), it then wins the simulation game.

Duplicator plays the role of Environment in the original AVASS game and maintains the AVASS state using its state space, which contains Q. When in a universal state it can choose the following state, but when in an existential state Spoiler chooses instead the branch by firing transition 1 or 2. Duplicator ensures that the sequence of transitions of Spoiler is indeed valid in the original AVASS, by punishing invalid transitions by entering state '$\top$,' where it can play any symbol and thus win the simulation game.

Proposition 6.3. *There is a logarithmic space reduction from AVASS state reachability to* BPP $\not\preceq$ FS.

6.3.3. **FS $\preceq$ BPP**. In this direction we reduce from the non-termination problem. Spoiler now plays in an FS depicted on the left of Figure 5 and plays for Environment in the original AVASS game. It still maintains the current state of the AVASS in its state space.

Duplicator now plays on a BPP depicted on the right of Figure 5. It plays the role of Controller in the original VASS game and maintains the vector values in its places c_i as before. We rely on *Duplicator's choice*: using the '$\exists$' label in existential states, Spoiler leaves the choice to Duplicator, who can punish Spoiler if it does not comply with its choice between actions '1' and '2' by putting a token in place '$\top$' from where it wins.

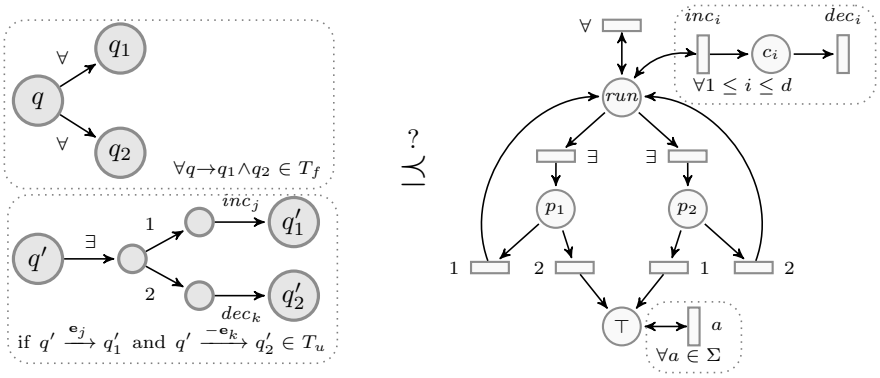


FIGURE 5. Reducing AVASS non-termination to a simulation FS $\preceq$ BPP.

Proposition 6.4. *There is a logarithmic space reduction from AVASS non-termination to* FS $\preceq$ BPP.

6.4. **Simulation Equivalence.** Two states s_1 and s_2 of two LTS $\mathcal{S}_1$ and $\mathcal{S}_2$ are *simulation equivalent*, noted $s_1 \simeq s_2$, if they simulate each other—though not necessarily by a single symmetric relation R—: $s_1 \preceq s_2$ and $s_2 \preceq s_1$. The reduction from AVASS state reachability to BPP $\not\preceq$ FS in Proposition 6.3 can easily be lifted to a reduction to BPP $\not\simeq$ FS. This proves that BPP $\simeq$ FS is 2-EXPTIME-hard, and improves on the EXPSPACE lower bound of Lasota [15].

Indeed, as noted by Lasota, $\mathcal{N} \preceq \mathcal{A}$ is equivalent to $\mathcal{N} + \mathcal{A} \simeq \mathcal{A}$, where '+' denotes a non-deterministic choice. Since a finite state system is also a BPP, $\mathcal{N} + \mathcal{A}$ can be represented by a BPP $\mathcal{N}'$ with copies of $\mathcal{N}$ and $\mathcal{A}$ and an initial place marked p_0 and two transitions with label '*init*' from p_0 to *run* and q_r. We match this new action on the FS side by adding a new initial state q'_r and a transition labelled by *init* to q_r to $\mathcal{A}$ to form $\mathcal{A}'$. Then Controller has a winning strategy to reach q_ℓ from q_r in the AVASS game if and only if Spoiler has a winning strategy in the simulation game between $\mathcal{N}'$ and $\mathcal{A}'$ starting from $(\{p_0\}, q'_r)$.

