Round-Bounded Control of Parameterized Systems

Benedikt Bollig, Mathieu Lehaut, Nathalie Sznajder

Round-Bounded Control of Parameterized Systems*

Benedikt Bollig[1], Mathieu Lehaut[2], and Nathalie Sznajder[2]

[1] CNRS, LSV & ENS Paris-Saclay, Université Paris-Saclay, France
[2] Sorbonne Université, CNRS, LIP6, F-75005 Paris, France

Abstract. We consider systems with unboundedly many processes that communicate through shared memory. In that context, simple verification questions have a high complexity or, in the case of pushdown processes, are even undecidable. Good algorithmic properties are recovered under round-bounded verification, which restricts the system behavior to a bounded number of round-robin schedules. In this paper, we extend this approach to a game-based setting. This allows one to solve synthesis and control problems and constitutes a further step towards a theory of languages over infinite alphabets.

1 Introduction

Ad-hoc networks, mobile networks, cache-coherence protocols, robot swarms, and distributed algorithms have (at least) one thing in common: They are referred to as *parameterized* systems, as they are usually designed to work for *any* number of processes. The last few years have seen a multitude of approaches to parameterized verification, which aims to ensure that a system is correct no matter how many processes are involved. We refer to [15] for an overview.

Now, the above-mentioned applications are usually part of an open world, i.e., they are embedded into an environment that is not completely under the control of a system. Think of scheduling problems, in which an unspecified number of jobs have to be assigned to (a fixed number of) resources with limited capacity. The arrival of a job and its characteristics are typically not under the control of the scheduler. However, most available verification techniques are only suitable for closed systems: A system is correct if *some* or *every* possible behavior satisfies the correctness criterion, depending on whether one considers reachability or, respectively, linear-time objectives.

This paper is a step towards a theory of synthesis and *control*, which provides a more fine-grained way to reason about parameterized systems. Our system model is essentially that from [24], but defined in a way that reveals similarities with data automata/class-memory automata, a certain automata model over infinite alphabets [8, 9]. Actually, we consider *parameterized pushdown systems*, as each process has a dedicated stack to model recursion. A parameterized pushdown system distinguishes between a finite-state *global process* (sometimes

* Partly supported by ANR FREDDA (ANR-17-CE40-0013).

referred to as a *global store* or *leader process*) and a *local process*. The global process can spawn new local processes. Thus, while a system configuration contains only one global state, the number of instantiations of local processes is unbounded. Moreover, when a local process takes a transition, it is allowed to read, and modify, the global store.

So far so good. Now, it is well-known that reachability is undecidable as soon as two pushdown processes communicate through shared memory. And even when local processes are finite-state, the problem is at least as hard as reachability in Petri nets [9]. This led La Torre, Madhusudan, and Parlato to consider round-bounded verification of parameterized systems, which restricts system executions to a bounded number of round-robin schedules [24]. Not only did they show that reachability drops to PSPACE, but the corresponding fixed-point computation also turned out to be practically feasible. Moreover, they give a sound method (i.e., a sufficient criterion) for proving that all reachable states can already be reached within a bounded number of round-robin schedules. This is done using a game that is different from the one we introduce here. Actually, we extend their model by adding the possibility to distinguish, in parameterized pushdown automata, between controllable global states and uncontrollable ones.

The classical reachability problem then turns into a reachability objective in an infinite-state game. As our main result, it is shown that the winner of such a game can be computed, though in (inherently) non-elementary time. Our proof makes a detour via games on multi-pushdown systems, which are undecidable in general but decidable under a bound on the number of *phases*, each restricting the number of pop operations to a dedicated stack [5,29]. Note that round-robin schedules maintain processes in a queue fashion. However, bounding the number of rounds allows us to store both the states of a local process as well as its stack contents in a configuration of a multi-pushdown system. It is worth noting that multi-pushdown systems have been employed in [23], too, to solve seemingly different verification problems involving queues.

Related Work. As already mentioned, there is a large body of literature on parameterized verification, mostly focusing on closed systems (e.g., [2,4,14,15]).

Infinite-state games have been extensively studied over vector addition systems with states (VASS) (e.g., [3,7,10,12,19]). However, reachability is already undecidable for simple subclasses of VASS games, unless coverability objectives are considered. Unfortunately, the latter do not allow us to require that *all* local processes terminate in a final state. Interestingly, tight links between VASS/energy games and games played on infinite domains have recently been established [16].

Underapproximate verification goes back to Qadeer and Rehof [27]. In the realm of multi-threaded recursive programs, they restricted the number of control switches between different threads. The number of processes, however, was considered to be fixed. Another kind of bounded verification of *parameterized* systems with thread creation was studied in [6]. Contrary to our restriction, the order in which processes evolve may vary from round to round.

We believe that our results will fertilize *synthesis* of parameterized systems [18] and more classical questions whose theoretical foundations go back to the 50s and Church's synthesis problem. Let us cite Brütsch and Thomas, who observed a lack of approaches to synthesis over infinite alphabets [11]: "It is remarkable, however, that a different kind of 'infinite extension' of the Büchi-Landweber Theorem has not been addressed in the literature, namely the case where the input alphabet over which ω-sequences are formed is infinite." Indeed, an execution of a parameterized system can be considered as a sequence of letters, each containing the process identifier of the process involved in performing the corresponding action. Recall that our model of parameterized systems is largely inspired by data automata/class-memory automata [8, 9], which were originally defined as language acceptors over infinite alphabets. The automata studied in [11] are quite different. Since synthesis problems are often reduced to game-theoretic questions, our work can be considered as an orthogonal step towards a theory of synthesis over infinite alphabets.

Outline. We define parameterized pushdown systems in Section 2, where we also recall known results on reachability questions. The control problem is addressed in Section 3, and we conclude in Section 4. Missing proof details can be found in the appendix.

2 Reachability in Parameterized Systems

We start with some preliminary definitions.

Words. Let Σ be a (possibly infinite) set. A *word* w over Σ is a finite or (countably) infinite sequence $a_0a_1a_2\ldots$ of elements $a_i \in \Sigma$. Let Σ^* denote the set of finite words over Σ, Σ^ω the set of infinite words, and $\Sigma^\infty = \Sigma^* \cup \Sigma^\omega$. Given $w \in \Sigma^\infty$, we denote by $|w|$ the *length* of w, i.e., $|w| = n$ if $w = a_0 \ldots a_{n-1} \in \Sigma^*$, and $|w| = \omega$ if $w \in \Sigma^\omega$. In particular, the length $|\varepsilon|$ of the empty word ε is 0.

Transition Systems. A *transition system* is a triple $\mathcal{T} = (V, E, v_{\mathsf{in}})$ such that V is a (possibly infinite) set of *nodes*, $E \subseteq V \times V$ is the *transition relation*, and $v_{\mathsf{in}} \in V$ is the initial node. For $(u, v) \in E$, we call v a *successor* of u.

A *partial run* of $\mathcal{T}$ is a non-empty, finite or infinite sequence $\rho = v_0v_1v_2\ldots \in V^\infty$ such that, for all $0 < i < |\rho|$, v_i is a successor of v_{i-1}. If, in addition, we have $v_0 = v_{\mathsf{in}}$, then we call ρ a *run*. A (partial) run from u to v is a finite (partial) run of the form $u \ldots v$. In particular, u is a partial run (of length 1) from u to u.

2.1 Parameterized Pushdown Systems

We consider parameterized systems in which processes may be created dynamically. Every process can manipulate a stack as well as its *local* state. Information shared by all the processes is modeled in terms of a *global* state.

Definition 1. *A* parameterized pushdown system *(PPS) is given by a tuple* $\mathcal{P} = (S, L, \Gamma, s_{\mathsf{in}}, \ell_{\mathsf{in}}, \Delta, F_{\mathsf{glob}}, F_{\mathsf{loc}})$ *where*

– S *is the finite set of* global states*, including the* initial global state s_{in}*,*
– L *is the finite set of* local states*, including the* initial local state ℓ_{in}*,*
– Γ *is the finite* stack alphabet*,*
– $\Delta \subseteq (S \times L) \times (Act \times \Gamma) \times (S \times L)$ *is the* transition relation *with* $Act = \{\mathsf{push}, \mathsf{pop}, \mathsf{int}\}$ *(where* int *stands for* internal*), and*
– $F_{\mathsf{glob}} \subseteq S$ *and* $F_{\mathsf{loc}} \subseteq L$ *are the sets of* accepting global states *and* accepting local states*, respectively. We assume that* $s_{\mathsf{in}} \notin F_{\mathsf{glob}}$*.*

A *configuration* of $\mathcal{P}$ is a tuple $c = (s, (\ell_1, \gamma_1), \ldots, (\ell_k, \gamma_k))$ where $k \in \mathbb{N}$ (possibly $k = 0$), $s \in S$ is the current global state, and, for each $p \in \{1, \ldots, k\}$, $\ell_p \in L$ and $\gamma_p \in \Gamma^*$ are respectively the local state and stack content of process p. We let $C_{\mathcal{P}}$ denote the set of configurations of $\mathcal{P}$. The *initial configuration* is (s_{in}) and a configuration $c = (s, (\ell_1, \gamma_1), \ldots, (\ell_k, \gamma_k))$ is *final* if $s \in F_{\mathsf{glob}}$ and $\{\ell_1, \ldots, \ell_k\} \subseteq F_{\mathsf{loc}}$. The *size* $|c|$ of a configuration c is the number k of processes in c.

The semantics of a PPS $\mathcal{P}$ is defined as a transition system $[\![\mathcal{P}]\!] = (V, E, v_{\mathsf{in}})$ where $V = C_{\mathcal{P}}$, $v_{\mathsf{in}} = (s_{\mathsf{in}})$, and the transition relation is $E = \bigcup_{p \geq 1} E_p$ with E_p defining the transitions of process p. Actually, E_p contains two types of transitions. The first type corresponds to the activity of a process that has already been created. Formally, for two configurations $(s, (\ell_1, \gamma_1), \ldots, (\ell_k, \gamma_k))$ and $(s', (\ell'_1, \gamma'_1), \ldots, (\ell'_k, \gamma'_k))$ of size $k \geq 1$,

$$((s, (\ell_1, \gamma_1), \ldots, (\ell_k, \gamma_k)), (s', (\ell'_1, \gamma'_1), \ldots, (\ell'_k, \gamma'_k))) \in E_p$$

if and only if $p \leq k$ and there are $op \in Act$ and $A \in \Gamma$ such that

– $((s, \ell_p), (op, A), (s', \ell'_p)) \in \Delta$,
– $\ell_q = \ell'_q$ and $\gamma_q = \gamma'_q$ for all $q \in \{1, \ldots, k\} \setminus \{p\}$, and
– one of the following holds: (i) $op = \mathsf{push}$ and $\gamma'_p = A \cdot \gamma_p$, (ii) $op = \mathsf{pop}$ and $\gamma_p = A \cdot \gamma'_p$, or (iii) $op = \mathsf{int}$ and $\gamma_p = \gamma'_p$ (in which case A is meaningless).

Note that the topmost stack symbol can be found at the leftmost position of γ_p.

The second type of transition is when a new process joins the system. For a configuration $(s, (\ell_1, \gamma_1), \ldots, (\ell_k, \gamma_k))$ of size $k \geq 0$,

$$((s, (\ell_1, \gamma_1), \ldots, (\ell_k, \gamma_k)), (s', (\ell_1, \gamma_1), \ldots, (\ell_k, \gamma_k), (\ell_{k+1}, \gamma_{k+1}))) \in E_p$$

if and only if $p = k + 1$ and there are $op \in Act$ and $A \in \Gamma$ such that $((s, \ell_{\mathsf{in}}), (op, A), (s', \ell_{k+1})) \in \Delta$ and one of the following holds: (i) $op = \mathsf{push}$ and $\gamma_{k+1} = A$, or (ii) $op = \mathsf{int}$ and $\gamma_{k+1} = \varepsilon$.

A *run* of $\mathcal{P}$ is a run of the transition system $[\![\mathcal{P}]\!]$. A finite run of $\mathcal{P}$ is *accepting* if it ends in a final configuration.

Similarly, we define a *parameterized finite-state system (PFS)*, which is a PPS without stacks. That is, a PFS is a tuple $\mathcal{P} = (S, L, s_{\mathsf{in}}, \ell_{\mathsf{in}}, \Delta, F_{\mathsf{glob}}, F_{\mathsf{loc}})$ where $\Delta \subseteq (S \times L) \times (S \times L)$ and the rest is defined as in PPS. Configurations in $C_{\mathcal{P}}$ are

tuples $c = (s, \ell_1, \ldots, \ell_k)$ with $k \geq 0$. The semantics of $\mathcal{P}$ is $[\![\mathcal{P}]\!] = (C_{\mathcal{P}}, E, (s_{\mathsf{in}}))$ with $E = \bigcup_{p \geq 1} E_p$ defined as follows:

$$((s, \ell_1, \ldots, \ell_k), (s', \ell'_1, \ldots, \ell'_k)) \in E_p$$

if and only if $p \leq k$, $((s, \ell_p), (s', \ell'_p)) \in \Delta$, and $\ell_q = \ell'_q$ for all $q \neq p$, and

$$((s, \ell_1, \ldots, \ell_k), (s', \ell_1, \ldots, \ell_k, \ell_{k+1})) \in E_p$$

if and only if $p = k + 1$ and $((s, \ell_{\mathsf{in}}), (s', \ell_{k+1})) \in \Delta$. The notions of runs and accepting runs are defined accordingly.

Reachability Problems. Consider Table 1. The problem PPS-Reachability (respectively, PFS-Reachability) consists in deciding if, in a given PPS (respectively, PFS), there is an accepting run, starting in the initial configuration.

In the general case, these problems are already known and we recall here the results. The first is folklore (cf. also [28]), as two stacks are already sufficient to simulate a Turing machine. For the second, we observe that parameterized systems without stacks are essentially Petri nets (cf. [9]).

