



## Concurrent parameterized games

Nathalie Bertrand, Patricia Bouyer, Anirban Majumdar

► **To cite this version:**

Nathalie Bertrand, Patricia Bouyer, Anirban Majumdar. Concurrent parameterized games. FSTTCS 2019 - 39th IARCS Annual Conference on. Foundations of Software Technology and Theoretical Computer Science, Dec 2019, Bombay, India. pp.1-15. hal-02351236

**HAL Id: hal-02351236**

**<https://hal.inria.fr/hal-02351236>**

Submitted on 6 Nov 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Concurrent parameterized games

Nathalie Bertrand 

Univ Rennes, Inria, CNRS, IRISA – Rennes (France)

Patricia Bouyer 

LSV, CNRS & ENS Paris-Saclay, Univ. Paris-Saclay – Cachan (France)

Anirban Majumdar

Univ. Rennes, Inria, CNRS, IRISA – Rennes (France)

LSV, CNRS & ENS Paris-Saclay, Univ. Paris-Saclay – Cachan (France)

---

## Abstract

Traditional concurrent games on graphs involve a fixed number of players, who take decisions simultaneously, determining the next state of the game. In this paper, we introduce a parameterized variant of concurrent games on graphs, where the parameter is precisely the number of players. Parameterized concurrent games are described by finite graphs, in which the transitions bear regular languages to describe the possible move combinations that lead from one vertex to another.

We consider the problem of determining whether the first player, say Eve, has a strategy to ensure a reachability objective against any strategy profile of her opponents as a coalition. In particular Eve’s strategy should be independent of the number of opponents she actually has. Technically, this paper focuses on an *a priori* simpler setting where the languages labeling transitions only constrain the number of opponents (but not their precise action choices). These constraints are described as semilinear sets, finite unions of intervals, or intervals.

We establish the precise complexities of the parameterized reachability game problem, ranging from PTIME-complete to PSPACE-complete, in a variety of situations depending on the constraints (semilinear predicates, unions of intervals, or intervals) and on the presence or not of non-determinism.

**2012 ACM Subject Classification** Theory of computation → Verification by model checking

**Keywords and phrases** concurrent games; parameterized verification

**Funding** Patricia Bouyer has been funded by ERC project EQualIS.

**Acknowledgements** We thank Christoph Haase for insightful discussions on semilinear sets.

## 1 Introduction

**Parameterized verification.** The generalisation and everyday usage of, for example, cloud computing and blockchains technology, calls for the verification of algorithms running on distributed systems. Concrete examples are consensus and leader-election algorithms, but also coherence protocols, etc. This explains the recent interest of the model-checking community for the verification of systems composed of an arbitrary number of agents [10, 5].

Verifying algorithms running on distributed systems for all possible number of agents at once calls for symbolic techniques. These are generic, and compare favorably –in terms of complexity– to applying standard verification techniques on a given instance with a fixed large number of agents. Therefore, beyond its original goal of verifying systems independently of the number of agents, parameterized verification can also be more efficient than standard verification for large systems. In the last 15 years, parameterized verification algorithms were successfully applied to various case studies, such as data-consistency for cache coherence protocols in uniform memory access multiprocessors [9], and the core of simple reliable broadcast protocols in asynchronous systems [13].

**Multiplayer concurrent games.** In parallel, for multi-agents systems, the AI and model-checking communities traditionally use concurrent games on graphs to model the complex



© N. Bertrand, P. Bouyer, A. Majumdar;

licensed under Creative Commons License CC-BY

39th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2019).

Editors: Arkadev Chattopadhyay and Paul Gastin; Article No. 31; pp. 31:1–31:15



Leibniz International Proceedings in Informatics

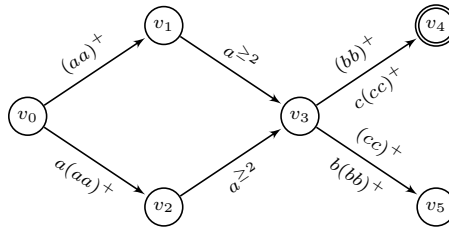
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

interactions between agents [1, 2]. An arena for  $n$  players is a directed graph where the transitions are labeled by  $n$ -tuples of actions. At each vertex of the graph, the  $n$  players select simultaneously and independently an action, and the next vertex is determined by the combined move consisting of all the actions. Most often, one considers infinite duration plays, that is plays generated by iterating this process forever. Concepts studied on multiplayer concurrent games include some borrowed from game theory, such as winning strategies (see *e.g.* [1]), rationality of players (see *e.g.* [11]), Nash equilibria (see *e.g.* [17, 6]).

**Concurrent games with a parameterized number of players.** The purpose of the current paper is to settle the foundations of concurrent games involving a parameterized number of players, paving the way to the modelling and verification of interactions involving an arbitrary number of agents. We envision that such games may later have applications in a variety of contexts, such as telecommunications and distributed algorithms. The conclusion presents a simple coordination game, and one of our long-term objectives is to solve the distributed synthesis problem of such games.

Generalising concurrent games to a parameterized number of agents can be done by replacing, on edges of the arena, tuples representing the choice of each of the agents by languages of finite yet *a priori* unbounded words. It seems natural to first consider regular languages, represented by regular expressions. For instance the label  $a^+$  represents that all players choose action  $a$ , while  $ab^+$  is the situation where the first player chooses  $a$ , while all other players play  $b$ . Such a parameterized arena can represent infinitely many interaction situations, one for each possible number of agents. In parameterized concurrent games, the agents do not know *a priori* the number of agents participating to the interaction. Each player observes the action it plays and the vertices the play goes through. These pieces of information may refine the knowledge each player has on the number of involved agents.

Figure 1 presents a first example of a parameterized arena. This arena represents a situation where the players need to figure out the parity of their number in order to make a correct decision (action  $b$  if there is an even number of players, and  $c$  otherwise). Here, players can collectively reach the target vertex  $v_4$ : they all play  $a$  in the two first steps, and from  $v_3$ , if the play went through  $v_1$  (resp.  $v_2$ ), they all play  $b$  (resp.  $c$ ).



■ **Figure 1** Example of a parameterized arena.

As for traditional concurrent games, one can consider natural questions such as, for instance, the distributed synthesis problem as in the above example, or the existence and computation of Nash equilibria. To start with, we consider a simpler decision problem: the first player, called Eve, is distinguished, and the question is whether she can ensure a reachability objective against the coalition of the other players, not knowing *a priori* the number of her opponents. She therefore must play uniformly, whatever the number of opponents she has. To simplify the exposition, we assume that the languages on transition of the arena are particularly simple: they only constrain the number of opponents Eve has. However, as discussed in Section 4, this simpler setting is not restrictive for the decision

problem we consider.

**Contributions.** After the definition of the parameterized game setting, the main contribution of this paper is the resolution of the so-called parameterized reachability game problem, with tight complexity bounds. We distinguish several cases, depending on whether arenas are deterministic or not, and on whether constraints on the number of opponents are intervals, finite unions of intervals, or semilinear sets.

The existence of a uniformly winning strategy for Eve reduces to the resolution of the *knowledge game*, a two-player reachability turn-based game. The latter is *a priori* exponential in the size of the original arena, since vertices include the knowledge Eve has on the possible number of her opponents, and this exponential blowup is unavoidable. Yet, when constraints are only intervals, the knowledge game is only of polynomial size. In this particular case, we prove the parameterized game problem to be PTIME-complete. For finite unions of intervals, and when the parameterized arena is deterministic, we show that if Eve has a winning strategy, she has one that can be represented by a polynomial size strategy tree. This small model property, together with the encoding of 3SAT allows us to prove the problem to be NP-complete. Finally, for finite unions of intervals and non-deterministic arenas, or for semilinear sets (with no assumption of non-determinism) the parameterized game problem is PSPACE-complete. The lower bound is obtained by a reduction from QBF-SAT, while the upper bound derives from a depth-first search algorithm on an exponential size tree, non-trivially extracted from the knowledge game. All the complexities are summarized in Table 1, on page 5.