Proposition 6.5. *There is a logarithmic space reduction from AVASS state reachability to* BPP $\not\simeq$ FS.

6.5. **Trace Inclusion.** Given a LTS $\mathcal{S} = \langle S, \Sigma, \to \rangle$, a *trace* of $\mathcal{S}$ from a state s_0 is a finite sequence $a_1 a_2 \cdots a_n$ in Σ^* that labels some sequence of transitions $s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} \cdots \xrightarrow{a_n} s_n$ of $\mathcal{S}$. Given two LTS $\mathcal{S}_1$ and $\mathcal{S}_2$ and two initial states s_1 and s_2, we write '$s_1 \subseteq s_2$' if the set of traces originating in s_1 is included in that originating in s_2. If also $s_2 \subseteq s_1$ then s_1 and s_2 are *trace equivalent*, noted '$s_1 \equiv s_2$.' The corresponding *regular trace inclusion* problems VASS $\subseteq$ FS and FS $\subseteq$ VASS were shown decidable by Jančar and Moller [12].

If $\mathcal{S}_2 = \langle S_2, \Sigma, \to \rangle$ is *deterministic*, i.e. if for all s in S_2 and a in Σ, there is at most one s' in S_2 such that $s \xrightarrow{a} s'$, then $s_1 \preceq s_2$ if and only if $s_1 \subseteq s_2$. This property was exploited by Lasota [15] to show that the problems BPP $\subseteq$ FS, FS $\subseteq$ BPP, and BPP $\equiv$ FS were all EXPSPACE-hard:

indeed the 'right-hand' systems he constructed in this proofs were always deterministic. Inspecting our proofs in Section 6.3, we see that we cannot use this argument.

6.5.1. **VASS $\subseteq$ FS**. We show here an ExpSpace upper bound to VASS $\subseteq$ FS. Together with the ExpSpace lower bound of Lasota [15] for BPP $\subseteq$ FS, this yields the following:

Proposition 6.6. VASS $\subseteq$ FS *is* ExpSpace*-complete.*

Proof Idea. Consider an instance of VASS $\subseteq$ FS: let $\mathcal{V} = \langle Q, d, T_u \rangle$ be a VASS with labelling function $\sigma : T_u \to \Sigma$ and $\mathcal{A} = \langle S, \Sigma, \to_{\mathcal{A}} \rangle$ be a FS, and the trace inclusion question $(q_0, \mathbf{0}) \overset{?}{\subseteq} s_0$ for some initial q_0 in Q and s_0 in S.

We first apply the subset construction to $\mathcal{A}$ from s_0, yielding a deterministic finite-state system $\mathcal{D} \overset{\text{def}}{=} \langle 2^S, \Sigma, \to_{\mathcal{D}} \rangle$ of exponential size. By the previous discussion, $(q_0, \mathbf{0}) \subseteq s_0$ if and only if $(q_0, \mathbf{0}) \preceq \{s_0\}$.

We now build a synchronous product of $\mathcal{V}$ and $\mathcal{D}$: this is a VASS $\mathcal{V}' \overset{\text{def}}{=} \langle Q \times 2^S, d, T'_u \rangle$ where $(q, E) \xrightarrow{\mathbf{u}} (q', E')$ if and only if there exists a in Σ such that $q \xrightarrow{\mathbf{u},a} q'$ in $\mathcal{V}$ and $E \xrightarrow{a}_{\mathcal{D}} E'$ in $\mathcal{D}$. In this product $\mathcal{V}'$, the triples in

$$F \overset{\text{def}}{=} \{(q, E, \mathbf{v}) \in Q \times 2^S \times \mathbb{N}^d \mid \exists a \in \Sigma . \exists u \in \mathbb{Z}^d . \exists q' \in Q . \mathbf{v} + \mathbf{u} \geq 0 \\ \wedge q \xrightarrow{\mathbf{u},a} q' \wedge \nexists E' \subseteq S . E \xrightarrow{a} E'\} \tag{10}$$

denote the set of immediately winning positions $((q, \mathbf{v}), E)$ for Spoiler in the simulation game between $\mathcal{V}$ and $\mathcal{D}$: it can fire the corresponding transition $q \xrightarrow{\mathbf{u},a} q'$ in $\mathcal{V}$ but Duplicator cannot reply with a transition $E \xrightarrow{a} E'$ in $\mathcal{D}$. Note that F is upward-closed, with a finite set of minimal elements $\min F$ of size bounded by $|Q| \cdot 2^{|S|} \cdot \|T_u\|_\infty$.