Theorem 1. PPS-Reachability *is undecidable, while* PFS-Reachability *is decidable (and as hard as Petri-net reachability).*

2.2 Round-Bounded Behaviors

To regain decidability in the case of PPS, we restrict ourselves to runs that are *round-bounded*, a notion introduced in [24]. Intuitively, during a *round*, the first process will do any number of transitions (possibly 0), then the second process will do any number of transitions, and so on. Once process $p + 1$ has started performing transitions, process p cannot act again in this round. A run is then said to be B-*round bounded* if it uses at most B rounds. Formally, given a natural number $B \geq 1$ and a PPS $\mathcal{P} = (S, L, \Gamma, s_{\mathsf{in}}, \ell_{\mathsf{in}}, \Delta, F_{\mathsf{glob}}, F_{\mathsf{loc}})$, we define the *bounded semantics* of $\mathcal{P}$ as the transition system $[\![\mathcal{P}]\!]^B = (V^B, E^B, v^B_{\mathsf{in}})$ where

- nodes are *enhanced configurations* of the form $v = (c, p, r)$ with $c \in C_{\mathcal{P}}$ a configuration, say, of size k, $p \in \{0, \ldots, k\}$ represents the last process that made a transition (or 0 if it is not yet defined), and $r \in \{1, \ldots, B\}$ is the number of the current round,
- the initial node is $v^B_{\mathsf{in}} = ((s_{\mathsf{in}}), 0, 1)$, and
- there is an edge between (c, p, r) and (c', p', r') if, in $[\![\mathcal{P}]\!] = (V, E, v_{\mathsf{in}})$, there is an edge (c, c') in $E_{p'}$ and either
  - $p' \geq p$ and $r' = r$, or
  - $p' < p$, $r < B$, and $r' = r + 1$.

The bounded semantics of a PFS is defined accordingly.

A B-*run* (or simply *run* if B is understood) of $\mathcal{P}$ is a run of $[\![\mathcal{P}]\!]^B$. A B-run is *accepting* if it is finite and ends in a node (c, p, r) where c is a final configuration.

Consider the problems on the right-hand side of Table 1 (note that B is encoded in unary). Deciding the existence of an accepting B-run is PSPACE-complete for both PPS and PFS.

Table 1. Reachability Problems

PPS-Reachability	PPS-Reachability$^{\text{rb}}$
I: PPS $\mathcal{P}$	**I:** PPS $\mathcal{P}$; $B \geq 1$ (given in unary)
Q: Is there an accepting run of $\mathcal{P}$?	**Q:** Is there an accepting B-run of $\mathcal{P}$?
PFS-Reachability	PFS-Reachability$^{\text{rb}}$
I: PFS $\mathcal{P}$	**I:** PFS $\mathcal{P}$; $B \geq 1$ (given in unary)
Q: Is there an accepting run of $\mathcal{P}$?	**Q:** Is there an accepting B-run of $\mathcal{P}$?

Theorem 2. PPS-Reachability$^{\text{rb}}$ *and* PFS-Reachability$^{\text{rb}}$ *are PSPACE-complete.*

The rest of this section is devoted to the proof of this theorem. Actually, we prove that PPS-Reachability$^{\text{rb}}$ is in PSPACE and PFS-Reachability$^{\text{rb}}$ is PSPACE-hard. The upper bound has already been stated in [24], the lower bound in [25], for a similar model. For the sake of completeness, we give proofs for both bounds.

PPS-Reachability$^{\text{rb}}$ is in PSPACE. We give an (N)PSPACE algorithm solving the problem PPS-Reachability$^{\text{rb}}$ using a slight variant of the notion of interfaces as described in [24]. Let $\mathcal{P} = (S, L, \Gamma, s_{\mathsf{in}}, \ell_{\mathsf{in}}, \Delta, F_{\mathsf{glob}}, F_{\mathsf{loc}})$ be a PPS and $B \geq 1$ be the maximal number of rounds.

An interface for a single process is a triple $\mathcal{I} = [t, (s_1, \ldots, s_B), (s'_1, \ldots, s'_B)] \in \{1, \ldots, B\} \times S^B \times S^B$ satisfying the following conditions:

1. For all $1 \leq i < t$, we have $s_i = s'_i$.
2. There are local states $\ell_{t-1}, \ldots, \ell_B$ and stack contents $\gamma_{t-1}, \ldots, \gamma_B$ such that (i) for all $t \leq i \leq B$ there is a finite partial run in $[\![\mathcal{P}]\!]$ from $c_i = (s_i, (\ell_{i-1}, \gamma_{i-1}))$ to $c'_i = (s'_i, (\ell_i, \gamma_i))$, (ii) this run has length at least two (i.e., it performs at least one transition) if $i = t$, and (iii) ℓ_{t-1} is the initial local state, $\gamma_{t-1} = \varepsilon$, and ℓ_B is an accepting local state.

We refer to the first B-tuple of $\mathcal{I}$ as $\mathcal{I}^\ell$ and to the second B-tuple as $\mathcal{I}^r$. The natural number t is the starting round and is referred to as $t_{\mathcal{I}}$. We say that an interface $\mathcal{I}_1$ is *compatible* with an interface $\mathcal{I}_2$ if $t_{\mathcal{I}_1} \leq t_{\mathcal{I}_2}$ and $\mathcal{I}_1^r = \mathcal{I}_2^\ell$.

Intuitively, an interface represents the possibility of a computation of a single process during a run of the PPS. Global states are the only piece of information needed to be able to coordinate between different processes, since a process cannot access the local content of another one. Moreover, when a process is created, it takes the last position in a round. The starting round t of each interface is needed to check that the order of the processes respects the order of their creation. In other words, interfaces can be viewed as the skeleton of a run of $\mathcal{P}$. This is formalised in the following lemma, which is illustrated in Figure 1 and whose proof can be found in Appendix A.

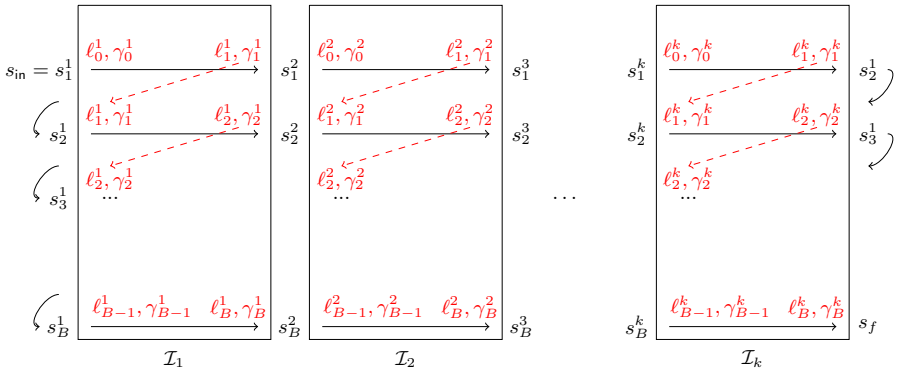

Fig. 1. A run as the composition of compatible interfaces; all starting rounds are 1

Lemma 1. *There is an accepting B-run of $\mathcal{P}$ if and only if there are k interfaces $\mathcal{I}_1, \ldots, \mathcal{I}_k$ for $k \geq 1$ verifying the following conditions:*

- *For all $1 < i \leq k$, $\mathcal{I}_{i-1}$ is compatible with $\mathcal{I}_i$.*
- *Let $\mathcal{I}_1^\ell = (s_1, \ldots, s_B)$ and $\mathcal{I}_k^r = (s'_1, \ldots, s'_B)$. Then, s_1 is the initial global state s_{in}, s'_B is an accepting global state, and $s_j = s'_{j-1}$ for all $1 < j \leq B$.*

Given $\mathcal{I} = [t, (s_1, \ldots, s_B), (s'_1, \ldots, s'_B)]$, one can check in polynomial time whether $\mathcal{I}$ is an interface. To do this, we check the emptiness of a pushdown automaton that simulates the actions of $\mathcal{P}$ on a single process and has special transitions to change the global state from s'_j to s_{j+1} (cf. Appendix B). As non-emptiness of a pushdown automaton can be checked in polynomial time [17], so can the validity of a given interface.

The algorithm to solve PPS-Reachability$^{\text{rb}}$ first guesses an interface $\mathcal{I}_1$ for the first process, and stores $t_{\mathcal{I}_1}$, $\mathcal{I}_1^\ell$, and $\mathcal{I}_1^r$. Then, it guesses an interface $\mathcal{I}_2$ for the second process, checks that it is compatible by comparing $t_{\mathcal{I}_2}$ and $\mathcal{I}_2^\ell$ with the previously stored $t_{\mathcal{I}_1}$ and $\mathcal{I}_1^r$, and then replaces $\mathcal{I}_1^r$ by $\mathcal{I}_2^r$ and $t_{\mathcal{I}_1}$ by $t_{\mathcal{I}_2}$ (so only $\mathcal{I}_1^\ell$, $t_{\mathcal{I}_2}$, and $\mathcal{I}_2^r$ are stored). We continue guessing compatible interfaces, storing at each step i the values of $\mathcal{I}_1^\ell$, $t_{\mathcal{I}_i}$, and $\mathcal{I}_i^r$. Eventually, the algorithm guesses that the last process has been reached. At that point, there are two halves of interfaces stored in memory: the left interface $\mathcal{I}_1^\ell = (s_1, \ldots, s_B)$ of the first process, and the right interface $\mathcal{I}_k^r = (s'_1, \ldots, s'_B)$ of the last process. We accept if, for all $i \in \{1, \ldots, B-1\}$, we have that $s'_i = s_{i+1}$, $s_1 = s_{\mathsf{in}}$, and $s'_B \in F_{\mathsf{glob}}$. By Lemma 1, there is an accepting B-run of $\mathcal{P}$.

PFS-Reachability$^{\text{rb}}$ is PSPACE-hard. This can be shown by a reduction from the non-emptiness of the intersection of a collection of finite automata $\mathcal{A}_1, \ldots, \mathcal{A}_n$, which is PSPACE-complete [21]. The bound B on the number of rounds will be n. We construct a PFS that non-deterministically guesses a word w in the first round. Moreover, in round i, it will check that w is accepted by $\mathcal{A}_i$. To do this, each process simulates one transition of $\mathcal{A}_i$ on one letter of w. That is, the number of processes is $|w|$. Each process performs exactly one action each round,

and, to ensure that the word w is the same for each $\mathcal{A}_i$, stores the corresponding letter in its local state. The global state stores the state of the currently simulated automaton. The full proof can be found in Appendix C.

3 Round-Bounded Control of Parameterized Systems

We will extend parameterized pushdown systems to a game-based setting with the aim of modeling systems with a centralized control that are embedded into an uncontrollable environment.

3.1 Parameterized Pushdown Games

Games. A *game* is given by an *arena*, i.e., a transition system $\mathcal{G} = (V, E, v_{\mathsf{in}})$ where $V = V_0 \uplus V_1$ is partitioned into the set of states controlled by Player 0 and Player 1, respectively, along with a winning condition $\mathcal{W} \subseteq V^\infty$.

A *play* of $\mathcal{G}$ is a run of the underlying transition system. A play is *maximal* if it is infinite, or ends in a node that has no successor. A maximal play is *winning* for Player 0 if it is in $\mathcal{W}$, otherwise it is winning for Player 1.

We will be concerned with two winning conditions: A *reachability condition* is given by a set of nodes $\mathcal{F} \subseteq V$. It induces the set $\mathcal{W}_\mathcal{F} = \{\rho = v_0 v_1 v_2 \ldots \in V^\infty \mid v_i \in \mathcal{F} \text{ for some } 0 \leq i < |\rho|\}$. A *parity condition* is given by a ranking function $\alpha : V \to \mathit{Col}$ where $\mathit{Col} \subseteq \mathbb{N}$ is a finite set of colors. It induces the set $\mathcal{W}_\alpha = \{\rho \in V^\omega \mid \min(\mathrm{Inf}_\alpha(\rho)) \text{ is even}\}$ with $\mathrm{Inf}_\alpha(v_0 v_1 v_2 \ldots) = \{m \in \mathit{Col} \mid m \text{ appears infinitely often in } \alpha(v_0)\alpha(v_1)\alpha(v_2)\ldots\}$. I.e., $\mathcal{W}_\alpha$ contains an infinite run if and only if the minimal color seen infinitely often is even.

Let $j \in \{0, 1\}$. A *strategy* for Player j is a partial mapping $f_j : V^* V_j \to V$ such that, for all $w \in V^*$ and $v \in V_j$, the following hold: if $f_j(wv)$ is defined, then $(v, f_j(wv)) \in E$; otherwise, v has no successor.

Fix strategies f_0 and f_1 for Players 0 and 1, respectively. An (f_0, f_1)-*play* of $\mathcal{G}$ is a maximal play $\rho = v_0 v_1 v_2 \ldots$ such that, for all $0 < i < |\rho|$ and $j \in \{0, 1\}$, if $v_{i-1} \in V_j$, then $f_j(v_0 \ldots v_{i-1}) = v_i$.

We say that f_j is *winning* if, for all strategies f_{1-j}, the unique maximal (f_0, f_1)-play is winning for Player j. A game is *determined* if either Player 0 has a winning strategy, or Player 1 has a winning strategy. Furthermore, we say that f_j is *memoryless* if, for all $w, w' \in V^*$ and $v \in V_j$, we have $f_j(wv) = f_j(w'v)$, i.e., the strategy only depends on the last node.

Theorem 3 (cf. [13, 33]). *Games with a parity winning condition are determined, and if Player j has a winning strategy, then Player j has a winning memoryless strategy.*

Parameterized Pushdown Games. We now introduce the special case of games played on the infinite transition system induced by a round-bounded PPS.

A round-bounded parameterized pushdown game is described by a PPS $\mathcal{P} = (S, L, \Gamma, s_{\mathsf{in}}, \ell_{\mathsf{in}}, \Delta, F_{\mathsf{glob}}, F_{\mathsf{loc}})$ together with a partition $S = S_0 \uplus S_1$. For a bound $B \geq 1$, the *B-round-bounded parameterized pushdown game* induced by $\mathcal{P}$ is the

game $\mathcal{G}_{\mathcal{P}}^{B}$ given by the transition system $[\![\mathcal{P}]\!]^{B} = (V^{B}, E^{B}, v_{\mathsf{in}}^{B})$ where a node $v = (c, p, r) \in V^{B}$ with $c = (s, (\ell_1, \gamma_1), \ldots, (\ell_k, \gamma_k))$ belongs to Player j if $s \in S_j$. We consider the reachability winning condition $\mathcal{W}_{\mathcal{F}}$ given by $\mathcal{F} = \{(c, p, r) \in V^{B} \mid c \text{ is a final configuration of } \mathcal{P}\}$. Since a reachability game can be easily transformed into a parity game, Theorem 3 implies that $\mathcal{G}_{\mathcal{P}}^{B}$ is determined.