**Related work.** Up to our knowledge, this contribution is the first to introduce and study a model of concurrent games with a parameterized number of players. Our model of parameterized concurrent games mixes interactions and an arbitrary number of agents. As far as we are aware, only a couple of other works in parameterized verification have defined a game semantics, and they all largely differ from the current setting. First, to study broadcast networks of many identical Markov decision processes, broadcast networks of two-player games were introduced [4]. There, the behaviour of each agent is the same and is described by a two-player turn-based game. Second, a control problem for an arbitrary size population of identical agents was studied in [3]. In that work, a controller plays against a parameterized number of agents, similarly to Eve playing against an unknown number of opponents. However, in contrast to our parameterized games, in the population control problem, the semantics is a turn-based game, and, most importantly, the arena is not centralized.

## 2 Game setting

We first introduce parameterized arenas, which form a simple setting for modelling games with a parameterized number of players. In such arenas, edges are labeled with sets of pairs  $(a, k)$  for  $a$  an action of Eve, and  $k$  a number of opponents. We discuss in Section 4 how a natural extension of concurrent games to a parameterized number of players, with regular languages on edges, reduces to this simpler setting. In the whole paper, we denote by  $\mathbb{N}$  the set of natural numbers (including 0) and write  $\mathbb{N}_{>0}$  for the set of positive natural numbers.

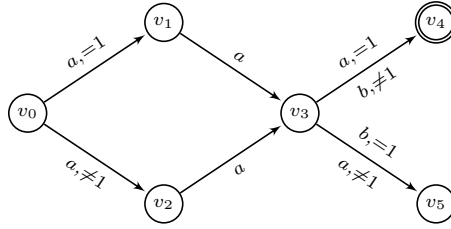
► **Definition 1.** A parameterized arena is a tuple  $\mathcal{A} = \langle V, \Sigma, \Delta \rangle$  where

- $V$  is a finite set of vertices;
- $\Sigma$  is a finite set of actions;
- $\Delta : V \times \Sigma \times \mathbb{N}_{>0} \rightarrow 2^V$  is the transition function.

## 31:4 Concurrent parameterized games

The arena is *deterministic* if for every  $v \in V$ , and every pair  $(a, k) \in \Sigma \times \mathbb{N}_{>0}$ , there is at most one vertex  $v' \in V$  such that  $v' \in \Delta(v, a, k)$ . Action  $a \in \Sigma$  is *enabled* at vertex  $v$  if there exists  $k \in \mathbb{N}_{>0}$  such that  $\Delta(v, a, k) \neq \emptyset$ . The arena is assumed to be *complete for enabled actions*: for every  $v \in V$ , if  $a$  is enabled at  $v$ , then for all  $k \in \mathbb{N}_{>0}$ ,  $\Delta(v, a, k) \neq \emptyset$ . This assumption is natural: Eve does not know how many opponents she has, and the successor vertex must exist whatever that number is. Given a predicate  $P \subseteq \mathbb{N}_{>0}$ ,  $\Delta(v, a, P)$  is a shorthand for  $\bigcup_{k \in P} \Delta(v, a, k)$ .

Further, for any  $v, v' \in V$  and  $a \in \Sigma$ , we introduce the following notation to represent the set of number of opponents that can lead from  $v$  to  $v'$  under action  $a$  of Eve:  $\nabla(v, a, v') = \{k \in \mathbb{N}_{>0} \mid v' \in \Delta(v, a, k)\}$ . Finally, we write  $E = \{(v, a, v') \mid \exists k \in \mathbb{N}_{>0}, v' \in \Delta(v, a, k)\}$  for the set of *edges* of the arena.



■ **Figure 2** Example of a parameterized reachability game.

► **Example 2.** An example of a deterministic parameterized reachability game is presented in Figure 2, with  $V = \{v_0, \dots, v_5\}$ ,  $\Sigma = \{a, b\}$ . Here and in other pictures, we use constraints to represent the transition function: for instance, the label ‘ $a, = 1$ ’ on the transition from  $v_0$  to  $v_1$  represents  $\Delta(v_0, a, 1) = \{v_1\}$ , and the label ‘ $a, \neq 1$ ’ means that for every  $k \neq 1$  (that is,  $k \geq 2$ ),  $\Delta(v_0, a, k) = \{v_2\}$ , or simply  $\Delta(v_0, a, \neq 1) = \{v_2\}$ . Moreover, we omit the constraint if it is trivial *e.g.*, for every  $k \in \mathbb{N}_{>0}$ ,  $\Delta(v_1, a, k) = \{v_3\}$ . On that example, action  $a$  is the only enabled action at vertices  $v_0, v_1$  and  $v_2$ , and both  $a$  and  $b$  are enabled at  $v_3$ . Also  $(v_0, a, v_1)$  is an example of edge. Finally,  $\nabla(v_3, a, v_4) = \{1\}$  and  $\nabla(v_3, b, v_4) = [2, \infty)$ .

Let  $k \in \mathbb{N}_{>0}$ . A  $k$ -*history*, for a coalition composed of  $k$  opponents of Eve, is a finite sequence  $v_0 a_0 \dots v_i \in (V \cdot \Sigma)^* \cdot V$  such that for every  $j < i$ ,  $v_{j+1} \in \Delta(v_j, a_j, k)$  (or equivalently  $k \in \bigcap_{j < i} \nabla(v_j, a_j, v_{j+1})$ ). A *history* in  $\mathcal{A}$  is a  $k$ -history for some  $k \in \mathbb{N}_{>0}$ . We note  $\text{Hist}(k)$  (resp.  $\text{Hist}$ ) for the set of  $k$ -histories (resp. histories) in  $\mathcal{G}$ . Similar notions of a  $k$ -*play* and a *play* are defined for infinite sequences.

► **Definition 3.** A strategy for Eve from  $v$  in  $\mathcal{A}$  is a mapping  $\sigma : \text{Hist} \rightarrow \Sigma$  that associates to every history  $hv' \in \text{Hist}$  an action  $\sigma(hv')$  which is enabled at  $v'$ . Further,  $\sigma$  is *memoryless* whenever for every  $hv', h'v' \in \text{Hist}$ ,  $\sigma(hv') = \sigma(h'v')$ .

A strategy for Eve is applied with no prior information on the number of her opponents. Given a strategy  $\sigma$ , an initial vertex  $v$  and  $k \in \mathbb{N}_{>0}$  a number of opponents, we define the outcome  $\text{Out}(\sigma, v, k)$  as the set of plays that  $\sigma$  induces from  $v$  when Eve has exactly  $k$  opponents. Formally,  $\text{Out}(\sigma, v, k)$  is the set of all  $k$ -plays  $\rho = v_0 a_0 v_1 a_1 v_2 \dots$  such that  $v = v_0$ , and for all  $i \geq 0$ ,  $\sigma(v_0 a_0 \dots v_i) = a_i$  and  $v_{i+1} \in \Delta(v_i, a_i, k)$ . The completeness assumption ensures that the set  $\text{Out}(\sigma, v, k)$  is not empty. Finally,  $\text{Out}(\sigma)$  is the set of all possible plays induced by  $\sigma$  from  $v$ :  $\text{Out}(\sigma, v) = \bigcup_{k \geq 1} \text{Out}(\sigma, v, k)$ .