Then Spoiler has a winning strategy from the initial pair $((q_0, \mathbf{0}), \{s_0\})$ if and only if there exists $(q, E, \mathbf{v})$ in $\min F$ such that there exists an execution of the VASS $\mathcal{V}'$

$$(q_0, \{s_0\}, \mathbf{0}) \xrightarrow{u_1} \cdots \xrightarrow{u_n} (q, E, \mathbf{v}') \tag{11}$$

for some n and $\mathbf{v}' \geq \mathbf{v}$. In other words, we can reduce $(q_0, \mathbf{0}) \subseteq s_0$ to a disjunction of *coverability* problems in $\mathcal{V}'$ a VASS with exponentially many states but the same dimension d and same bitsize $\log(\|T_u\| + 1)$.

By the complexity upper bounds of Rackoff [18] (see also Rosier and Yen [20] and Blockelet and Schmitz [4]), such a disjunction of coverability instances can be solved by a nondeterministic algorithm that guesses the element $(q, E, \mathbf{v})$ to cover, and guess and check a coverability witness of the form (11). This requires exponential space in the dimension but polynomial space in the bitsize and number of states of $\mathcal{V}'$, hence overall exponential space in the size of the original instance $\langle \Sigma, \mathcal{V}, \sigma, \mathcal{A}, q_0, s_0 \rangle$. □

6.5.2. **FS $\subseteq$ VASS**. The previous argument does not work for this direction. In fact, this is not surprising considering that Totzke [21] recently claimed the problem FS $\subseteq$ OCN to be Ackermann-complete, where 'OCN' denotes one-dimensional VASS.

7. Concluding Remarks

Alternating VASS provide a unified formalism to reason about VASS games, along with simple complexity arguments for state reachability objectives. This allows us to improve on all the previously known complexity bounds for regular VASS simulations, and show in particular that VASS $\preceq$ FS is 2-ExpTime-complete.

The main open question at this point is whether the upper bounds for non-termination and parity objectives on AVASS could be lowered to 2-ExpTime, and thus to close the gap between 2-ExpTime-hardness and Tower for FS $\preceq$ VASS. A first step to this end could be to extend the PTime upper bound of Chaloupka [6] for the fixed bitsize and unknown initial credit case from dimension two to arbitrary fixed dimensions. However, quoting Chaloupka, 'since the presented results about 2-dimensional VASS are relatively complicated, we suspect this problem is difficult.'

Acknowlegements. The authors thank Stefan Göller who drew our attention to [12, 15] and to the fact that the exact complexities of the two regular simulation problems were unknown.

References

[1] P. A. Abdulla, A. Bouajjani, and J. d'Orso. Deciding monotonic games. In *CSL 2003*, volume 2803 of *LNCS*, pages 1–14. Springer, 2003. doi:10.1007/978-3-540-45220-1_1.

[2] P. A. Abdulla, R. Mayr, A. Sangnier, and J. Sproston. Solving parity games on integer vectors. In *Concur 2013*, volume 8052 of *LNCS*, pages 106–120. Springer, 2013. doi:10.1007/978-3-642-40184-8_9.

[3] B. Bérard, S. Haddad, M. Sassolas, and N. Sznajder. Concurrent games on VASS with inhibition. In *Concur 2012*, volume 7454 of *LNCS*, pages 39–52. Springer, 2012. doi:10.1007/978-3-642-32940-1_5.

[4] M. Blockelet and S. Schmitz. Model-checking coverability graphs of vector addition systems. In *MFCS 2011*, volume 6907 of *LNCS*, pages 108–119. Springer, 2011. doi:10.1007/978-3-642-22993-0_13.

[5] T. Brázdil, P. Jančar, and A. Kučera. Reachability games on extended vector addition systems with states. In *ICALP 2010*, volume 6199 of *LNCS*, pages 478–489. Springer, 2010. doi:10.1007/978-3-642-14162-1_40.