Parameterized games on PFS are defined similarly as for PPS. Note that, without a bound on the number of rounds, games on PFS are already undecidable, which is shown by an easy adaptation of the undecidability proof for VASS games [1]. Therefore, we only define control for round-bounded games:

CONTROL$^{\mathrm{rb}}$
I: PPS $\mathcal{P} = (S_0 \uplus S_1, L, \Gamma, s_{\mathsf{in}}, \ell_{\mathsf{in}}, \Delta, F_{\mathsf{glob}}, F_{\mathsf{loc}})$; $B \geq 1$
Q: Does Player 0 have a winning strategy in $\mathcal{G}_{\mathcal{P}}^{B}$?

We are now ready to present our main result, which is shown in the remainder of this section:

Theorem 4. CONTROL$^{\mathrm{rb}}$ *is decidable, and inherently non-elementary.*

3.2 Upper bound

Decidability of CONTROL$^{\mathrm{rb}}$ comes from decidability of games on phase-bounded multi-pushdown systems (short: multi-pushdown games), which were first studied in [29] and rely on the phase-bounded multi-pushdown automata from [22].

Multi-Pushdown Games. Intuitively, a *phase* is a sequence of actions in a run during which only one fixed "active" stack can be read (i.e., either make a pop transition or a zero-test transition), but push and internal transitions are unrestricted. There are no other constraints on the number of transitions or the order of the transitions done during a phase.

Definition 2. *A* multi-pushdown system (MPS) *is a tuple* $\mathcal{M} = (\kappa, N, S_0 \uplus S_1, \Gamma, \Delta, s_{\mathsf{in}}, \alpha)$ *where the natural number* $\kappa \geq 1$ *is the* phase bound*,* $N \in \mathbb{N}$ *is the number of stacks,* $S = S_0 \uplus S_1$ *is the partitioned finite set of* states*,* Γ *is the finite* stack alphabet*,* $\Delta \subseteq S \times Act_{\mathsf{zero}} \times \{1, \ldots, N\} \times \Gamma \times S$ *is the* transition relation *where* $Act_{\mathsf{zero}} = \{\mathsf{push}, \mathsf{pop}, \mathsf{int}, \mathsf{zero}\}$*,* $s_{\mathsf{in}} \in S$ *is the initial state, and* $\alpha : S \to Col$ *with* $Col \subseteq \mathbb{N}$ *a finite set is the* ranking function*.*

The associated game $\mathcal{G}_{\mathcal{M}}$ is then played on the transition system $[\![\mathcal{M}]\!] = (V = V_0 \uplus V_1, E, v_{\mathsf{in}})$ defined as follows.

A node $v \in V$ is of the form $v = (s, \gamma_1, \ldots, \gamma_N, st, ph)$ where $s \in S$, $\gamma_\sigma \in \Gamma^*$ is the content of stack σ, and $st \in \{0, \ldots, N\}$ and $ph \in \{1, \ldots, \kappa\}$ are used to keep track of the current active stack (0 when it is undefined) and the current phase, respectively. For $j \in \{0, 1\}$, we let $V_j = \{(s, \gamma_1, \ldots, \gamma_N, st, ph) \in V \mid s \in S_j\}$.

Given nodes $v = (s, \gamma_1, \ldots, \gamma_N, st, ph) \in V$ and $v' = (s', \gamma'_1, \ldots, \gamma'_N, st', ph') \in V$, we have an edge $(v, v') \in E$ if and only if there exist $op \in Act_{\mathsf{zero}}$, $\sigma \in \{1, \ldots, N\}$, and $A \in \Gamma$ such that $(s, op, \sigma, A, s') \in \Delta$ and the following hold:

- $\gamma_\tau = \gamma'_\tau$ for all $\tau \neq \sigma$,
- $\gamma_\sigma = \gamma'_\sigma$ if $op = \mathsf{int}$, $\gamma'_\sigma = A \cdot \gamma_\sigma$ if $op = \mathsf{push}$, $\gamma_\sigma = A \cdot \gamma'_\sigma$ if $op = \mathsf{pop}$, and $\gamma_\sigma = \gamma'_\sigma = \varepsilon$ if $op = \mathsf{zero}$,
- if $op \in \{\mathsf{int}, \mathsf{push}\}$, then $st = st'$ and $ph = ph'$ (the active stack and, hence, the phase do not change),
- if $op \in \{\mathsf{pop}, \mathsf{zero}\}$, then either $st = 0$, $st' = \sigma$, and $ph = ph' = 1$ (this is the first time a current stack is defined), or $st = \sigma$, $st' = \sigma$, and $ph = ph'$ (the stack σ corresponds to the current active stack), or $st \neq \sigma$, $ph < \kappa$, $st' = \sigma$, and $ph' = ph + 1$ (stack σ is not the active stack so that a new phase starts).

The initial node is $v_{\mathsf{in}} = (s_{\mathsf{in}}, \varepsilon, \ldots, \varepsilon, 0, 1)$. The winning condition of $\mathcal{G}_\mathcal{M}$ is a parity condition given by $\overline{\alpha} : V \to Col$ where, for $v = (s, \gamma_1, \ldots, \gamma_N, st, ph)$, we let $\overline{\alpha}(v) = \alpha(s)$.

The control problem for MPS, denoted by $\textsc{Control}^{\text{MPS}}$, is defined as follows: Given an MPS $\mathcal{M}$, does Player 0 have a winning strategy in $\mathcal{G}_\mathcal{M}$?

Theorem 5 ([5,29]). $\textsc{Control}^{\text{MPS}}$ *is decidable, and is non-elementary in the number of phases.*

The upper bound was first shown in [29] by adopting the technique from [32], which reduces pushdown games to games played on finite-state arenas. On the other hand, [5] proceeds by induction on the number of phases, reducing a $(\kappa+1)$-phase game to a κ-phase game. Similarly, we could try a direct proof of our Theorem 4 by induction on the number of rounds. However, this proof would be very technical and essentially reduce round-bounded parameterized systems to multi-pushdown systems. Therefore, we proceed by reduction to multi-pushdown games, providing a modular proof with clearly separated parts.

From Parameterized Pushdown Games to Multi-Pushdown Games. We reduce $\textsc{Control}^{\text{rb}}$ to $\textsc{Control}^{\text{MPS}}$. Let $\mathcal{P} = (S, L, \Gamma, s_{\mathsf{in}}, \ell_{\mathsf{in}}, \Delta, F_{\mathsf{glob}}, F_{\mathsf{loc}})$, with $S = S_0 \uplus S_1$, be a PPS and $B \geq 1$. We will build an MPS $\mathcal{M}$ such that Player 0 has a winning strategy in $\mathcal{G}^B_\mathcal{P}$ if and only if Player 0 has a winning strategy in $\mathcal{G}_\mathcal{M}$. In the following, given $s \in S$, we let $pl(s) \in \{0, 1\}$ denote the player associated with s, i.e., $pl(s) = 0$ if and only if $s \in S_0$.

The main idea of the reduction is to represent a configuration

$$(s, (\ell_1, \boxed{\uparrow \gamma_1}), \ldots, (\ell_{p-1}, \boxed{\uparrow \gamma_{p-1}}), (\ell_p, \boxed{\uparrow \gamma_p}), (\ell_{p+1}, \boxed{\uparrow \gamma_{p+1}}), \ldots, (\ell_k, \boxed{\uparrow \gamma_k}), p, r)$$

of $\mathcal{G}^B_\mathcal{P}$ as a configuration in $\mathcal{G}_\mathcal{M}$ of the form depicted in Figure 2.

Component $j \in \{0, 1\}$ of the global state denotes the current player (which, by default, is $pl(s)$). We explain f_1 and f_2 further below.

The process p that has moved last is considered as the *active* process whose local state ℓ_p is kept in the global state of $\mathcal{G}_\mathcal{M}$ along with s, and whose stack contents γ_p is accessible on stack 1 (in the correct order). This allows the multi-pushdown game to simulate transitions of process p, modifying its local state and stack contents accordingly (see *Basic Transitions* in the formalization below).

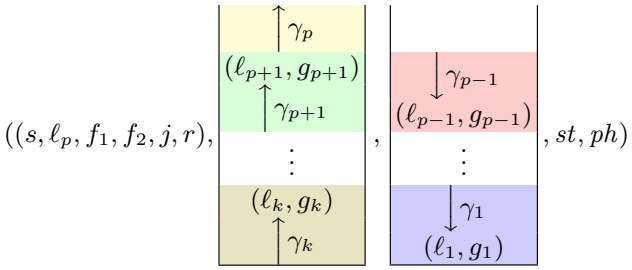

Fig. 2. Encoding of a configuration in $\mathcal{G}_{\mathcal{P}}^{B}$ by a configuration in $\mathcal{G}_{\mathcal{M}}$

If a player decides to take a transition for some process $p' > p$, she will store ℓ_p on stack 2 and shift the contents of stack 1 onto stack 2 until she retrieves the local state $\ell_{p'}$ of p' along with its stack contents $\gamma_{p'}$ (see Figure 3 and *Transitions for Process Change* in the formalization of $\mathcal{M}$).

If, on the other hand, the player decides to take a transition for some process $p' < p$, then she stores ℓ_p on stack 1 and shifts the contents of stack 2 onto stack 1 to recover the local state $\ell_{p'}$ and stack contents $\gamma_{p'}$ (see Figure 4 and *Transitions for Round Change*). This may imply two phase switches, one to shift stack symbols from 2 to 1, and another one to continue simulating the current process on stack 1. However, $2B - 1$ phases are sufficient to simulate B rounds.

There are a few subtleties: First, at any time, we need to know whether the current configuration of $\mathcal{G}_{\mathcal{M}}$ corresponds to a final configuration in $\mathcal{G}_{\mathcal{P}}^{B}$. To this aim, the state component $(s, \ell_p, f_1, f_2, j, r)$ of $\mathcal{M}$ contains the flags $f_1, f_2 \in \{\checkmark, \times\}$ where, as an invariant, we maintain $f_1 = \checkmark$ if and only if $\{\ell_{p+1}, \ldots, \ell_k\} \subseteq F_{\mathsf{loc}}$ and $f_2 = \checkmark$ if and only if $\{\ell_1, \ldots, \ell_{p-1}\} \subseteq F_{\mathsf{loc}}$. Thus, Player 0 wins in $\mathcal{G}_{\mathcal{M}}$ as soon as she reaches a configuration with global state $(s, \ell, f_1, f_2, j, r)$ such that $s \in F_{\mathsf{glob}}$, $\ell \in F_{\mathsf{loc}}$, and $f_1 = f_2 = \checkmark$. To faithfully maintain the invariant, every local state ℓ_q that is pushed on one of the two stacks, comes with an additional flag $g_q \in \{\checkmark, \times\}$, which is $\checkmark$ if and only if all local states *strictly below* on the stack are contained in F_{loc}. It is then possible to keep track of a property of *all* local states on a given stack simply by inspecting and locally updating the topmost stack symbols.

Second, one single transition in $\mathcal{P}$ is potentially simulated by several transitions in $\mathcal{M}$ in terms of the gadgets given in Figures 3 and 4. The problem here is that once Player j commits to taking a transition by entering a gadget, she is not allowed to get stuck. To ensure progress, there are transitions from inside a gadget to a state win_{1-j} that is winning for Player $1 - j$.

Third, suppose that, in a non-final configuration of $\mathcal{G}_{\mathcal{P}}^{B}$, it is Player 1's turn, but no transition is available. Then, Player 1 wins the play. But how can Player 1 prove in $\mathcal{G}_{\mathcal{M}}$ that no transition is available in the original game $\mathcal{G}_{\mathcal{P}}^{B}$? Actually, he will give the control to Player 0, who will eventually get stuck and, therefore, lose (cf. transitions for *Change of Player* below).

Let us define the MPS $\mathcal{M} = (\kappa, N, S' = S'_0 \uplus S'_1, \Gamma', \Delta', s'_{\mathsf{in}}, \alpha)$ formally. We let $\kappa = 2B - 1$, $N = 2$ (the number of stacks), and $\Gamma' = \Gamma \uplus (L \times \{\checkmark, \text{✗}\})$.

States. The set of states is $S' = \{s'_{\mathsf{in}}\} \uplus S_{\mathrm{sim}} \uplus \{\mathrm{win}_0, \mathrm{win}_1\} \uplus \mathfrak{I}$ where s'_{in} is the initial state. Moreover, $S_{\mathrm{sim}} = S \times L \times \{\checkmark, \text{✗}\}^2 \times \{0,1\} \times \{1, \ldots, B\}$. A state $(s, \ell, f_1, f_2, j, r) \in S_{\mathrm{sim}}$ stores the global state s and the local state ℓ of the last process p that executed a transition. The third and forth component f_1 and f_2 tell us whether all processes $p' > p$ and, respectively, $p' < p$ of the current configuration are in a local final state (indicated by $\checkmark$). Then, j denotes the player that is about to play (usually, we have $j = pl(s)$, but there will be deviations). Finally, r is the current round that is simulated. Recall that $(s, \ell, f_1, f_2, j, r)$ represents a final configuration if and only if $s \in F_{\mathsf{glob}}$, $\ell \in F_{\mathsf{loc}}$, and $f_1 = f_2 = \checkmark$. Let $\mathfrak{F} \subseteq S_{\mathrm{sim}}$ be the set of such states. The states win_0 and win_1 are self-explanatory. Finally, we use several intermediate states, contained in $\mathfrak{I}$, which will be determined below along with the transitions.

The partition $S' = S'_0 \uplus S'_1$ is defined as follows: First, we have $s'_{\mathsf{in}} \in S'_{pl(s_{\mathsf{in}})}$. Concerning states from S_{sim}, we let $(s, \ell, f_1, f_2, j, r) \in S'_j$. The states win_0 and win_1 both belong to Player 0 (but this does not really matter). Membership of intermediate states is defined below. The ranking function α maps win_0 to 0, and everything else to 1. In fact, we only need a reachability objective and use the parity condition to a very limited extent.