Given an arena  $\mathcal{A} = \langle V, \Sigma, \Delta \rangle$ , a target vertex  $t \in V$  defines a *reachability game*  $\mathcal{G} = (\mathcal{A}, t)$  for Eve. A strategy  $\sigma$  for Eve from  $v$  in the reachability game  $\mathcal{G} = (\mathcal{A}, t)$  is *winning* if all

plays in  $\text{Out}(\sigma, v)$  eventually reach  $t$ . If there exists a winning strategy from  $v$ , then we say that  $v$  belongs to the *winning region* of Eve.

► **Example 4.** Resuming Example 2, one can show that Eve has a winning strategy  $\sigma$  from  $v_0$  to reach the target  $v_4$  defined by  $\sigma(v_0) = \sigma(v_0av_1) = \sigma(v_0av_2) = a$ ,  $\sigma(v_0av_1av_3) = a$  and  $\sigma(v_0av_2av_3) = b$ . Intuitively, the decision at vertex  $v_3$  depends on whether the play went through  $v_1$  –in this case Eve deduces that she has a single opponent– or  $v_2$ . Note that no memoryless strategy is winning for Eve: if she always chooses  $a$  at  $v_3$ , she is losing against more than 1 opponents; and similarly for  $b$ . The winning region for Eve is  $\{v_0, v_4\}$ .

The purpose of this paper is to establish the complexity of the following decision problem:

PARAMETERIZED REACHABILITY GAME PROBLEM

**Input:** A parameterized reachability game  $\mathcal{G} = (\mathcal{A}, t)$  and an initial vertex  $v$ .

**Question:** Does Eve have a winning strategy from  $v$  in  $\mathcal{G}$ ?

For algorithmic reasons, we assume the transition function  $\Delta$  of  $\mathcal{A}$  can be described in a finite way. More precisely, the sets  $\nabla(v, a, v')$  for  $v, v' \in V$  and  $a \in \Sigma$  should be simple enough.

We first consider constraints described by closed intervals (since we deal with sets of natural numbers, it is no restriction to assume intervals to be closed) or finite unions of closed intervals. If  $[a, b]$  (resp.  $[a, \infty)$ ) is an interval, then we say  $a$  is a left endpoint and  $b$  (resp.  $\infty$ ) is a right endpoint. As a complexity parameter, we use  $\#\text{endpoints}_{\mathcal{A}}$ , the number of endpoints used in constraints in  $\mathcal{A}$ . All the complexities will be functions of this parameter, independently of the precise values of the endpoints.

More generally, we also consider semilinear predicates over  $\mathbb{N}$ . A simple example of a semilinear predicate is the predicate “divisible by  $p$ ”, where  $p \in \mathbb{N}_{>0}$ . W.l.o.g. we assume semilinear sets are given as finite unions of ultimately periodic sets of integers. A set  $S \subseteq \mathbb{N}$  is *ultimately periodic* if there exist a threshold  $t \in \mathbb{N}$  and a period  $p \in \mathbb{N}$  such that for all  $a, b \in \mathbb{N}$  with  $a, b \geq t$  and  $a \equiv b \pmod{p}$ , we have  $a \in S$  iff  $b \in S$ . For complexity issues, all constants are assumed to be represented in binary. In that context, as a complexity parameter, we use  $\#\text{pred}_{\mathcal{A}}$ , the number of predicates used on edges of  $\mathcal{A}$ .

### 3 Resolution of the parameterized reachability game problem

In this section, we study the complexity of the parameterized reachability game problem.

► **Theorem 5.** *The complexity of the parameterized reachability game problem is stated in Table 1.*

|             |                            | Deterministic arenas | Non-deterministic arenas |
|-------------|----------------------------|----------------------|--------------------------|
| Constraints | Intervals                  | PTIME-complete       |                          |
|             | Finite unions of intervals | NP-complete          | PSPACE-complete          |
|             | Semilinear sets            | PSPACE-complete      |                          |

■ **Table 1** Complexity of the parameterized reachability game problem

Note that the complexities for constraints given as (finite unions of) intervals are independent of values of endpoints used in the constraints. When constraints are given as

semilinear sets, the complexity does depend on  $\#\text{pred}_A$  as well as the size of the encodings of the semilinear sets.

The rest of this section is devoted to proving these complexity results. To do so, we start with defining a finite two-player game abstraction, the *knowledge game*, which precisely captures the partial-information aspect of our parameterized game model.

### 3.1 The knowledge game

From a parameterized reachability game, we construct a standard two-player turn-based game. We do not recall this notion here, and refer to [12, Chap. 2] for it.

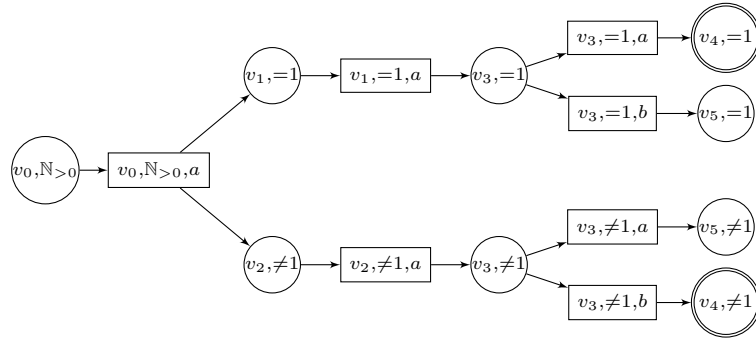
► **Definition 6.** Let  $\mathcal{G} = (\mathcal{A}, t)$  be a parameterized game, with  $\mathcal{A} = \langle V, \Sigma, \Delta \rangle$ . The knowledge game associated with  $\mathcal{G}$  is the two-player turn-based reachability game  $\mathcal{K}_{\mathcal{G}} = (V_E \cup V_A, \Delta_{\mathcal{K}}, F)$ , between Eve and Adam, such that  $V_E \subseteq V \times 2^{\mathbb{N}_{>0}}$  and  $V_A \subseteq V_E \times \Sigma$  are Eve and Adam vertices, respectively;  $\Delta_{\mathcal{K}} \subseteq (V_E \times V_A) \cup (V_A \times V_E)$  is the edge relation; and  $F = V_E \cap \{(t, K) \mid K \subseteq \mathbb{N}_{>0}\}$  is the set of target vertices. They are defined inductively by

- $\{(v, \mathbb{N}_{>0}) \mid v \in V\} \subseteq V_E$ ;
- $\forall (v, K) \in V_E, \forall a \in \Sigma$  enabled at  $v$ ,  $(v, K, a) \in V_A$  and  $((v, K), (v, K, a)) \in \Delta_{\mathcal{K}}$ ;
- $\forall (v, K, a) \in V_A, \forall v' \in V$  such that  $K \cap \nabla(v, a, v') \neq \emptyset$ ,  $(v', K \cap \nabla(v, a, v')) \in V_E$  and  $((v, K, a), (v', K \cap \nabla(v, a, v'))) \in \Delta_{\mathcal{K}}$ ;

A strategy for Eve in  $\mathcal{K}_{\mathcal{G}}$  is a function  $\lambda : (V_E \cdot V_A)^* \cdot V_E \rightarrow V_A$  compatible with  $\Delta_{\mathcal{K}}$ . We borrow standard notions of outcomes and winning strategies from the literature.