[6] J. Chaloupka. Z-reachability problem for games on 2-dimensional vector addition systems with states is in P. *Fund. Inform.*, 123(1):15–42, 2013. doi:10.3233/FI-2013-798.

[7] K. Chatterjee, L. Doyen, T. A. Henzinger, and J.-F. Raskin. Generalized mean-payoff and energy games. In *FSTTCS 2010*, volume 8 of *LIPIcs*, pages 505–516. LZI, 2010. doi:10.4230/LIPIcs.FSTTCS.2010.505.

[8] K. Chatterjee, M. Randour, and J.-F. Raskin. Strategy synthesis for multi-dimensional quantitative objectives. In *Concur 2012*, volume 7454 of *LNCS*, pages 115–131. Springer, 2012. doi:10.1007/978-3-642-32940-1_10.

[9] S. Demri, M. Jurdziński, O. Lachish, and R. Lazić. The covering and boundedness problems for branching vector addition systems. *J. Comput. Syst. Sci.*, 79(1):23–38, 2012. doi:10.1016/j.jcss.2012.04.002.

[10] D. Figueira, S. Figueira, S. Schmitz, and Ph. Schnoebelen. Ackermannian and primitive-recursive bounds with Dickson's Lemma. In *LICS 2011*, pages 269–278. IEEE Computer Society, 2011. doi:10.1109/LICS.2011.39.

[11] P. C. Fischer, A. R. Meyer, and A. L. Rosenberg. Counter machines and counter languages. *Math. Syst. Theory*, 2(3):265–283, 1968. doi:10.1007/BF01694011.

[12] P. Jančar and F. Moller. Checking regular properties of Petri nets. In *Concur '95*, volume 962 of *LNCS*, pages 348–362. Springer, 1995. doi:10.1007/3-540-60218-6_26.

[13] M. I. Kanovich. Petri nets, Horn programs, linear logic and vector games. *Ann. Pure App. Logic*, 75(1–2):107–135, 1995. doi:10.1016/0168-0072(94)00060-G.
[14] A. Kučera and R. Mayr. Simulation preorder over simple process algebras. *Inform. and Comput.*, 173(2):184–198, 2002. doi:10.1006/inco.2001.3122.
[15] S. Lasota. ExpSpace lower bounds for the simulation preorder between a communication-free Petri net and a finite-state system. *Inf. Process. Lett.*, 109(15): 850–855, 2009. doi:10.1016/j.ipl.2009.04.003.
[16] P. Lincoln, J. Mitchell, A. Scedrov, and N. Shankar. Decision problems for propositional linear logic. *Ann. Pure App. Logic*, 56(1–3):239–311, 1992. doi:10.1016/0168-0072(92)90075-B.
[17] R. Lipton. The reachability problem requires exponential space. Technical Report 62, Yale University, 1976.
[18] C. Rackoff. The covering and boundedness problems for vector addition systems. *Theor. Comput. Sci.*, 6(2):223–231, 1978. doi:10.1016/0304-3975(78)90036-1.
[19] J.-F. Raskin, M. Samuelides, and L. V. Begin. Games for counting abstractions. In *AVoCS 2004*, volume 128(6) of *ENTCS*, pages 69–85. Elsevier, 2005. doi:10.1016/j.entcs.2005.04.005.
[20] L. Rosier and H.-C. Yen. A multiparameter analysis of the boundedness problem for vector addition systems. *J. Comput. Syst. Sci.*, 32(1):105–135, 1986. doi:10.1016/0022-0000(86)90006-1.
[21] P. Totzke. Personnal communication.
[22] A. Urquhart. The complexity of decision procedures in relevance logic II. *J. Symb. Log.*, 64(4):1774–1802, 1999. doi:10.2307/2586811.
[23] H.-C. Yen and C.-L. Chen. On minimal elements of upward-closed sets. *Theor. Comput. Sci.*, 410(24–25):2442–2452, 2009. doi:10.1016/j.tcs.2009.02.036.

ENS Cachan & INRIA, France
E-mail address: {jbcourtois,schmitz}@lsv.ens-cachan.fr