Initial Transitions. For all transitions $(s_{\mathsf{in}}, \ell_{\mathsf{in}}) \xrightarrow{(op,A)} (s', \ell')$ in $\mathcal{P}$, we introduce, in $\mathcal{M}$, a transition $s'_{\mathsf{in}} \xrightarrow{(op,1,A)} (s', \ell', \checkmark, \checkmark, pl(s'), 1)$.

Final Transitions. For all states $(s, \ell, f_1, f_2, j, r) \in \mathfrak{F}$, we will have a transition $(s, \ell, f_1, f_2, j, r) \xrightarrow{\mathsf{int}} \mathrm{win}_0$ (we omit the stack symbol, as it is meaningless), which will be the only transition outgoing from $(s, \ell, f_1, f_2, j, r)$. Moreover, $\mathrm{win}_0 \xrightarrow{\mathsf{int}} \mathrm{win}_0$ and $\mathrm{win}_1 \xrightarrow{\mathsf{int}} \mathrm{win}_1$.

Basic Transitions. We now define the transitions of $\mathcal{M}$ simulating transitions of $\mathcal{P}$ that do not change the process. For all $(s, \ell, f_1, f_2, j, r) \in S_{\mathrm{sim}} \setminus \mathfrak{F}$ and transitions $(s, \ell) \xrightarrow{(op,A)} (s', \ell')$ from Δ (in $\mathcal{P}$), the MPS $\mathcal{M}$ has a transition $(s, \ell, f_1, f_2, j, r) \xrightarrow{(op,1,A)} (s', \ell', f_1, f_2, pl(s'), r)$.

Transitions for Process Change. For all $(s, \ell, f_1, f_2, j, r) \in S_{\mathrm{sim}} \setminus \mathfrak{F}$, we introduce, in $\mathcal{M}$, the gadget given in Figure 3. As we move to another process, the current local state ℓ is pushed on stack 2, along with flag f_2, which tells us whether, henceforth, all states on stack 2 *below* the new stack symbol are local accepting states. Afterwards, the value of f_2 kept in the global state has to be updated, depending on whether $\ell \in F_{\mathsf{loc}}$ or not. Actually, maintaining the value of f_2 is done in terms of additional (but finitely many) states. For the sake of readability, however, we rather consider that f_2 is a variable and use $\mathsf{upd}(f_2, \ell)$ to update its value. We continue shifting the contents of stack 1 onto stack 2 (updating f_2 when retrieving a local state). Now, there are two possibilities. We

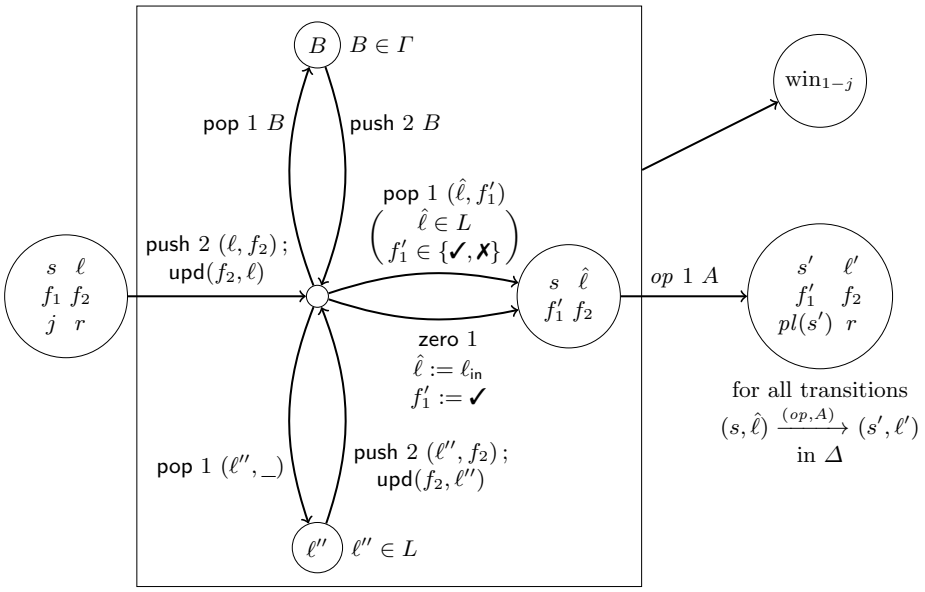

Fig. 3. Change from process p to some process $p' > p$ (staying in the same round). All intermediate states belong to Player j; from every intermediate state, there is an outgoing internal transition to win_{1-j}. Moreover, $\mathsf{upd}(f_2, \bar{\ell})$ stands for the update rule IF $(f_2 = \checkmark \wedge \bar{\ell} \in F_{\mathsf{loc}})$ THEN $f_2 := \checkmark$ ELSE $f_2 := \times$.

may eventually pop a new current local state $\hat{\ell}$ and then simulate the transition of the corresponding existing process. Or, when there are no more symbols on stack 1, we create a new process.

Transitions for Round Change. For all $(s, \ell, f_1, f_2, j, r) \in S_{\text{sim}} \setminus \mathfrak{F}$ such that $r < B$, we introduce, in $\mathcal{M}$, the gadget given in Figure 4. It is similar to the previous gadget. However, we now shift symbols from stack 2 onto stack 1 and have to update f_1 accordingly.

Change of Player. When Player 1 thinks he does not have an outgoing transitions (in $\mathcal{P}$), he can give the token to Player 0. That is, for all $(s, \ell, f_1, f_2, 1, r) \in S_{\text{sim}} \setminus \mathfrak{F}$, we introduce the transition $(s, \ell, f_1, f_2, 1, r) \xrightarrow{\mathsf{int}} (s, \ell, f_1, f_2, 0, r)$.

Lemma 2. *Player 0 has a winning strategy in $\mathcal{G}_{\mathcal{M}}$ if and only if Player 0 has a winning strategy in $\mathcal{G}_{\mathcal{P}}^{B}$.*

The proof of this lemma can be found in Appendix D and E.

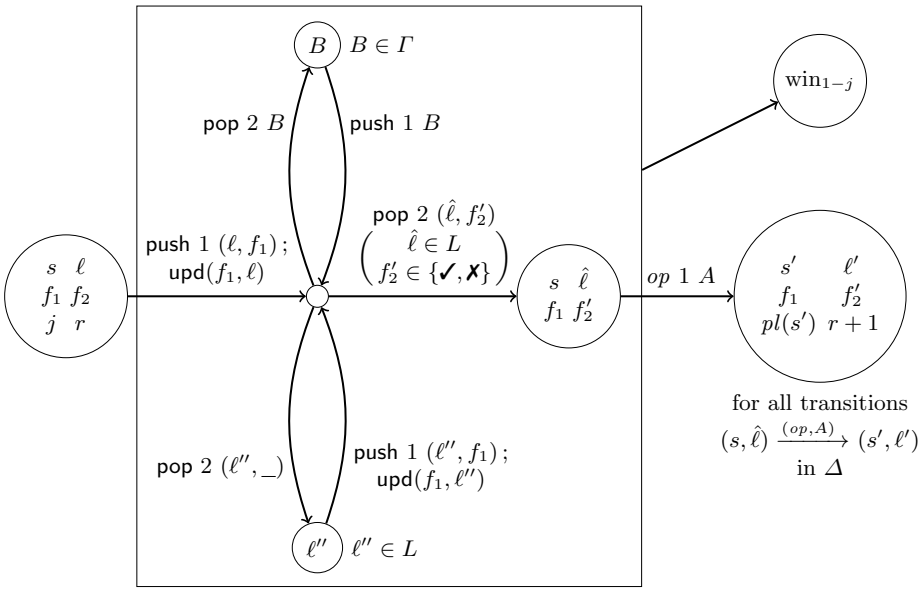

Fig. 4. Go from a process p to some process $p' < p$ (involving a round change). All intermediate states belong to Player j; from every intermediate state, there is an outgoing internal transition to win_{1-j}. Moreover, $\mathsf{upd}(f_1, \bar{\ell})$ stands for the update rule If $(f_1 = \checkmark \wedge \bar{\ell} \in F_{\mathsf{loc}})$ Then $f_1 := \checkmark$ Else $f_1 := \times$.

3.3 Lower bound

Our lower-bound proof is inspired by [5], but we reduce from the satisfiability problem for first-order formulas on finite words, which is known to be non-elementary [30]. Note that the lower bound already holds for PFS.

Let Var be a countably infinite set of variables and Σ a finite alphabet. Formulas φ are built by the grammar $\varphi ::= a(x) \mid x < y \mid \neg(x < y) \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \exists x.\varphi \mid \forall x.\varphi$ where $x, y \in \mathrm{Var}$ and $a \in \Sigma$.

Let $w = a_0 \dots a_{n-1} \in \Sigma^*$ be a word. Variables are interpreted as positions of w, so a valuation is a (partial) function $\nu : \mathrm{Var} \to \{0, \dots, n-1\}$. The satisfaction relation is defined as follows. We let $w, \nu \models a(x)$ if and only if $a_{\nu(x)} = a$. Moreover, $w, \nu \models x < y$ if and only if $\nu(x) < \nu(y)$. Quantification, negation, disjunction, and conjunction are defined as usual. We refer to [31] for details. A formula φ without free variables is *satisfiable* if there is a word w such that $w, \emptyset \models \varphi$. We suppose that φ is given in prenex normal form.

We build a PFS-based round-bounded game that is winning for Player 0 if and only if φ is satisfiable. In the first round of the game, Player 0 chooses a word w by creating a different process for each letter of w, each of them holding the corresponding letter in its local state. To prove that w is indeed a model of φ, the following rounds are devoted to the valuation of the variables appearing

in φ, $\nu(x) = i$ being represented by memorizing the variable x in the local state of the i^{th} process. If x appears in the scope of a universal quantifier, the choice of the process is made by Player 1, otherwise it is made by Player 0. The last round is used to check the valuation of the variables. To this end, the players will inductively choose a subformula to check, until they reach an atomic proposition: If the subformula is a disjunction $\varphi_1 \vee \varphi_2$, Player 0 chooses either φ_1 or φ_2; if it is a conjunction, Player 1 chooses the next subformula. Finally, to verify whether $a(x)$ is satisfied, we check that there is a process with letter a and variable x in its local state. For $x < y$, we check that the process with x in its local state is eventually followed by a distinct process with y in its local state. This check is done during the same round, which guarantees that the positions corresponding to x and y are in the correct order. The number of states needed and the number of rounds are linearly bounded in the length of the formula. A formalization of this construction and its correctness proof can be found in Appendix F.

4 Conclusion

We extended the verification of round-bounded parameterized systems to a game-based setting, which allows us to model an uncontrollable environment. It would be interesting to consider game-based extensions for the setting from [6], too. Moreover, as games constitute an important approach to verifying branching-time properties (e.g., [26]), our results may be used for branching-time model checking of parameterized systems (using a variant of data logics [20] and a reduction of the model-checking problem to a parameterized pushdown game).

References

1. P. A. Abdulla, A. Bouajjani, and J. d'Orso. Deciding monotonic games. In *CSL'03*, volume 2803 of *LNCS*, pages 1–14. Springer, 2003.
2. P. A. Abdulla and G. Delzanno. Parameterized verification. *Int. J. Softw. Tools Technol. Transf.*, 18(5):469–473, October 2016.
3. P. A. Abdulla, R. Mayr, A. Sangnier, and J. Sproston. Solving parity games on integer vectors. In *CONCUR'13*, volume 8052, pages 106–120. Springer, 2013.
4. B. Aminof, S. Jacobs, A. Khalimov, and S. Rubin. Parameterized model checking of token-passing systems. In *VMCAI'14*, volume 8318 of *LNCS*, pages 262–281. Springer, 2014.
5. M. F. Atig, A. Bouajjani, K. Narayan Kumar, and P. Saivasan. Parity games on bounded phase multi-pushdown systems. In *NETYS'17*, volume 10299 of *LNCS*, pages 272–287, 2017.
6. M. F. Atig, A. Bouajjani, and S. Qadeer. Context-bounded analysis for concurrent programs with dynamic creation of threads. *Log. Methods Comput. Sci.*, 7(4), 2011.
7. B. Bérard, S. Haddad, M. Sassolas, and N. Sznajder. Concurrent games on VASS with inhibition. In *CONCUR'12*, volume 7454 of *LNCS*, pages 39–52. Springer, 2012.
8. H. Björklund and T. Schwentick. On notions of regularity for data languages. *Theoretical Computer Science*, 411(4-5):702–715, 2010.