It is not hard to see that the game  $\mathcal{K}_{\mathcal{G}}$  is finite. Indeed, one can show by induction that every Eve's vertex  $(v, K)$  (hence every Adam's vertex  $(v, K, a)$ ) is such that  $K$  is an intersection of finitely many sets of the form  $\nabla(v', a, v'')$  or  $\mathbb{N}_{>0}$ .

► **Example 7.** Figure 3 represents the knowledge game associated with the parameterized game from Example 2. Circle vertices belong to Eve, and rectangle ones to Adam. In this two-player game, Eve has a winning strategy from  $(v_0, \mathbb{N}_{>0})$  to reach the doubly-circled target vertices.



■ **Figure 3** Knowledge game for the example of Figure 2.

We now investigate the size of  $\mathcal{K}_{\mathcal{G}}$ , that the number of its vertices and edges, w.r.t. the complexity measures we introduced for the parameterized game  $\mathcal{G}$ . Note that the size only might not reflect the complexity of *building* the knowledge game, in particular when constraints are given as semilinear predicates (one for instance needs to check emptiness of intersections of predicates); we discuss this further in the proof of Proposition 11.



► **Lemma 8.** For  $\mathcal{G} = (\mathcal{A}, t)$  a parameterized game with  $\mathcal{A} = \langle V, \Sigma, \Delta \rangle$ , the size of the associated knowledge game  $\mathcal{K}_{\mathcal{G}}$  is polynomial in both  $|V|$  and  $|\Sigma|$ , and

1. exponential in  $\#\text{pred}_{\mathcal{A}}$ , for constraints defined by semilinear predicates;
2. exponential in  $\#\text{endpoints}_{\mathcal{A}}$ , for constraints defined by finite unions of intervals; and
3. polynomial in  $\#\text{endpoints}_{\mathcal{A}}$ , for constraints defined as intervals.

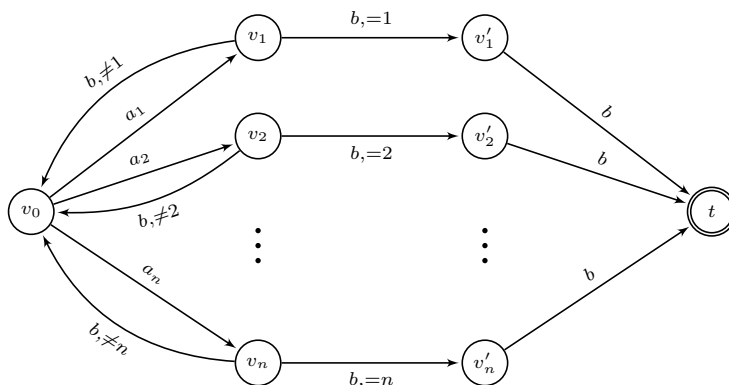
Furthermore, the exponential blowup is unavoidable in the two first cases.

**Proof.** By definition, all pairs  $(v, \mathbb{N}_{>0})$  for  $v \in V$  belong to  $V_E$  representing that Eve has no initial knowledge of the number of her opponents. Further knowledge sets for vertices in  $\mathcal{K}_{\mathcal{G}}$  are obtained by taking the intersection of existing knowledge sets with sets of the form  $\nabla(v, a, v')$ .

Therefore, when constraints in the arena are given by semilinear predicates, the number of knowledge sets is bounded by  $2^{\#\text{pred}_{\mathcal{A}}}$ . Hence  $|V_E| \leq 2^{\#\text{pred}_{\mathcal{A}}} |V|$  and  $|V_A| \leq 2^{\#\text{pred}_{\mathcal{A}}} |V| |\Sigma|$ , yielding an overall exponential bound on  $|\mathcal{K}_{\mathcal{G}}|$ . Note that it is exponential in the number of predicates, but not in the size of their encodings.

When constraints are defined by finite unions of intervals, the number of knowledge sets is bounded by  $3^{\#\text{endpoints}_{\mathcal{A}}}$ . Indeed, a finite union of intervals can be encoded by a word on the alphabet formed of the set of endpoints, with a repetition for singletons; for instance, if  $E = \{2, 5, 8, 11, 17, 23, \infty\}$ , writing  $a_i$  for the  $i$ -th letter of  $E$ ,  $[2, 8] \cup \{11\} \cup [17, \infty)$  is represented by the string  $a_1 a_3 a_4 a_4 a_5 a_7$ . Hence  $|V_E| \leq 3^{\#\text{endpoints}_{\mathcal{A}}} |V|$  and  $|V_A| \leq 3^{\#\text{endpoints}_{\mathcal{A}}} |V| |\Sigma|$ , yielding an overall exponential bound on  $|\mathcal{K}_{\mathcal{G}}|$ . Note that it is exponential in the number of endpoints, but not in the size of their encodings.

Finally, when constraints are defined by intervals, a better upper bound can be obtained. All knowledge sets in  $\mathcal{K}_{\mathcal{G}}$  are intervals whose endpoints appear in the constraints of  $\mathcal{A}$ . There can be at most  $\#\text{endpoints}_{\mathcal{A}}^2$  such intervals, so that  $|V_E| \leq \#\text{endpoints}_{\mathcal{A}}^2 |V|$  and  $|V_A| \leq \#\text{endpoints}_{\mathcal{A}}^2 |V| |\Sigma|$ , yielding an overall polynomial bound on  $|\mathcal{K}_{\mathcal{G}}|$ .



■ **Figure 4** A deterministic game  $\mathcal{G}_n$  ( $n \in \mathbb{N}_{>0}$ ), whose size is polynomial in  $n$  and whose knowledge game is exponential in  $n$ .

The exponential upper bound is reached by the family  $(\mathcal{G}_n)_{n \in \mathbb{N}_{>0}}$  of *deterministic* parameterized games depicted on Figure 4, and for which the constraints are unions of intervals (a particular case of semilinear predicates). Both the number of endpoints, and the number of predicates are linear in  $n$ . The associated knowledge game has vertices  $(v_0, K)$  for every non-empty subset  $K$  of  $\{1, \dots, n\}$ . Indeed, intuitively, from vertex  $(v_0, K)$  in  $\mathcal{K}_{\mathcal{G}_n}$ , for any  $k \in K$ , the successor vertex in two steps by  $a_k$  and  $b$ , in case the number of opponents is not  $k$ , is the vertex  $(v_0, K \setminus \{k\})$ . Thus  $|\mathcal{G}_n| \in O(n)$  and  $|\mathcal{K}_{\mathcal{G}_n}| \in O(n2^n)$ . ◀



We now state the correctness of the knowledge game construction:

► **Theorem 9.** *Eve has a winning strategy  $\sigma$  from  $v_0$  in  $\mathcal{G}$  if and only if she has a winning strategy  $\lambda$  from  $(v_0, \mathbb{N}_{>0})$  in  $\mathcal{K}_{\mathcal{G}}$ .*