9. M. Bojańczyk, C. David, A. Muscholl, T. Schwentick, and L. Segoufin. Two-variable logic on data words. *ACM Trans. Comput. Log.*, 12(4):27, 2011.
10. T. Brázdil, P. Jancar, and A. Kucera. Reachability games on extended vector addition systems with states. In *ICALP'10, Part II*, volume 6199 of *LNCS*, pages 478–489. Springer, 2010.
11. B. Brütsch and W. Thomas. Playing games in the Baire space. In *Proc. Cassting Workshop on Games for the Synthesis of Complex Systems and 3rd Int. Workshop on Synthesis of Complex Parameters*, volume 220 of *EPTCS*, pages 13–25, 2016.
12. J.-B. Courtois and S. Schmitz. Alternating vector addition systems with states. In *MFCS'14*, volume 8634 of *LNCS*, pages 220–231. Springer, 2014.
13. E. A. Emerson and C. S. Jutla. Tree automata, mu-calculus and determinacy. In *Proceedings of FOCS'91*, pages 368–377. IEEE Computer Society, 1991.
14. E. A. Emerson and K. S. Namjoshi. On reasoning about rings. *Int. J. Found. Comput. S.*, 14(4):527–550, 2003.
15. J. Esparza. Keeping a crowd safe: On the complexity of parameterized verification. In *STACS'14*, volume 25 of *Leibniz International Proceedings in Informatics*, pages 1–10. Leibniz-Zentrum für Informatik, 2014.
16. D. Figueira and M. Praveen. Playing with repetitions in data words using energy games. In *Proceedings of LICS'18*, pages 404–413. ACM, 2018.
17. J. E. Hopcroft, R. Motwani, Rotwani, and J. D. Ullman. *Introduction to Automata Theory, Languages and Computability.* Addison-Wesley Longman Publishing Co., Inc., 2nd edition, 2000.
18. S. Jacobs and R. Bloem. Parameterized synthesis. *Log. Methods Comput. Sci.*, 10(1), 2014.
19. P. Jancar. On reachability-related games on vector addition systems with states. In *RP'15*, volume 9328 of *LNCS*, pages 50–62. Springer, 2015.
20. A. Kara. *Logics on data words: Expressivity, satisfiability, model checking.* PhD thesis, Technical University of Dortmund, 2016.
21. D. Kozen. Lower bounds for natural proof systems. In *Proceedings of SFCS'77*, pages 254–266. IEEE Computer Society, 1977.
22. S. La Torre, P. Madhusudan, and G. Parlato. A robust class of context-sensitive languages. In *LICS'07*, pages 161–170. IEEE Computer Society Press, 2007.
23. S. La Torre, P. Madhusudan, and G. Parlato. Context-bounded analysis of concurrent queue systems. In *Proceedings of TACAS'08*, volume 4963 of *LNCS*, pages 299–314. Springer, 2008.
24. S. La Torre, P. Madhusudan, and G. Parlato. Model-checking parameterized concurrent programs using linear interfaces. In *CAV'10*, volume 6174 of *LNCS*, pages 629–644. Springer, 2010.
25. S. La Torre, P. Madhusudan, and G. Parlato. Model-checking parameterized concurrent programs using linear interfaces. Technical Report 2142/15410, University of Illinois, 2010. Available at http://hdl.handle.net/2142/15410.
26. M. Lange and C. Stirling. Model checking games for branching time logics. *J. Log. Comput.*, 12(4):623–639, 2002.
27. S. Qadeer and J. Rehof. Context-bounded model checking of concurrent software. In *TACAS'05*, volume 3440 of *LNCS*, pages 93–107. Springer, 2005.
28. G. Ramalingam. Context-sensitive synchronization-sensitive analysis is undecidable. *ACM Trans. Program. Lang. Syst.*, 22(2):416–430, 2000.
29. A. Seth. Games on multi-stack pushdown systems. In *LFCS'09*, volume 5407 of *LNCS*, pages 395–408. Springer, 2009.
30. L. J. Stockmeyer. *The Complexity of Decision Problems in Automata Theory and Logic.* PhD thesis, MIT, 1974.

31. W. Thomas. Languages, automata and logic. In A. Salomaa and G. Rozenberg, editors, *Handbook of Formal Languages*, volume 3, pages 389–455. Springer, 1997.
32. I. Walukiewicz. Pushdown processes: Games and model-checking. *Inf. Comput.*, 164(2):234–263, 2001.
33. W. Zielonka. Infinite games on finitely coloured graphs with applications to automata on infinite trees. *TCS*, 200(1-2):135–183, 1998.

A Proof of Lemma 1

Given a configuration $c = (s, (\ell_1, \gamma_1), \ldots, (\ell_k, \gamma_k))$, we let $proj_S(c) = s$, and $proj_{L,i}(c) = (\ell_i, \gamma_i)$ if $i \leq k$ and $proj_{L,i}(c) = (\ell_{\mathsf{in}}, \varepsilon)$ if $k < i$.

Let ρ be an accepting run of $[\![\mathcal{P}]\!]^B$ using k processes. Note that ρ can be divided into $\rho = \rho^0\rho^1\cdots\rho^{B'}$, with $B' \leq B$, where each ρ^j is the part of the run corresponding to round j, and $\rho^0 = ((s_{\mathsf{in}}), 0, 1)$. For every $1 \leq i \leq k$ and $1 \leq j \leq B'$, let $(c_\alpha^{(i,j)})_{\alpha \leq n(i,j)}$ be the (possibly empty) finite sequence of configurations visited by process i during round j in the run ρ. Formally, for each $1 \leq j \leq B'$, $\rho^j = \rho^{1j}\cdots\rho^{kj}$, with $\rho^{ij} = (c_1^{(i,j)}, i, j)\cdots(c_{n(i,j)}^{(i,j)}, i, j)$. Note that ρ^{ij} may be empty. Fix a round $1 \leq j \leq B'$, and a process $1 \leq i \leq k$. Let i',j' be such that $\rho^{i'j'}$ is the last non-empty part of ρ before ρ^{ij}. Let $s_0^{(i,j)} = proj_S(c_{n(i',j')}^{(i',j')})$ and $(\ell_0^{(i,j)}, \gamma_0^{(i,j)}) = proj_{L,i}(c_{n(i',j')}^{(i',j')})$ be respectively the global state and local state reached by process i just before process i starts in round j. Let $s_m^{(i,j)} = proj_S(c_m^{(i,j)})$ and $(\ell_m^{(i,j)}, \gamma_m^{(i,j)}) = proj_{L,i}(c_m^{(i,j)})$, for $1 \leq m \leq n_{(i,j)}$. Since $((c_{n(i',j')}^{(i',j')}, i', j'), (c_1^{(i,j)}, i, j)) \in E^B$ and, for all $1 \leq m < n(i,j)$, $((c_m^{(i,j)}, i, j), (c_{m+1}^{(i,j)}, i, j)) \in E^B$, by definition, we have $(c_{n(i',j')}^{(i',j')}, c_1^{(i,j)}) \in E_i$ and, for all $1 \leq m < n(i,j)$, $(c_m^{(i,j)}, c_{m+1}^{(i,j)}) \in E_i$. Observe that this implies that $((s_m^{(i,j)}, (\ell_m^{(i,j)}, \gamma_m^{(i,j)})), (s_{m+1}^{(i,j)}, (\ell_{m+1}^{(i,j)}, \gamma_{m+1}^{(i,j)}))) \in E$, for all $0 \leq m < n(i,j)$. Hence there is a finite run in $[\![\mathcal{P}]\!]$ from $(s_0^{(i,j)}, (\ell_0^{(i,j)}, \gamma_0^{(i,j)}))$ to $(s_{n(i,j)}^{(i,j)}, (\ell_{n(i,j)}^{(i,j)}, \gamma_{n(i,j)}^{(i,j)}))$. If ρ^{ij} is empty we let $s_{n(i,j)}^{(i,j)} = s_0^{(i,j)}$ and $(\ell_{n(i,j)}^{(i,j)}, \gamma_{n(i,j)}^{(i,j)})) = ((\ell_0^{(i,j)}, \gamma_0^{(i,j)}))$. For all $B' < j \leq B$, for all $1 \leq i \leq k$, let $n(i,j) = 0$, and $s_0^{(i,j)} = s_{n(k,B')}^{(k,B')}$, and $(\ell_0^{(i,j)}, \gamma_0^{(i,j)}) = (\ell_{n(i,B')}^{(i,B')}, \gamma_{n(i,B')}^{(i,B')})$, i.e. the global state stays unchanged since the end of the run ρ, and the local state and stack of each process stays at it was at the end of their local run in the last round B'.

For each process i, let t_i be the smallest round $1 \leq j \leq B'$ such that ρ^{ij} is not empty. Thus, for all $1 \leq i \leq k$, we define the interface $\mathcal{I}_i = [t_i, (s_0^{(i,1)}, \ldots, s_0^{(i,B)}), (s_{n(i,1)}^{(i,1)}, \ldots, s_{n(i,B)}^{(i,B)})]$. From the above, we know that there exists a finite run in $[\![\mathcal{P}]\!]$ from $(s_0^{(i,j)}, (\ell_0^{(i,j)}, \gamma_0^{(i,j)}))$ to $(s_{n(i,j)}^{(i,j)}, (\ell_{n(i,j)}^{(i,j)}, \gamma_{n(i,j)}^{(i,j)}))$, for all $1 \leq j \leq B$. Moreover, in the first round, if a process plays, it is for the first time. Hence, by definition, $(\ell_0^{(i,1)}, \gamma_0^{(i,1)}) = proj_{L,i}(c)$ for a configuration c of size smaller than i. Then, $(\ell_0^{(i,1)}, \gamma_0^{(i,1)}) = (\ell_{\mathsf{in}}, \varepsilon)$. Since ρ is winning, every process ends in a final local state, i.e., $\ell_{n(i,B)}^{(i,B)}$ is accepting, for all i. Finally, for all $j < t_i$, ρ^{ij} is empty. By construction then, $s_0^{(i,j)} = s_{n(i,j)}^{(i,j)}$. All of this ensures that $\mathcal{I}_i$ is indeed an interface. By construction, for all process $1 < i \leq k$ and for all rounds $j \geq 1$ we have $s_0^{(i,j)} = s_{n(i-1,j)}^{(i-1,j)}$. Moreover, by definition of a run of $\mathcal{P}$, if a process i appears in round j, then necessarily, process $i+1$ appears for the first time in round $j' \geq j$. Hence for all $1 \leq i < k$, $t_i \leq t_{i+1}$ and interface $\mathcal{I}_i$ is compatible with interface $\mathcal{I}_2$. By construction, for all $1 \leq j \leq B$, $s_0^{(1,j)} = s_{n(k,j-1)}^{(k,j-1)}$, and

because ρ is winning we also have $s_0^{(1,1)} = s_{\mathsf{in}}$ and $s_{n_{(k,B)}}^{(k,B)}$ is accepting, then all the conditions of the lemma are fulfilled.

Conversely, let $\mathcal{I}_1, \ldots, \mathcal{I}_k$ be interfaces verifying the conditions. Let $s^{(i,j)}$ be the j-th component of $\mathcal{I}_i^\ell$ and $s'^{(i,j)}$ be the j-th component of $\mathcal{I}_i^r$ and consider the following partial run in $[\![\mathcal{P}]\!]$: $\rho_{ij} = (s_1^{(i,j)}, (\ell_1^{(i,j)}, \gamma_1^{(i,j)})) \ldots (s_{n(i,j)}^{(i,j)}, (\ell_{n(i,j)}^{(i,j)}, \gamma_{n(i,j)}^{(i,j)}))$ with $s_1^{(i,j)} = s^{(i,j)}$ and $s_{n(i,j)}^{i,j} = s'^{(i,j)}$. The existence of such a run, as well as the local states and stack contents, is ensured by the definition of an interface. Moreover, $(\ell_1^{(i,j)}, \gamma_1^{(i,j)}) = (\ell_{n(i,j-1)}^{(i,j-1)}, \gamma_{n(i,j-1)}^{(i,j-1)})$, for all $j > 1$ and $(\ell_1^{(i,j)}, \gamma_1^{(i,j)}) = (\ell_{\mathsf{in}}, \varepsilon)$ if $j = 1$.

We build a run ρ of $\mathcal{P}$ as follows. For $1 \leq j \leq B$, and $1 \leq i \leq k$ such that $t_{\mathcal{I}_i} \leq j$, let $\rho^{(i,j)}$ be the (possibly empty) sequence of nodes $(c_2^{(i,j)}, i, j), \ldots, (c_{n(i,j)}^{(i,j)}, i, j)$ where $c_m^{(i,j)} = (s_m^{(i,j)}, (\ell_1, \gamma_1), \ldots, (\ell_{k'}, \gamma_{k'}))$ with $k' \leq k$ the greatest element of the set $\{1 \leq i' \leq k \mid t_{\mathcal{I}_{i'}} < j\} \cup \{i\}$ and

- $\ell_i = \ell_m^{(i,j)}$ and $\gamma_i = \gamma_m^{(i,j)}$,
- for $p < i$, $\ell_p = \ell_{n(p,j)}^{(p,j)}$ (i.e. the last local state of process p) and $\gamma_p = \gamma_{n(p,j)}^{(p,j)}$,
- for $i < p \leq k'$, $\ell_p = \ell_{n(p,j-1)}^{(p,j-1)}$ and $\gamma_p = \gamma_{n(p,j-1)}^{(p,j-1)}$ (same but with the previous round)

For i such that $t_{\mathcal{I}_i} > j$, $\rho^{(i,j)} = \varepsilon$, and we let $\rho^{(0,0)} = ((s_{\mathsf{in}}), 0, 1)$.

We show that the sequence $\rho = \rho^{(0,0)}\rho^{(1,1)} \ldots \rho^{(k,1)}\rho^{(1,2)} \ldots \rho^{(k,B)}$ is a run in $[\![\mathcal{P}]\!]^B$ by induction on the pair (i,j). The base case where $i = 0, j = 0$ is trivial. Let $1 \leq i < k$ and $1 \leq j \leq B$ and assume that $\rho^{(0,0)}\rho^{(1,1)} \ldots \rho^{(i,j)}$ is a run. Let (i', j') be the successor of (i,j).

- Suppose that $j = j'$ and $i' = i + 1$. Without loss of generality, assume that $\rho^{(i,j)}$ and $\rho^{(i+1,j)}$ are not empty. Let $k' = |c_2^{(i+1,j)}|$. If $k' > i + 1$, then $k' = |c_{n(i,j)}^{(i,j)}|$ and $t_{\mathcal{I}_i} < j$ by compatibility of the interfaces, hence $j > 1$. Then,
$$c_{n(i,j)}^{(i,j)} = (s_{n(i,j)}^{(i,j)}, (\ell_1, \gamma_1) \ldots (\ell_{k'}, \gamma_{k'}))$$
with
$$(\ell_{i+1}, \gamma_{i+1}) = (\ell_{n(i+1,j-1)}^{(i+1,j-1)}, \gamma_{n(i+1,j-1)}^{(i+1,j-1)}) = (\ell_1^{(i+1,j)}, \gamma_1^{(i+1,j)}))$$
and
$$c_2^{(i+1,j)} = (s_2^{(i+1,j)}, (\ell'_1, \gamma'_1) \ldots (\ell'_{k'}, \gamma'_{k'}))$$
with
$$(\ell'_{i+1}, \gamma'_{i+1}) = (\ell_2^{(i+1,j)}, \gamma_2^{(i+1,j)})).$$
Moreover, for all $p \leq i$, $(\ell_i, \gamma_i) = (\ell'_i, \gamma'_i) = (\ell_{n(p,j)}^{(p,j)}, \gamma_{n(p,j)}^{(p,j)})$ and for all $i + 1 < p \leq k'$, $(\ell_p, \gamma_p) = (\ell'_p, \gamma'_p) = (\ell_{n(p,j-1)}^{(p,j-1)}, \gamma_{n(p,j-1)}^{(p,j-1)})$. Since $s_1^{(i+1,j)} = s^{(i+1,j)} = s'^{(i,j)} = s_{n(i,j)}^{(i,j)}$ because $\mathcal{I}_i$ is compatible with $\mathcal{I}_{i+1}$, and by definition of an interface, $((s_{n(i,j)}^{(i,j)}, (\ell_{i+1}, \gamma_{i+1})), (s_2^{(i+1,j)}, (\ell'_{i+1}, \gamma'_{i+1}))) \in E$. Hence

from the above equalities, $((c^{(i,j)}_{n(i,j)}, i, j), (c^{(i+1,j)}_2, i+1, j)) \in E^B$. Again, by definition of an interface and by definition of $\rho^{(i+1,j)}$, $\rho^{(i+1,j)}$ is a partial run of E^B. Hence $\rho^{(0,0)} \cdots \rho^{(i+1,j)}$ is a run in $[\![\mathcal{P}]\!]^B$. Now, suppose that $k' = i+1$. Either $|c^{(i,j)}_{n(i,j)}| = i+1$ (if $t_{\mathcal{I}_i} = t_{\mathcal{I}_{i+1}} = j$ or if $t_{\mathcal{I}_{i+1}} < j$), or $|c^{(i,j)}_{n(i,j)}| = i$ (if $t_{\mathcal{I}_{i+1}} = j$ and $t_{\mathcal{I}_i} < j$). In the first case, with the same reasonning than above, we can conclude. In the second case, it means that $c^{(i,j)}_{n(i,j)} = (s^{(i,j)}_{n(i,j)}, (\ell_1, \gamma_1) \dots (\ell_i, \gamma_i))$ and $c^{(i+1,j)}_2 = (s^{(i+1,j)}_2, (\ell'_1, \gamma'_1) \dots (\ell'_{i+1}, \gamma'_{i+1})$ with $(\ell'_{i+1}, \gamma'_{i+1}) = (\ell^{(i+1,j)}_2, \gamma^{(i+1,j)}_2)$. Since $t_{\mathcal{I}_{i+1}} = j$, by definition of the interface, we know that $(\ell^{(i+1,j)}_1, \gamma^{(i+1,j)}_1) = (\ell_{\mathsf{in}}, \varepsilon)$ and that $\big((s^{(i+1,j)}, (\ell_{\mathsf{in}}, \varepsilon)), (s^{(i+1,j)}_2, (\ell^{(i+1,j)}_2, \gamma^{(i+1,j)}_2))\big) \in E$. It is then easy to check that we have $((c^{(i,j)}_{n(i,j)}, i, j), (c^{(i+1,j)}_2, i+1, j)) \in E^B$ and we can conclude as above.

– Suppose now that $j' = j+1$. Necessarily, since $t_{\mathcal{I}_{i'}} \leq t_{\mathcal{I}_i}$, $|c^{(i,j)}_{n(i,j)}| = |c^{(i',j+1)}_2|$. Again, without loss of generality, suppose that $i = k$ and $i' = 1$. Then

$$\begin{aligned} c^{(1,j+1)}_2 &= (s^{(1,j+1)}_2, (\ell'_1, \gamma'_1), \dots, (\ell'_k, \gamma'_k)) \\ &= (s^{(1,j+1)}_2, (\ell^{(1,j+1)}_2, \gamma^{(1,j+1)}_2), \dots, (\ell^{(k,j)}_{n(k,j)}, \gamma^{(k,j)}_{n(k,j)})), \end{aligned}$$

and

$$\begin{aligned} c^{(k,j)}_{n(k,j)} &= (s^{(k,j)}_{n(k,j)}, (\ell_1, \gamma_1), \dots, (\ell_k, \gamma_k)) \\ &= (s^{(k,j)}_{n(k,j)}, (\ell^{(1,j)}_{n(1,j)}, \gamma^{(1,j)}_{n(1,j)}), \dots, (\ell^{(k,j)}_{n(k,j)}, \gamma^{(k,j)}_{n(k,j)})). \end{aligned}$$

By the second condition of the lemma, one can conclude that $s^{(k,j)}_{n(k,j)} = s'^{(k,j)} = s^{(1,j+1)} = s^{(1,j+1)}_1$. The existence of the run ρ_{1j+1} allows to conclude.

As we used the runs in $[\![\mathcal{P}]\!]$ given by the interfaces to build $\rho^{(i,j)}$, then using the same transitions of $\mathcal{P}$ shows that $\rho^{(i,j)}$ is a valid run in $[\![\mathcal{P}]\!]^B$. Since $\mathcal{I}_i$ and $\mathcal{I}_{i+1}$ are compatible (for $i < k$), using the first step $(s^{(i,j)}_1, (\ell^{(i,j)}_1, \gamma^{(i,j)}_1)) \to (s^{(i,j)}_2, (\ell^{(i,j)}_2, \gamma^{(i,j)}_2))$ shows that there is also an edge in $[\![\mathcal{P}]\!]^B$ from $(c^{(i,j)}_{n(i,j)}, i, j)$ to $(c^{(i+1,j)}_2, i+1, j)$. Similarly, the second condition of the lemma ensures that there is an edge from $(c^{(k,j)}_{n(k,j)}, k, j)$ to $(c^{(1,j+1)}_{n(1,j+1)}, 1, j+1)$ for $j < B$.

Finally, the second condition of the lemma ensures that the last global state is accepting, thus ρ is accepting.

Observe that we have supposed in this proof that no $\rho^{(i,j)}$ was empty. If this was not the case, the global state will remain the same in the different interfaces, and the same proof applies, with tedious modifications of the indices. Again, if a whole round is missing, or if the run does not start in round 1, then one can rewrite the numbers of the round in a correct way, with no other modification, and with the number of rounds even smaller than before.

B Checking Validity of an Interface

We define the pushdown automaton $\mathcal{A}_{\mathcal{I}} = (Q, \Sigma, \Gamma \uplus \{Z\}, \delta, q_0, Z, F)$ with $Q = \{\text{Win}\} \cup (S \times L \times \{1, \ldots, B\}) \cup \left(\overline{S} \times L \times \{1, \ldots, B\}\right)$ with $\overline{S} = \{\overline{s} \mid s \in S\}$, $q_0 = (s_1, \ell_0, 1)$ if $t > 1$, $q_0 = (\overline{s_1}, \ell_0, 1)$ otherwise, $F = \{\text{Win}\}$, and δ defined as follows. For every $X, A \in \Gamma$, for every $s, s' \in S$, for every $\ell, \ell' \in L$, for every $j \in 1, \ldots, B$, for every $a \in \Sigma$,

1. $((s, \ell, j), X) \xrightarrow{a} ((s', \ell', j), AX)$ if $j \geq t$, $a \in \Sigma_{\mathsf{push}}$ and $((s, \ell), (a, A), (s', \ell')) \in \Delta$,
   $((s, \ell, j), A) \xrightarrow{a} ((s', \ell', j), \varepsilon)$ if $j \geq t$, $a \in \Sigma_{\mathsf{pop}}$ and $((s, \ell), (a, A), (s', \ell')) \in \Delta$,
   $((s, \ell, j), X) \xrightarrow{a} ((s', \ell', j), X)$ if $j \geq t$, $a \in \Sigma_{\mathsf{int}}$ and $((s, \ell), (a, A), (s', \ell')) \in \Delta$,
2. $((\overline{s}, \ell, j), X) \xrightarrow{a} ((s', \ell', j), AX)$ if $a \in \Sigma_{\mathsf{push}}$ and $((s, \ell), (a, A), (s', \ell')) \in \Delta$,
   $((\overline{s}, \ell, j), A) \xrightarrow{a} ((s', \ell', j), \varepsilon)$ if $a \in \Sigma_{\mathsf{pop}}$ and $((s, \ell), (a, A), (s', \ell')) \in \Delta$,
   $((\overline{s}, \ell, j), X) \xrightarrow{a} ((s', \ell', j), X)$ if $a \in \Sigma_{\mathsf{int}}$ and $((s, \ell), (a, A), (s', \ell')) \in \Delta$,
3. $((s'_j, \ell, j), X) \xrightarrow{\varepsilon} ((s_{j+1}, \ell, j+1), X)$ if $j < B$ and $j \neq t - 1$,
4. $((s'_{t-1}, \ell, t-1), X) \xrightarrow{\varepsilon} ((\overline{s_t}, \ell, t), X)$,
5. $((s'_B, \ell, B-1), X) \xrightarrow{\varepsilon} (\text{Win}, X)$ if $\ell \in F_{\mathsf{loc}}$.

Intuitively, the two first kind of transitions corresponds to the transitions of $\mathcal{P}$ for a single process, while the third and fourth kinds allow to non-deterministically change the global state from s'_j to s_{j+1}, and then check correspondence between the next components of the interface (and in $\mathcal{P}$ corresponds to other processes acting and modifying the global state). The third component of Q allows us to track the index of the component of the interface we are following. $\mathcal{A}_{\mathcal{I}}$ accepts only after taking each transition of the third or fourth kind exactly once, and after that reaching global state s'_B with an accepting local state, which is the only way to reach the final state Win through the last kind of transition. The second and fourth kind of transitions lead to a copy of the global state in the round corresponding to the first round of the interface, to ensure that the run being simulated is not empty.

The correctness of our construction is established in the following lemma.

Lemma 3. *There is an accepting run of $\mathcal{A}_{\mathcal{I}}$ if and only if $\mathcal{I}$ is an interface.*

Proof. Suppose there is an accepting run of $\mathcal{A}_{\mathcal{I}}$. By construction, it is necessarily of the form $\rho^1 \cdot \rho^2 \cdots \rho^B \cdot (Win, \gamma_B)$, where, for all $1 \leq j \leq B$, $\rho^j = ((s^1_j, \ell^1_j, j), \gamma^1_j) \cdots ((s^{i_j}_j, \ell^{i_j}_j, j), \gamma^{i_j}_j)$ with $s^1_j = s_j$, the j-th component of $\mathcal{I}^\ell$ if $j \neq t$, and $s^1_j = \overline{s_j}$ otherwise, $s^{i_j}_j = s'_j$, the j-th component of $\mathcal{I}^r$, and for all $1 \leq i \leq i_j$, $\gamma^i_j = Z \cdot {\gamma'}^i_j$, for some ${\gamma'}^i_j \in \Gamma^*$. Moreover, for all $1 \leq j \leq B - 1$, $\ell^{i_j}_j = \ell^1_{j+1}$. Finally, for all $1 \leq i < i_j$, $((s^i_j, (\ell^i_j, {\gamma'}^i_j)), (s^{i+1}_j, (\ell^{i+1}_j, \gamma^{i+1}_j))) \in E_1$.

For all $j < t$, by construction, no transition of the first kind can be taken, which ensures that $\rho^j = ((s^1_j, \ell^1_j, j), \gamma^1_j)$ and thus that $s_j = s^1_j = s'_j$. Again, by the structure of $\mathcal{A}_{\mathcal{I}}$, $|\rho^t| > 1$.

For $t-1 \leq j < B$, let $\ell_j = \ell^1_{j+1}$, $\gamma_j = {\gamma'}^1_{j+1}$, $\ell_B = \ell_B^{i_B}$ and $\gamma_B = \gamma_B^{i_B}$. Then, for all $j \geq t$, there is a partial run in $[\![\mathcal{P}]\!]$ starting in $c_j = (s^1_j, (\ell^1_j, \gamma^1_j)) = (s_j, (\ell_{j-1}, \gamma_{j-1}))$ to $c'_j = (s^{i_j}_j, (\ell^{i_j}_j, \gamma^{i_j}_j) = (s'_j, (\ell^1_{j+1}, \gamma^1_{j+1})) = (s'_j, (\ell_j, \gamma_j))$.

Since ρ^1 starts in $((s_1, \ell_{\mathsf{in}}, 1), Z)$, and since all run ρ^j for $1 \leq j < t$, $\rho^j = ((s_1, \ell_0, j), Z)$, by construction $\ell_{t-1} = \ell_{\mathsf{in}}$, and $\gamma_{t-1} = \varepsilon$. By construction of $\mathcal{A}_{\mathcal{I}}$, $\ell_B = \ell_B^{i_B} \in F_{\mathsf{loc}}$. So, $\mathcal{I}$ is indeed an interface.

Conversely, suppose $\mathcal{I}$ is an interface. One can build an accepting run of $\mathcal{A}_{\mathcal{I}}$: $\rho^1 \cdots \rho^t \cdots \rho^B$, where $\rho^j = ((s_j, \ell_0, j), Z)$ for all $1 \leq j \leq t-1$. For $t \leq j \leq B$, let

$$(s_j, (\ell_{j-1}, \gamma_{j-1}))(s^j_1, (\ell^j_1, \gamma^j_1)) \dots (s^j_{i_j-1}, (\ell^j_{i_j-1}, \gamma^j_{i_j-1}))(s'_j, (\ell_j, \gamma_j))$$

be the run of $[\![\mathcal{P}]\!]$ ensured by $\mathcal{I}$. Then, when $j = t$, $\rho^t = ((\overline{s_t}, \ell_{\mathsf{in}}, t), Z)((s^t_1, \ell^t_1, t), \gamma^t_1 \cdot Z) \dots ((s^j_{i_t-1}, \ell^t_{i_t-1}, t), \gamma^t_{i_t-1} \cdot Z)((s'_t, \ell_t, t)\gamma_t \cdot Z)$. If $t < j < B$, $\rho^j = ((s_j, \ell_{j-1}, j), \gamma_{j-1} \cdot Z)((s^j_1, \ell^j_1, j), \gamma^j_1 \cdot Z) \dots ((s^j_{i_j-1}, \ell^j_{i_j-1}, j), \gamma^j_{i_j-1} \cdot Z)((s'_j, \ell_j, j)\gamma_j \cdot Z)$, which is a run of $\mathcal{A}_{\mathcal{I}}$ using only transitions of the first type. Finally, $\rho^B = \rho^j = ((s_j, \ell_{j-1}, j), \gamma_{j-1} \cdot Z)(s^j_1, \ell^j_1, j), \gamma^j_1 \cdot Z) \dots (s^j_{i_j-1}, \ell^j_{i_j-1}, j), \gamma^j_{i_j-1} \cdot Z)(s'_j, \ell_j, j)\gamma_j \cdot Z)(\mathrm{Win}, \gamma_j \cdot Z)$. The last transition is possible because ℓ_B is accepting.