**Proof sketch.** There is a correspondence between histories in  $\mathcal{G}$  and  $\mathcal{K}_{\mathcal{G}}$ . Every history  $h = v_0 a_0 v_1 \cdots v_i$  in  $\mathcal{G}$ , can be lifted to the history  $\kappa(h) = (v_0, K_0)(v_0, K_0, a_0)(v_1, K_1) \cdots (v_i, K_i)$  in  $\mathcal{K}_{\mathcal{G}}$  where:  $K_0 = \mathbb{N}_{>0}$ , and for every  $1 \leq j \leq i$ ,  $K_j = K_{j-1} \cap \nabla(v_{j-1}, a_{j-1}, v_j)$ . Note that  $\kappa(h)$  is well-defined since, by definition of a history,  $K_i$  is not empty. Conversely, any history  $H = (v_0, K_0)(v_0, K_0, a_0)(v_1, K_1) \cdots (v_i, K_i)$  in  $\mathcal{K}_{\mathcal{G}}$  projects to  $\iota(H) = v_0 a_0 v_1 \cdots v_i$  which is a history in  $\mathcal{G}$ . Moreover, for every  $k \in K_i$ ,  $\iota(H)$  is a  $k$ -history in  $\mathcal{G}$ . Using  $\kappa$  and  $\iota$ , one can easily lift winning strategies from  $\mathcal{G}$  to  $\mathcal{K}_{\mathcal{G}}$  and, vice versa project winning strategies from  $\mathcal{K}_{\mathcal{G}}$  to  $\mathcal{G}$ , to prove the desired equivalence. ◀

### 3.2 The simple case of intervals

► **Proposition 10.** *When constraints are intervals, the parameterized reachability game problem is PTIME-complete.*

When constraints are intervals only, the knowledge game is polynomial in the size of the parameterized arena (see Lemma 8) and it can be computed in polynomial time. Hence the parameterized reachability game problem is in PTIME. It is moreover complete for this class, since two-player reachability games are PTIME-hard (by straightforward reduction from the CIRCUIT-SAT problem). We thus obtain the above complexity result, independently of whether the arena is deterministic or not.

### 3.3 General PSPACE upper bound

► **Proposition 11.** *The parameterized reachability game problem is in PSPACE when constraints are given as finite unions of intervals or semilinear sets.*

**Proof sketch.** To prove this result, we rely on the knowledge game construction, which has been proven correct for the existence of winning strategies (see Theorem 9). Let  $\mathcal{G} = (\mathcal{A}, t)$  be a parameterized reachability game, and  $v_0$  be an initial vertex. We show that one can decide in polynomial space in the size of  $\mathcal{G}$  whether Eve has a winning strategy from  $(v_0, \mathbb{N}_{>0})$  in  $\mathcal{K}_{\mathcal{G}}$ .

For each vertex  $(v, K) \in V_E$  of Eve in  $\mathcal{K}_{\mathcal{G}}$ , we define a reachability game  $\mathcal{K}_{\mathcal{G}}[v, K]$ , which is the restriction of  $\mathcal{K}_{\mathcal{G}}$  to vertices  $(v', K, a)$  and  $(v', K')$  that are reachable from  $(v, K)$  via vertices with same knowledge set  $K$  only. Formally,  $\mathcal{K}_{\mathcal{G}}[v, K]$  is the restriction of  $\mathcal{K}_{\mathcal{G}}$  to the following sets of vertices, defined inductively:

$$\begin{cases} V_E^0 = \{(v, K)\} \\ V_A^i = \{(v', K, a) \mid v' \neq t \text{ and } (v', K) \in V_E^i \text{ and } ((v', K), (v', K, a)) \in \Delta_{\mathcal{G}}\} \\ V_E^{i+1} = \{(v', K') \mid \exists (v'', K, a) \in V_A^i \text{ s.t. } ((v'', K, a), (v', K')) \in \Delta_{\mathcal{G}}\} \end{cases}$$

Notice that in  $\mathcal{K}_{\mathcal{G}}[v, K]$ , all Adam vertices have knowledge set  $K$ . Also Eve vertices  $(v', K')$  with knowledge  $K' \subsetneq K$  or with  $v' = t$  have no successors: we refer to them as the *output vertices of  $\mathcal{K}_{\mathcal{G}}[v, K]$* . We write  $O^{[v, K]}$  for the set of such vertices.

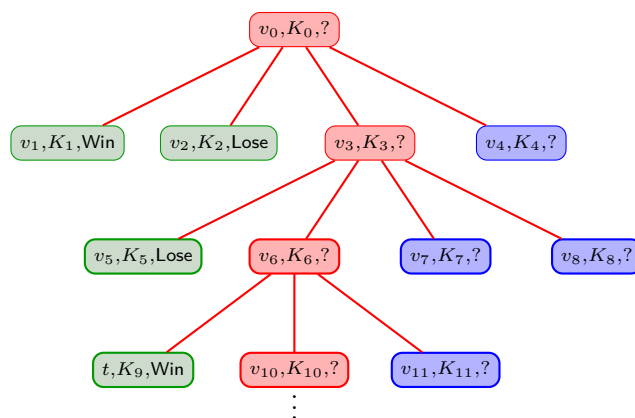
The game  $\mathcal{K}_{\mathcal{G}}[v, K]$  is polynomial in the size of  $\mathcal{G}$ . Indeed, there are at most  $(|\Sigma| + 1)|V|$  many Eve or Adam vertices with second component exactly  $K$  and at most  $|E||V|$  many Eve vertices with second component strictly smaller than  $K$ . When constraints are given as finite

unions of intervals, this game can be computed in polynomial time in  $\#\text{endpoints}_{\mathcal{A}}$ . For semilinear sets,  $\mathcal{K}_{\mathcal{G}}[v, K]$  can be computed in polynomial space in the size of the encodings of the predicates as finite unions of ultimately periodic sets; in particular, if  $P$  is a semilinear predicate one needs to check whether  $(P \cap K) \subsetneq K$  (to decide whether one obtains an output vertex of  $\mathcal{K}_{\mathcal{G}}[v, K]$ ). Once constructed,  $\mathcal{K}_{\mathcal{G}}[v, K]$  can be solved in polynomial time in  $|\mathcal{G}|$  since this is a standard two-player turn-based reachability game. We use these games in sub-routines for solving the parameterized reachability game problem.

Using the subgames  $\mathcal{K}_{\mathcal{G}}[v, K]$ , we consider the following exponential-size tagged tree  $\mathcal{T}$  defined inductively as follows: the root  $n_0 = (v_0, \mathbb{N}_{>0})$  is the initial vertex of  $\mathcal{K}_{\mathcal{G}}$ , and  $(v', K')$  is a child of  $(v, K)$  if  $(v', K') \in O^{[v, K]}$  is an output vertex of  $\mathcal{K}_{\mathcal{G}}[v, K]$ . Our aim is to tag each node  $n = (v, K)$  of  $\mathcal{T}$  with Win or Lose, to reflect whether Eve has a winning strategy from  $(v, K)$  in  $\mathcal{K}_{\mathcal{G}}$ . We define the following tagging function:

$$\text{tag}((v, K)) = \begin{cases} \text{Win} & \text{if } v = t \\ \text{Win} & \text{if Eve has a winning strategy in } \mathcal{K}_{\mathcal{G}}[v, K] \text{ from } (v, K) \text{ to reach} \\ & \text{the set } \{\alpha \in O^{[v, K]} \mid \text{tag}(\alpha) = \text{Win}\} \\ \text{Lose} & \text{otherwise.} \end{cases}$$

One can show the correctness of the tagging function:  $\text{tag}((v, K)) = \text{Win}$  if and only if Eve has a winning strategy in  $\mathcal{K}_{\mathcal{G}}$  from  $(v, K)$ . Finally, the root of the tree can be tagged in polynomial space, by a depth-first search algorithm on  $\mathcal{T}$  (see Figure 5). The height of  $\mathcal{T}$  is polynomially bounded, in  $\#\text{endpoints}_{\mathcal{A}}$  in the case of finite unions of intervals, and in  $\#\text{pred}_{\mathcal{A}}$  in the case of semilinear predicates. Once the tag of a node has been computed, its whole subtree can be forgotten. Therefore one can “reuse” polynomial space to repeatedly solve the games  $\mathcal{K}_{\mathcal{G}}[v, K]$  for different  $v$  and  $K$ . In the DFS tagging, the size of the stack is at most the height of tree times the maximal number of successors of a vertex  $v$  in  $\mathcal{G}$ . Finally polynomial space is sufficient to store the knowledge of one node of the  $\mathcal{T}$ . ◀



■ **Figure 5** Illustration of the polynomial space DFS tagging algorithm: the Win/Lose tags of green nodes have already been computed (and their subtrees have been removed); the tags of red nodes are being computed (hence the label “?”); and the blue nodes are waiting to be processed (we also use label “?”). For instance, before tagging  $(v_6, K_6)$ , one needs to first compute the tag of  $(v_{10}, K_{10})$  (which is ongoing), then compute the tag of  $(v_{11}, K_{11})$  (which is waiting).

### 3.4 An NP upper bound for deterministic arenas when constraints are finite unions of intervals

The previous PSPACE upper bound can be improved when the arena is deterministic and constraints are given by finite unions of intervals.

► **Proposition 12.** *The parameterized reachability game problem is in NP, when constraints are finite unions of intervals and when restricting to deterministic arenas.*

**Proof sketch.** Pick an arbitrary winning strategy  $\sigma$  for Eve, and consider the (labeled) tree  $\mathcal{T}_\sigma$  it induces: nodes are histories, and the children of a node are the possible next histories (depending on the number of opponents). This tree is finite because  $\sigma$  is winning, and one can add to the node label the knowledge Eve has for the corresponding history. This tree satisfies the following properties: (i) along any path of  $\mathcal{T}_\sigma$ , the number of distinct knowledge sets is at most  $\#\text{endpoints}_{\mathcal{A}}$ ; and (ii) the knowledge at sibling nodes form a partition of the knowledge at their parent node. The second property has the following consequence. At each level of the tree, the knowledge of all nodes form a partition of  $\mathbb{N}_{>0}$  using endpoints from the arena description, so that the number of nodes at each level is bounded by  $\#\text{endpoints}_{\mathcal{A}}$ . Also, if a node has the same knowledge as its parent, it cannot have siblings. This allows to *compress* linear parts of the tree, and to transform an arbitrary winning strategy into one whose tree is "small", *i.e.* polynomial in the size of the arena. ◀

### 3.5 Lower bounds

We prove all lower bounds mentioned in Table 1. We start with the PSPACE-hardness when constraints are finite unions of intervals and arenas are a priori non-deterministic.

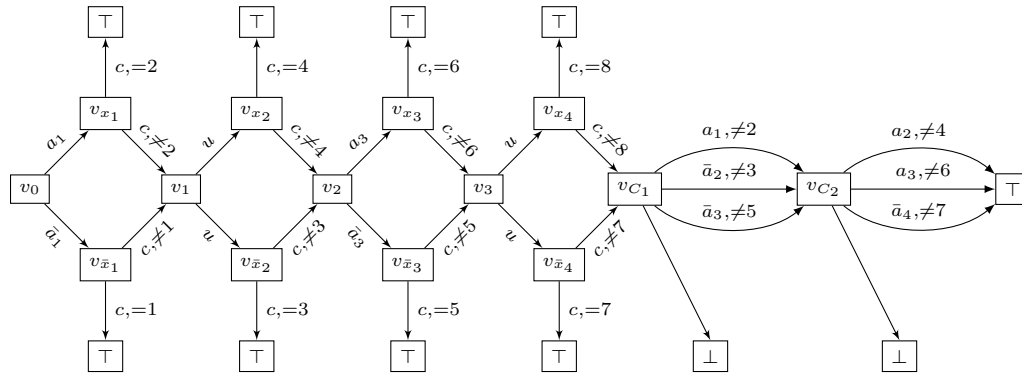
► **Proposition 13.** *When constraints are finite unions of intervals, the parameterized reachability game problem is PSPACE-hard.*

**Proof sketch.** The proof is by reduction from QBF-SAT, which is known to be PSPACE-complete [16]. Let  $\varphi = \exists x_1 \forall x_2 \exists x_3 \dots \forall x_{2r} \cdot (C_1 \wedge C_2 \wedge \dots \wedge C_m)$  be a quantified Boolean formula in prenex normal form, where for every  $1 \leq h \leq m$ ,  $C_h = \ell_{h,1} \vee \ell_{h,2} \vee \ell_{h,3}$ , and for every  $1 \leq j \leq 3$ ,  $\ell_{h,j} \in \{x_i, \neg x_i \mid 1 \leq i \leq 2r\}$  are the literals. From  $\varphi$ , we construct an arena  $\mathcal{A}_\varphi = \langle V, \Sigma, \Delta \rangle$  (see an illustrative example in Figure 6) as follows:

- $V = \{v_0, v_1, \dots, v_{2r-1}, v_{2r}\} \cup \{v_{x_1}, v_{\bar{x}_1}, \dots, v_{x_{2r}}, v_{\bar{x}_{2r}}\} \cup \{v_{C_1}, v_{C_2}, \dots, v_{C_m}, v_{C_{m+1}}\} \cup \{\perp, \top\}$ , where we identify  $v_{2r}$  with  $v_{C_1}$ , and  $v_{C_{m+1}}$  with  $\top$ .
- $\Sigma = \{u, c\} \cup \bigcup_{1 \leq i \leq 2r} \{a_i, \bar{a}_i\}$
- For every  $0 \leq s \leq r-1$ ,  $1 \leq i \leq 2r$ ,  $1 \leq h \leq m$  and  $1 \leq j \leq 3$ :
  1.  $\Delta(v_{2s}, a_{2s+1}, \geq 1) = \{v_{x_{2s+1}}\}$  and  $\Delta(v_{2s}, \bar{a}_{2s+1}, \geq 1) = \{v_{\bar{x}_{2s+1}}\}$
  2.  $\Delta(v_{2s+1}, u, \geq 1) = \{v_{x_{2s+2}}, v_{\bar{x}_{2s+2}}\}$
  3.  $\Delta(v_{x_i}, c, \neq 2i) = \{v_i\}$  and  $\Delta(v_{x_i}, c, = 2i) = \{\top\}$
  4.  $\Delta(v_{\bar{x}_i}, c, \neq 2i-1) = \{v_i\}$  and  $\Delta(v_{\bar{x}_i}, c, = 2i-1) = \{\top\}$
  5.  $\Delta(v_{C_h}, a_i, \neq 2i) = \{v_{C_{h+1}}\}$  if  $\ell_{h,j} = x_i$ ;  $\Delta(v_{C_h}, \bar{a}_i, \neq 2i-1) = \{v_{C_{h+1}}\}$  if  $\ell_{h,j} = \neg x_i$

To obtain a complete arena, all unspecified transitions lead to a sink state  $\perp$ .