Transitions from ρ^j to ρ^{j+1} are transitions of the third type (fourth type when $j = t-1$). When $j < t$, it is possible since for all $1 \leq j < t$, $s_j = s'_j$. Moreover, the transition from ρ^t to ρ^{t+1} is possible since the run of $[\![\mathcal{P}]\!]$ is not empty, so the last configuration reached in ρ^t is necessarily in $S \times L \times t$.

C Lower Bound of Theorem 2

Let $\mathcal{A}_1, \dots, \mathcal{A}_n$ be n finite automata over a finite alphabet Σ. That is, $\mathcal{A}_i = (Q_i, \delta_i, q^i_0, F_i)$ where $\delta_i \subseteq Q_i \times \Sigma \times Q_i$ is the transition relation. We denote by $\mathcal{L}(\mathcal{A}_i) \subseteq \Sigma^*$ the language of $\mathcal{A}_i$, which is defined as usual. The intersection problem asks whether there is a (nonempty) word $w \in \Sigma^+$ such that $w \in \mathcal{L}(\mathcal{A}_i)$ for all $i \in \{1, \dots, n\}$.

The bound B on the number of rounds will be the number of finite automata n. We construct a PFS that non-deterministically guesses a word $w \in \Sigma^+$ in the first round. Moreover, in round i, it will check that w is accepted by $\mathcal{A}_i$. To do this, each process will simulate one transition of $\mathcal{A}_i$ on one letter of w. That is, the number of processes is $|w|$. Each process will do exactly one action each round, and, to ensure that the word w is the same for each $\mathcal{A}_i$, will store in its local state which letter it represents. The global state stores the state of the currently simulated automaton. If we get a final state at the end of each round, it means that each $\mathcal{A}_i$ accepts the same word w.

Formally, we define $\mathcal{P} = (S, L, s_{\mathsf{in}}, \ell_{\mathsf{in}}, \Delta, F_{\mathsf{glob}}, F_{\mathsf{loc}})$ as follows:

- $S = \{s_{\mathsf{in}}\} \cup \Big(\big(\bigcup_{i \in \{1,\dots,n\}} Q_i \big) \times \{1, \dots, n\} \Big)$,
- $L = \{\ell_{\mathsf{in}}\} \cup (\Sigma \times \{0,1\} \times \{1, \dots, n\})$,
- $F_{\mathsf{glob}} = F_n \times \{n\}$, $F_{\mathsf{loc}} = \Sigma \times \{0,1\} \times \{n\}$, and

– $\Delta = \Delta_1 \cup \Delta_2 \cup \Delta_3 \cup \Delta_4$ where:

$$\Delta_1 = \{ \quad (s_{\mathsf{in}}, \ell_{\mathsf{in}}) \to ((q', 1), (a, 1, 1)) \quad | \quad (q_0^1, a, q') \in \delta_1 \quad \}$$

$$\Delta_2 = \{ \quad ((q, 1), \ell_{\mathsf{in}}) \to ((q', 1), (a, 0, 1)) \quad | \quad (q, a, q') \in \delta_1 \quad \}$$

$$\Delta_3 = \left\{ ((q_f, i), (a, 1, i)) \to ((q', i+1), (a, 1, i+1)) \;\middle|\; \begin{array}{l} q_f \in F_i \\ i \in \{1, \dots, n-1\} \\ (q_0^{i+1}, a, q') \in \delta_{i+1} \end{array} \right\}$$

$$\Delta_4 = \left\{ ((q, i), (a, 0, i-1)) \to ((q', i), (a, 0, i)) \;\middle|\; \begin{array}{l} i \in \{2, \dots, n\} \\ (q, a, q') \in \delta_i \end{array} \right\}$$

The only technical point here is that we need to differentiate between the first process and the others, because the first process has to initiate the next round. This is the meaning of the second component of the local state: the only process that will store 1 is the first process to play, all the others will store 0. Similarly, we also store the number of the round in the global state and each local state to ensure that every process plays exactly once in each round.

There are four kinds of transitions: Δ_1 and Δ_2 are the first transitions of respectively the first process and the non-first processes, while Δ_3 and Δ_4 are respectively the transitions of the first and non-first processes in the other rounds.

Lemma 4. *There is an accepting n-run of $\mathcal{P}$ if and only if* $\bigcap_{1 \le i \le n} \mathcal{L}(\mathcal{A}_i) \setminus \{\varepsilon\} \neq \emptyset$.

The rest of this section is devoted to the proof of the lemma.

$\boxed{\Rightarrow}$ Let $q_0^i \xrightarrow{a_1} \dots \xrightarrow{a_k} q_k^i$ be an accepting run of $\mathcal{A}_i$. Then we have the following run of $\mathcal{P}$: $(s_{\mathsf{in}}) \cdot \rho^1 \cdots \rho^n$ with

$$\rho^i = ((q_1^i, i)(a_1, 1, i)) \cdots ((q_k^i, i)(a_1, 1, i) \cdots (a_k, 0, i))$$

for all $1 \le i \le n$. It is easily verifiable from the definitions that it is a run, using the fact that $q_k^i \in F_i$ for all $1 \le i < n$. It is accepting because $q_k^n \in F_n$.

$\boxed{\Leftarrow}$ The structure of $\mathcal{P}$ impose that in an accepting run with k processes each process does exactly one transition in each round, and since we are restricted to round-bounded runs they also play in the correct order. Hence an accepting run with k processes is necessarily of the form $(s_{\mathsf{in}}) \cdot \rho^1 \cdots \rho^n$ with $\rho^i = ((q_1^i, i)(a_1, 1, i)) \cdots ((q_k^i, i)(a_1, 1, i) \cdots (a_k, 0, i))$ for all $1 \le i \le n$ given above with some $q_j^i \in Q_i$ and $a_j \in \Sigma$ for $1 \le i \le n$ and $1 \le j \le k$. Since transitions from Δ_3 are the only way to start a new round, we can also deduce that $q_k^i \in F_i$ for all $1 \le i \le n-1$, and since the run is accepting then also $q_k^n \in F_n$.

Let us pose $w = a_1 \dots a_k$. By definition of $\mathcal{P}$, for each $\mathcal{A}_i$ and $1 \le j \le k$, there is a transition $(q_{j-1}^i, a_j, q_j^i) \in \delta_i$. Thus for each $\mathcal{A}_i$ there is an accepting run $q_0^i \xrightarrow{a_1} \dots \xrightarrow{a_k} q_k^i$ on w.

D List of states and transitions of $\mathcal{G}_{\mathcal{M}}$

Here is the list of states and transitions used in $\mathcal{G}_{\mathcal{M}}$ that were left undefined. We will define three sets $\mathfrak{I}_?$, $\mathfrak{I}_{\mathrm{NP}}$, $\mathfrak{I}_{\mathrm{NR}}$ such that $\mathfrak{I} = \mathfrak{I}_? \uplus \mathfrak{I}_{\mathrm{NP}} \uplus \mathfrak{I}_{\mathrm{NR}}$.

First, for all $s \in S, \ell \in L, f_1, f_2 \in \{\checkmark, \times\}, j \in \{0,1\}, r \leq B$ we have $?(s, \ell, f_1, f_2, j, r) \in \mathfrak{I}_?$. For all such states, there is a transition

1. $?(s, \ell, f_1, f_2, j, r) \xrightarrow{\mathsf{int}} \mathrm{win}_{1-j}$

Moreover, for all $(s, \ell) \xrightarrow{(op,A)} (s', \ell')$ in Δ, there is

2. $?(s, \ell, f_1, f_2, j, r) \xrightarrow{op\ 1\ A} (s', \ell', f_1, f_2, pl(s'), r)$.

Finally if there exists at least one transition $(s, \ell) \xrightarrow{(op,A)} (s', \ell')$ in Δ, and $j \neq pl(s)$, then there is an additional transition

3. $?(s, \ell, f_1, f_2, j, r) \xrightarrow{\mathsf{int}} \mathrm{win}_j$ if $j \neq pl(s)$.

Then we have the set $\mathfrak{I}_{\mathrm{NP}}$ of intermediate states for a process change. For every $B \in \Gamma, s \in S, \ell \in L, f_2, g \in \{\checkmark, \times\}, j \in \{0,1\}, r \leq B$, we have $\mathrm{np}(s, f_2, j, r)$, $\mathrm{np}^B(s, f_2, j, r)$, and $\mathrm{np}^\ell(s, f_2, j, r)$ in $\mathfrak{I}_{\mathrm{NP}}$. The associated transitions are:

1. $(s, \ell, f_1, f_2, j, r) \xrightarrow{\mathsf{push}\ 2\ (\ell, f_2)} \mathrm{np}(s, \mathsf{upd}(f_2, \ell), j, r)$,
2. $\mathrm{np}(s, f_2, j, r) \xrightarrow{\mathsf{pop}\ 1\ B} \mathrm{np}^B(s, f_2, j, r)$,
3. $\mathrm{np}^B(s, f_2, j, r) \xrightarrow{\mathsf{push}\ 2\ B} \mathrm{np}(s, f_2, j, r)$,
4. $\mathrm{np}(s, f_2, j, r) \xrightarrow{\mathsf{pop}\ 1\ (\ell, \_)} \mathrm{np}^\ell(s, f_2, j, r)$,
5. $\mathrm{np}^\ell(s, f_2, j, r) \xrightarrow{\mathsf{push}\ 2\ (\ell, f_2)} \mathrm{np}(s, \mathsf{upd}(f_2, \ell), j, r)$,
6. $\mathrm{np}(s, f_2, j, r) \xrightarrow{\mathsf{pop}\ 1\ (\ell, g)} ?(s, \ell, g, f_2, j, r)$, and
7. $\mathrm{np}(s, f_2, j, r) \xrightarrow{\mathsf{zero}\ 1} ?(s, \ell_0, \checkmark, f_2, j, r)$.

where $\mathsf{upd}(f_2, \ell) = \checkmark$ iff $f_2 = \checkmark \wedge \ell \in F_{\mathsf{loc}}$.

And finally there is the set $\mathfrak{I}_{\mathrm{NR}}$ of intermediate states for a round change. For every $B \in \Gamma, s \in S, \ell \in L, f_1, g \in \{\checkmark, \times\}, j \in \{0,1\}, r \leq B$, we have $\mathrm{nr}(s, f_1, j, r)$, $\mathrm{nr}^B(s, f_1, j, r)$, and $\mathrm{nr}^\ell(s, f_1, j, r)$ in $\mathfrak{I}_{\mathrm{NR}}$., with the following transitions:

1. $(s, \ell, f_1, f_2, j, r) \xrightarrow{\mathsf{push}\ 2\ (\ell, f_1)} \mathrm{nr}(s, \mathsf{upd}(f_1, \ell), j, r)$,
2. $\mathrm{nr}(s, f_1, j, r) \xrightarrow{\mathsf{pop}\ 2\ B} \mathrm{nr}^B(s, f_1, j, r)$,
3. $\mathrm{nr}^B(s, f_1, j, r) \xrightarrow{\mathsf{push}\ 1\ B} \mathrm{nr}(s, f_1, j, r)$,
4. $\mathrm{nr}(s, f_1, j, r) \xrightarrow{\mathsf{pop}\ 2\ (\ell, \_)} \mathrm{nr}^\ell(s, f_1, j, r)$,
5. $\mathrm{nr}^\ell(s, f_1, j, r) \xrightarrow{\mathsf{push}\ 1\ (\ell, f_1)} \mathrm{nr}(s, \mathsf{upd}(f_1, \ell), j, r)$, and
6. $\mathrm{nr}(s, f_1, j, r) \xrightarrow{\mathsf{pop}\ 2\ (\ell, g)} ?(s, \ell, f_1, g, j, r + 1)$.

To simplify the proof of correctness, we assume that after a transition of type 6 the first stack becomes the active stack, forcing a phase change so that the phase number is always incremented by 2 after going in a round change gadget. This can be done for instance by using intermediate states and doing dummy push then pop transitions on stack 1.

E Proof of Lemma 2

By construction, every play of $\mathcal{G}_{\mathcal{P}}^{B}$ is closely mirrored by a play of the game $\mathcal{G}_{\mathcal{M}}$ we built (and vice-versa). Despite having more intermediate states in the gadgets, the possible plays in $\mathcal{G}_{\mathcal{M}}$ are restricted in a way such that basically the only thing a player can choose is a process and a transition to be executed by that process, which corresponds to what a player can do in $\mathcal{G}_{\mathcal{P}}^{B}$. Let us formalize this intuition by giving a mapping π from plays of $\mathcal{G}_{\mathcal{P}}^{B}$ to plays of $\mathcal{G}_{\mathcal{M}}$.

In the base game $\mathcal{G}_{\mathcal{P}}^{B}$, for all configurations c, c', round r and processes $p' < p$, there is a transition $(c, p, r) \to (c', p', r+1)$ iff there is a transition $(c, p', r+1) \to (c', p', r+1)$. Similarly, for all $p' > p$, there is a transition $(c, p, r) \to (c', p', r)$ iff there is a transition $(c, p', r) \to (c', p', r)$. In $\mathcal{G}_{\mathcal{M}}$, a transition from "$(c, p, r)$" to "$(c', p', r+1)$" will be simulated by a sequence of transitions corresponding to a "dummy transition" from "(c, p, r)" to "$(c, p', r+1)$" followed by an actual transition to "$(c', p', r+1)$". This will be similar for $p' > p$.

By abuse of notation, we say that a node v is in S_{sim} if $v = (s, \overline{\gamma_1}, \overline{\gamma_2}, st, ph)$ with $s \in S_{\text{sim}}$.

Let $v = ((s, \ell, f_1, f_2, j, r), \overline{\gamma_1}, \overline{\gamma_2}, 1, ph) \in S_{\text{sim}}$ be a node of $\mathcal{G}_{\mathcal{M}}$, with

$$\overline{\gamma_1} = \gamma_p \cdot (\ell_{p+1}, g_{p+1}) \cdot \gamma_{p+1} \cdots (\ell_k, g_k) \cdot \gamma_k$$

$$\overline{\gamma_2} = \tilde{\gamma}_{p-1} \cdot (\ell_{p-1}, g_{p-1}) \cdots \tilde{\gamma}_1 \cdot (\ell_1, g_1)$$

for some $\ell_1, \ldots, \ell_k \in L$, $\gamma_1, \ldots, \gamma_k \in \Gamma^*$, and $g_1, \ldots, g_k \in \{\checkmark, \text{✗}\}$, where $\tilde{\gamma}$ denotes the mirror of γ.