From  $v_0$ , a first phase consists in choosing a valuation for the variables: Eve can choose the truth values of existentially quantified variables in vertices  $v_{2s}$  (with actions  $a_{2s+1}$  for true and  $\bar{a}_{2s+1}$  for false), and her opponents resolve the non-determinism of action  $u$  ( $u$  stands for universal) to choose the truth values of universally quantified variables in vertices  $v_{2s-1}$ . Due to the constraints on the edges, the knowledge of Eve at  $v_{C_1}$  contains for every



■ **Figure 6** Reduction for formula  $\varphi = \exists x_1 \forall x_2 \exists x_3 \forall x_4 \cdot (x_1 \vee \neg x_2 \vee \neg x_3) \wedge (x_2 \vee x_3 \vee \neg x_4)$ . Knowledge of Eve at  $v_{C_1}$  contains for every variable  $x_i$ , either  $2i$  or  $2i-1$  (and not both); containing  $2i$  (resp.  $2i-1$ ) encodes that  $x_i$  has been set to false (resp. true).

variable  $x_i$ , either  $2i$  or  $2i-1$  (and not both); where containing  $2i$  (resp.  $2i-1$ ) encodes the fact that  $x_i$  has been set to false by Eve or her opponents (resp. true).

From  $v_{C_1}$  a second phase starts where one checks whether the generated valuation makes all clauses in  $\varphi$  true. Sequentially, Eve chooses for every clause a literal that makes the clause true and these choices must be consistent with the first phase. To enforce this, plays with  $2i-1$  and  $2i$  opponents check the consistency of the assignment for variable  $x_i$ . For instance, if action  $a_i$  (encoding  $x_i$  set to true) against  $2i-1$  opponents leads from  $v_{C_h}$  to  $v_{C_{h+1}}$ , this means that  $v_{x_i}$  was visited, hence that  $x_i$  was set to true. On the contrary, if  $v_{x_i}$  was not visited, hence  $x_i$  was set to false, then against  $2i-1$  opponents, action  $a_i$  will lead to  $\perp$ . The role of  $\bar{a}_i$  is dual; it encodes assigning false to  $x_i$ , and will be checked with plays against  $2i$  opponents.

The above reduction ensures the following equivalence: Eve has a winning strategy in the parameterized game  $\mathcal{G}_\varphi = (\mathcal{A}_\varphi, \top)$  if and only if  $\varphi$  is true. ◀

Note that the reduction can also be done with only three actions, which is the maximal number of enabled actions from any vertex. The reduction uses unions of intervals (due to  $\neq i$  constraints). Finally the arena is non-deterministic at each vertex corresponding to universal quantifiers in  $\varphi$ . We extend this reduction in two ways to get rid of nondeterminism. First, instead of QBF-SAT, one can encode 3SAT (which is known to be NP-complete [8]) and obtain a deterministic parameterized game:

► **Corollary 14.** *When constraints are finite unions of intervals, and arenas are deterministic, the parameterized reachability game problem is NP-hard.*

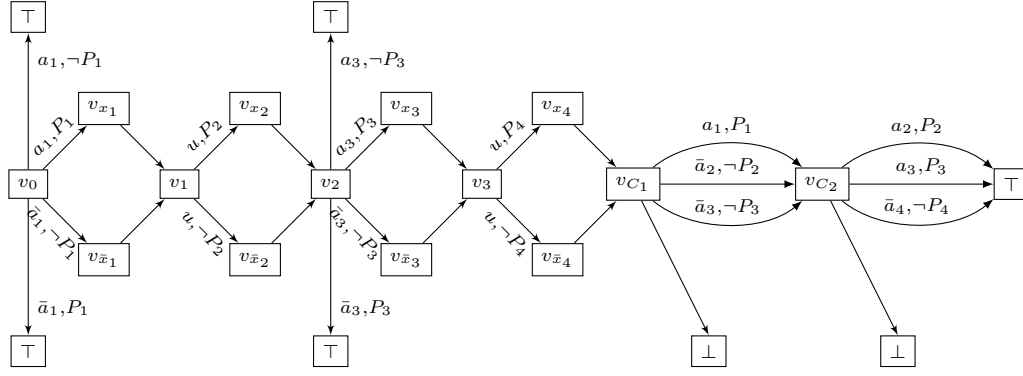
Second, increasing the expressive power of predicates can encode universal quantifiers without nondeterminism:

► **Proposition 15.** *When constraints are semilinear sets and arenas are deterministic, the parameterized reachability game problem is PSPACE-hard.*

**Proof sketch.** We slightly modify the construction of the proof of Proposition 13 as shown on Figure 7. For every  $1 \leq i \leq 2r$ ,  $p_i$  is the  $i$ -th prime number, and  $P_i$  the semilinear predicate “is a multiple of  $p_i$ ”.

Intuitively, at the end of the first phase, the truth value of variable  $x_i$  is witnessed by the fact that the set of possible number of opponents is a multiple of  $p_i$  if  $x_i$  is set to true (that

## 31:12 Concurrent parameterized games



■ **Figure 7** Reduction for formula  $\varphi = \exists x_1 \forall x_2 \exists x_3 \forall x_4 \cdot (x_1 \vee \neg x_2 \vee \neg x_3) \wedge (x_2 \vee x_3 \vee \neg x_4)$ . Predicate  $P_i$  is “divisible by  $i$ -th prime”.

is  $P_i$  is satisfied), and it is not a multiple of  $p_i$  if  $x_i$  is set to false (that is,  $\neg P_i$  is satisfied). The rest of the proof is identical to that of Proposition 13. ◀

### 4 Discussion: Beyond the number of players

Our model of parameterized game, with constraints on the number of opponents for Eve, is actually a simplification of a general concurrent game model, where the number of players is a parameter. This general model, motivated in introduction, is an extension of the multiplayer concurrent games of [2], where tuples of actions are replaced with languages.

- **Definition 16.** A language-based parameterized arena is a tuple  $\mathcal{A}_L = (V, \Sigma, \Delta_L)$  where
- $V$  is a finite set of vertices;
  - $\Sigma$  is a finite set of actions;
  - $\Delta_L : V \times \Sigma^{\geq 2} \rightarrow 2^V$  is the transition function.

The fact that Eve has at least one opponent explains the term  $\Sigma^{\geq 2}$  in the transition function. We assume that for every  $(v, v') \in V^2$ ,  $\nabla_L(v, v') \stackrel{\text{def}}{=} \{w \in \Sigma^{\geq 2} \mid v' \in \Delta_L(v, w)\}$  is regular. Figure 1 in introduction provides an example of a language-based parameterized arena.

The game is then played as follows, when  $k+1$  is the number of players, called Eve, Adam<sub>1</sub>, ..., Adam<sub>k</sub>: from vertex  $v$ , each of the players select simultaneously and independently an action in  $\Sigma$ ; concatenating all the letters (Eve first, and then all Adams' actions), it forms a word  $w$ ; the next vertex of the game is then one of the vertices  $v'$  in  $\Delta_L(v, w)$ ; the game then resumes from vertex  $v'$ . Strategies for Eve, and outcomes can be defined similarly to that of parameterized arenas in Section 2. The language-based parameterized game problem is then to decide whether Eve has a strategy that is winning against any number of opponents:

#### LANGUAGE-BASED PARAMETERIZED REACHABILITY GAME PROBLEM

**Input:** A language-based parameterized reachability game  $\mathcal{G} = (\mathcal{A}, t)$  and a vertex  $v$ .

**Question:** Does Eve have a winning strategy from  $v$  in  $\mathcal{G}$ ?

Language-based parameterized arenas generalize parameterized arenas: one can for instance replace rules of the form  $v' \in \Delta(v, a, k)$  in a parameterized arena by  $v' \in \Delta_L(v, a\Sigma^k)$  to construct a language-based parameterized arena, preserving the winning region for Eve. For our problem of existence of a winning strategy for Eve, the reduction in the other direction also holds:

► **Proposition 17.** *The language-based parameterized reachability game problem reduces in polynomial time to the parameterized reachability game (with semilinear predicates).*

**Proof sketch.** From a language-based parameterized arena, one can obtain an equivalent one (*i.e.* preserving the winning region for Eve) by first taking a left quotient of languages by any possible letter, and then projecting the obtained languages to lengths of words. Describing the reduction is simpler with the  $\nabla$  functions (and equivalent to using the  $\Delta$  ones). We set  $\nabla(v, a, v') = \{|u| \mid u \in a^{-1}\nabla_L(v, v')\}$ , where  $a^{-1}\nabla_L(v, v')$  is the left quotient by  $a$  of  $\nabla_L(v, v')$ . Since  $\nabla_L(v, v')$  is regular, the set  $\nabla(v, a, v')$  is semilinear [15]. Moreover, one can compute in polynomial time a representation for  $\nabla(v, a, v')$  as a union of polynomially many ultimately periodic sets, with a polynomial encoding [7, 14]. Clearly enough this polynomial time reduction preserves the winning region for Eve. ◀

Thanks to Proposition 17, and using Propositions 11 and 15 we obtain the precise complexity of the language-based parameterized reachability game problem:

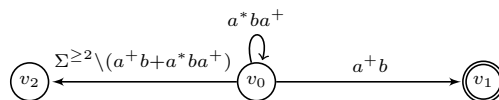
► **Theorem 18.** *The language-based parameterized reachability game problem is PSPACE-complete.*

## 5 Conclusion

In this paper, we introduce parameterized concurrent reachability games as a natural extension of the traditional concurrent games, where the number of players is unknown a priori. We consider different variants of a parameterized arena where the constraints on the number of opponents can be represented by intervals, finite unions of intervals, or semilinear sets. We have shown the existence of a uniform winning strategy for the first player to be PSPACE-complete in the general case, NP-complete when the arena is deterministic and the constraints are unions of intervals, and PTIME-complete when restricting to intervals only.

In this paper, we focused on reachability objectives. However the knowledge game approach also applies to more general objectives, like Büchi or parity, and even for quantitative objectives such as mean-payoff objectives. There is indeed a tight connection between strategies in the original game and strategies in the knowledge game, making the knowledge game abstraction correct for a variety of objectives. We plan to investigate complexity issues for objectives beyond reachability.

In future work, we also wish to investigate further this parameterized games model. In particular, it will be interesting to consider standard game theory concepts such as Nash equilibria. Also, to solve coordination problems, we will look for algorithms to synthesize strategies for all the players to achieve a global common goal. The figure below presents a simple coordination game, where we assume each player has a distinct identifier from 1 to some  $n \in \mathbb{N}$ , and their global objective is to reach the target vertex  $v_1$ .



If the players do not know beforehand the total number of players, but know their identifiers, a winning strategy profile is as follows: player  $i$  plays action  $a$  for the first  $i-1$  steps, then plays  $b$ , and finally plays  $a$  for the remaining steps. Doing so, each player will in turn play action  $b$ , and when the last player does, the play reaches  $v_1$ . Synthesizing automatically winning profiles in such games is one of our long-term goals.

---

References

---

- 1 Luca de Alfaro, Thomas A. Henzinger, and Orna Kupferman. Concurrent reachability games. In *Proceedings of the 39th Annual Symposium on Foundations of Computer Science (FOCS'98)*, pages 564–575. IEEE Computer Society, 1998. doi:10.1109/SFCS.1998.743507.
- 2 Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman. Alternating-time temporal logic. *Journal of the ACM*, 49:672–713, 2002. doi:10.1145/585265.585270.
- 3 Nathalie Bertrand, Miheer Dewaskar, Blaise Genest, and Hugo Gimbert. Controlling a population. In *Proceedings of the 28th International Conference on Concurrency Theory (CONCUR'17)*, volume 85 of *LIPICs*, pages 12:1–12:16. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. doi:10.4230/LIPICs.CONCUR.2017.12.
- 4 Nathalie Bertrand, Paulin Fournier, and Arnaud Sangnier. Playing with probabilities in reconfigurable broadcast networks. In *Proceedings of the 17th International Conference on Foundations of Software Science and Computation Structure (FoSSaCS'14)*, volume 8412 of *Lecture Notes in Computer Science*, pages 134–148. Springer, April 2014. doi:10.1007/978-3-642-54830-7\_9.
- 5 Roderick Bloem, Swen Jacobs, Ayrat Khalimov, Igor Konnov, Sasha Rubin, Helmut Veith, and Josef Widder. *Decidability of Parameterized Verification*. Synthesis Lectures on Distributed Computing Theory. Morgan & Claypool Publishers, 2015. doi:10.2200/S00658ED1V01Y201508DCT013.
- 6 Patricia Bouyer, Romain Brenguier, Nicolas Markey, and Michael Ummels. Pure Nash equilibria in concurrent games. *Logical Methods in Computer Science*, 11(2:9), 2015. doi:10.2168/LMCS-11(2:9)2015.
- 7 Marek Chrobak. Finite automata and unary languages. *Theoretical Computer Science*, 47(3):149–158, 1986. doi:10.1016/0304-3975(86)90142-8.
- 8 Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing (STOC'71)*, pages 151–158. ACM, 1971. doi:10.1145/800157.805047.
- 9 Giorgio Delzanno. Constraint-based verification of parameterized cache coherence protocols. *Formal Methods in System Design*, 23(3):257–301, 2003. doi:10.1023/A:1026276129010.
- 10 Javier Esparza. Keeping a crowd safe: On the complexity of parameterized verification (invited talk). In *Proceedings of the 31st International Symposium on Theoretical Aspects of Computer Science (STACS'14)*, volume 25 of *LIPICs*, pages 1–10. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2014. doi:10.4230/LIPICs.STACS.2014.1.
- 11 Dana Fisman, Orna Kupferman, and Yoav Lustig. Rational synthesis. In *Proceedings of the 16th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'10)*, volume 6015 of *Lecture Notes in Computer Science*, pages 190–201. Springer, 2010. doi:10.1007/978-3-642-12002-2\_16.
- 12 Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors. *Automata, Logics, and Infinite Games: A Guide to Current Research*, volume 2500 of *Lecture Notes in Computer Science*. Springer, 2002. doi:10.1007/3-540-36387-4.
- 13 Igor Konnov, Helmut Veith, and Josef Widder. What you always wanted to know about model checking of fault-tolerant distributed algorithms. In *Proceedings of the 10th International Andrei Ershov Informatics Conference (PSI'15)*, volume 9609 of *Lecture Notes in Computer Science*, pages 6–21. Springer, 2015. doi:10.1007/978-3-319-41579-6\_2.
- 14 Andrew Martinez. Efficient computation of regular expressions from unary NFAs. In *Proceedings of the 5th International Workshop on Descriptive Complexity of Formal Systems (DCFS'02)*, pages 174–187. Department of Computer Science, The University of Western Ontario, Canada, 2002.
- 15 Rohit Parikh. On context-free languages. *Journal of the ACM*, 13(4):570–581, 1966. doi:10.1145/321356.321364.



- 16 Larry J. Stockmeyer and Albert R. Meyer. Word problems requiring exponential time (preliminary report). In *Proceedings of the 5th Annual ACM Symposium on Theory of Computing (STOC'73)*, pages 1–9. ACM, 1973. doi:10.1145/800125.804029.
- 17 Michael Ummels and Dominik Wojtczak. The complexity of Nash equilibria in limit-average games. In *Proceedings of the 22nd International Conference on Concurrency Theory (CONCUR'11)*, volume 6901 of *Lecture Notes in Computer Science*, pages 482–496. Springer, 2011. doi:10.1007/978-3-642-23217-6\_32.