We say that v is *well-defined* if the following conditions are satisfied:

1. For all $1 \leq i \leq p-1$, $g_i = \checkmark$ if and only if for all $1 \leq p' < i$, $\ell_{p'} \in F_{\mathsf{loc}}$,
2. For all $p+1 \leq i \leq k$, $g_i = \checkmark$ if and only if for all $i < p' \leq k$, $\ell_{p'} \in F_{\mathsf{loc}}$,
3. $f_1 = \checkmark$ if and only if for all $p < p' \leq k$, $\ell_{p'} \in F_{\mathsf{loc}}$,
4. $f_2 = \checkmark$ if and only if for all $1 \leq p' < p$, $\ell_{p'} \in F_{\mathsf{loc}}$,
5. $j = pl(s)$, and
6. $ph = 2r$.

Note that every reachable $v \in S_{\text{sim}}$ in $\mathcal{G}_{\mathcal{M}}$ is either well-defined, or only fails item 5 because $j = 0 \neq 1 = pl(s)$ (in case of a "Change of Player" initiated by Player 1). We extend this definition to nodes of $\mathfrak{I}_?$ in the following way: $(?(s, \ell, f_1, f_2, j, r), \overline{\gamma_1}, \overline{\gamma_2}, 1, ph)$ is well-defined if $((s, \ell, f_1, f_2, j, r), \overline{\gamma_1}, \overline{\gamma_2}, 1, ph)$ is well-defined.

First, we give this mapping on individual nodes. We will extend it to plays after that. For a configuration $c = (s, (\ell_1, \gamma_1) \ldots (\ell_k, \gamma_k))$ and a process $p \in \{1, \ldots, k\}$, we define the following flags

$$g^{<}(c, p) = \checkmark \text{ iff } \ell_1, \ldots, \ell_{p-1} \in F_{\mathsf{loc}}$$
$$g^{>}(c, p) = \checkmark \text{ iff } \ell_{p+1}, \ldots, \ell_k \in F_{\mathsf{loc}}$$

and the following two stacks

$$\tau_1(c,p) = \gamma_p{\cdot}(\ell_{p+1}, g^{>}(c,p+1)) \cdot \gamma_{p+1} \cdots (\ell_k, g^{>}(c,k)) \cdot \gamma_k$$
$$\tau_2(c,p) = \quad \tilde{\gamma}_{p-1} \cdot (\ell_{p-1}, g^{<}(c,p-1)) \cdots \tilde{\gamma}_1 \cdot (\ell_1, g^{<}(c,1))$$

Let $c = (s, (\ell_1, \gamma_1) \dots (\ell_k, \gamma_k))$ and $u = (c, p, r)$. Its image is defined by

$$\pi(u) = ((s, \ell_p, g^{>}(c,p), g^{<}(c,p), pl(s), r), \tau_1(c,p), \tau_2(c,p), 1, 2r)$$

Observe that $\pi(u)$ is well-defined and is in S_{sim}.

Conversely, if $v \in S_{\text{sim}}$ is well-defined, then $v = ((s, \ell_p, f_1, f_2, pl(s), r), \overline{\gamma_1}, \overline{\gamma_2}, 1, 2r)$ with

$$\overline{\gamma_1} = \gamma_p \cdot (\ell_{p+1}, g_{p+1}) \cdot \gamma_{p+1} \cdots (\ell_k, g_k) \cdot \gamma_k$$

$$\overline{\gamma_2} = \tilde{\gamma}_{p-1} \cdot (\ell_{p-1}, g_{p-1}) \cdots \tilde{\gamma}_1 \cdot (\ell_1, g_1)$$

We then define $\hat{\pi}(v) = ((s, (\ell_1, \gamma_1) \dots (\ell_k, \gamma_k)), p, r)$. $\hat{\pi}$ is similarly defined on well-defined nodes in $\mathfrak{I}_?$. If $v \in S_{\text{sim}}$, then $\pi(\hat{\pi}(v)) = v$.

When $u' = (c', p', r')$ is a successor of $u = (c, p, r)$ in $\mathcal{G}^B_{\mathcal{P}}$, we do not necessarily have $\pi(u')$ successor of $\pi(u)$ in $\mathcal{G}_{\mathcal{M}}$, because of the mechanism of process or round change. We introduce a notation to describe the (unique) part of run that allows to go from $\pi(u)$ to $\pi(u')$.

Let $v = ((s, \ell, f_1, f_2, j, r), \overline{\gamma_1}, \overline{\gamma_2}, 1, ph) \in S_{\text{sim}}$ a node of $\mathcal{G}_{\mathcal{M}}$, with

$$\overline{\gamma_1} = \gamma_p \cdot (\ell_{p+1}, g_{p+1}) \cdot \gamma_{p+1} \cdots (\ell_k, g_k) \cdot \gamma_k$$

$$\overline{\gamma_2} = \tilde{\gamma}_{p-1} \cdot (\ell_{p-1}, g_{p-1}) \cdots \tilde{\gamma}_1 \cdot (\ell_1, g_1)$$

for some $\ell_1, \dots, \ell_k \in L$, $\gamma_1, \dots, \gamma_k \in \Gamma^*$, and $g_1, \dots, g_k \in \{\checkmark, \text{✗}\}$. Moreover, for $1 \leq i \leq k$, let $\gamma_i = B^i_1 \cdots B^i_{|\gamma_i|}$.

For every $p' \in \{1, \dots, p-1\} \cup \{p+1, \dots, k+1\}$, we define a run $\text{next}^{p'}_v$ that ends in a node $v_{p'} = (?(s, \ell_{p'}, f_1, f_2, j, r'), \overline{\gamma'_1}, \overline{\gamma'_2}, st, ph')$, with

$$\overline{\gamma'_1} = \gamma_{p'} \cdot (\ell_{p'+1}, g'_{p'+1}) \cdot \gamma_{p'+1} \cdots (\ell_k, g'_k) \cdot \gamma_k$$

$$\overline{\gamma'_2} = \tilde{\gamma}_{p'-1} \cdot (\ell_{p'-1}, g'_{p'-1}) \cdots \tilde{\gamma}_1 \cdot (\ell_1, g'_1).$$

and $r' = r$ if $p' > p$, $r' = r + 1$ otherwise.

Let $1 \leq i \leq k + 1 - p$, we define next^{p+i}_v. For that, we use the following notations: for $0 \leq x \leq i - 1$,

- for all $1 \leq n \leq |\gamma_{p+x}|$, $\overline{\gamma}^{(x)}_{1,n-1} = B^{p+x}_n \cdot \overline{\gamma}^{(x)}_{1,n}$ and $B^{p+x}_n \cdot \overline{\gamma}^{(x)}_{2,n-1} = \overline{\gamma}^{(x)}_{2,n}$ with $\overline{\gamma}^{(0)}_{1,0} = \overline{\gamma_1}$, $\overline{\gamma}^{(0)}_{2,0} = (\ell_p, f_2) \cdot \overline{\gamma_2}$, $\overline{\gamma}^{(x)}_{1,|\gamma_{p+x}|} = (\ell_{p+x+1}, g_{p+x+1}) \cdot \overline{\gamma}^{(x+1)}_{1,0}$ and $\overline{\gamma}^{(x+1)}_{2,0} = (\ell_{p+x+1}, g'^x) \cdot \overline{\gamma}^{(x)}_{2,|\gamma_{p+x}|}$
- $g'^x = \mathsf{upd}(g'^{x-1}, \ell_{p+x})$, with $g'^{-1} = f_2$,

Then, we can define next_v^{p+i}:

$$\text{next}_v^{p+i} = \Pi_{x=0}^{i-1}(\mathit{transfer}^{NP}_{\ell_{p+x}} \cdot \mathit{transfer}^{NP}_{\gamma_{p+x}}) \cdot (?(s, \ell_{p+i}, g_{p+i}, g'^{i-1}, j, r), \overline{\gamma}^{(i)}_{1,0}, \overline{\gamma_2}^{(i-1)}, 1, ph)$$

where

- $\mathit{transfer}^{NP}_{\ell_{p+x}} = \big(\text{np}^{\ell_{p+x}}(s, g'^{x-1}, j, r), \overline{\gamma}^{(x)}_{1,0}, \overline{\gamma_2}^{(x)}, 1, ph\big) \cdot \big(\text{np}(s, g'^{x}, j, r), \overline{\gamma}^{(x)}_{1,0}, \overline{\gamma}^{(x)}_{2,0}, 1, ph\big)$ for $x \geq 1$ with $\overline{\gamma_2}^{(x)} = \overline{\gamma}^{(x)}_{2,|\gamma_{p+x}|}$,
- $\mathit{transfer}^{NP}_{\ell_p} = \big(\text{np}(s, g'^{0}, j, r), \overline{\gamma}^{(0)}_{1,0}, \overline{\gamma}^{(0)}_{2,0}, 1, ph\big)$.
- $\mathit{transfer}^{NP}_{\gamma_{p+x}} = \Pi_{n=1}^{|\gamma_{p+x}|}\big(\text{np}^{B^{p+x}_n}(s, g'^{x}, j, r), \overline{\gamma}^{(x)}_{1,n}, \overline{\gamma}^{(x)}_{2,n-1}, 1, ph\big) \cdot \big(\text{np}(s, g'^{x}, j, r), \overline{\gamma}^{(x)}_{1,n}, \overline{\gamma}^{(x)}_{2,n}, 1, ph\big)$,
- $\ell_{k+1} = \ell_0$, $g_{k+1} = \checkmark$, and $\overline{\gamma}^{(k+1)}_{1,0} = \varepsilon$.

Let now $1 \leq i \leq p-1$, we define next_v^{p-i}, using similar notations: for $0 \leq x \leq i-1$,

- for all $1 \leq n \leq |\gamma_{p+x}|$, $\overline{\gamma'}^{(x)}_{2,n-1} = B^{p-x}_{|\gamma_{p-x}|-n+1} \cdot \overline{\gamma'}^{(x)}_{2,n}$ and $B^{p-x}_{|\gamma_{p-x}|-n+1} \cdot \overline{\gamma'}^{(x)}_{1,n-1} = \overline{\gamma'}^{(x)}_{1,n}$ with $\overline{\gamma'}^{(0)}_{1,0} = (\ell_p, f_1) \cdot \overline{\gamma_1}$, $\overline{\gamma'}^{(0)}_{2,0} = \overline{\gamma_2}$, $\overline{\gamma'}^{(x+1)}_{1,0} = (\ell_{p-x}, g^{x-1}) \cdot \overline{\gamma'}^{(x)}_{1,|\gamma_{p-x}|}$ and $\overline{\gamma'}^{(x)}_{2,|\gamma_{p-x}|} = (\ell_{p-x}, g_{p-x}) \cdot \overline{\gamma'}^{(x+1)}_{2,0}$
- $g^x = \mathsf{upd}(g^{x-1}, \ell_{p-x})$, with $g^{-1} = f_1$.

We then have

$$\text{next}_v^{p-i} = \mathit{transfer}^{NR}_{\ell_p} \cdot \Pi_{x=1}^{i-1}(\mathit{transfer}^{NR}_{\gamma_{p-x}} \cdot \mathit{transfer}^{NR}_{\ell_{p-x}}) \cdot \mathit{transfer}^{NR}_{\gamma_{p-i}} \cdot (?(s, \ell_{p-i}, g^{i-1}, g_{p-i}, j, r+1), \overline{\gamma'_1}^{(i)}, \overline{\gamma'}^{(i+1)}_{2,0}, 2, ph+2)$$

where

- $\mathit{transfer}^{NR}_{\ell_p} = \big(\text{nr}(s, g^0, j, r), \overline{\gamma'}^{(0)}_{1,0}, \overline{\gamma'}^{(0)}_{2,0}, 1, ph\big)$,
- $\mathit{transfer}^{NR}_{\gamma_{p-x}} = \Pi_{n=1}^{|\gamma_{p-x}|}\big(\text{nr}^{B^{p-x}_{|\gamma_{p-x}|-n+1}}(s, g^{x-1}, j, r), \overline{\gamma'}^{(x)}_{1,n-1}, \overline{\gamma'}^{(x)}_{2,n}, 2, ph+1\big) \cdot \big(\text{nr}(s, g^{x-1}, j, r), \overline{\gamma'}^{(x)}_{1,n}, \overline{\gamma'}^{(x)}_{2,n}, 2, ph+1\big)$,
- $\mathit{transfer}^{NR}_{\ell_{p-x}} = \big(\text{nr}^{\ell_{p-x}}(s, g^{x-1}, j, r), \overline{\gamma'_1}^{(x)}, \overline{\gamma'}^{(x)}_{2,0}, 2, ph+1\big) \cdot \big(\text{nr}(s, g^{x}, j, r), \overline{\gamma'}^{(x)}_{1,0}, \overline{\gamma'}^{(x)}_{2,0}, 2, ph+1\big)$ for $x \geq 1$ with $\overline{\gamma'_1}^{(x)} = \overline{\gamma'}^{(x)}_{1,|\gamma_{p-x}|}$.

By convention, we let $\text{next}_v^{p} = \varepsilon$.

Lemma 5. *Let $\hat{\rho}$ be a finite play of $\mathcal{G}_\mathcal{M}$ ending in $v = \pi(u)$ for some u. If $\hat{\rho} \cdot \hat{\rho}' \cdot v'$ is a play where v' is in S_{sim} and $\hat{\rho}'$ is a partial play (or ε) with no nodes in S_{sim}, then there is a unique p' such that $\hat{\rho}'$ is of the form $\text{next}_v^{p'}$ and a unique u' successor of u such that $v' = \pi(u')$.*

Let $c = ((s, (\ell_1, \gamma_1) \dots (\ell_k, \gamma_k))$ and $u = (c, p, r)$ and $v = ((s, \ell, f_1, f_2, j, r), \overline{\gamma_1}, \overline{\gamma_2}, 1, ph)$. There are three possible ways to go from v to v': either directly with a basic transition, by doing a process change, or by doing a round change.

In the case of a basic transition, this corresponds to $p' = p$ (so $\text{next}_v^{p'} = \varepsilon$), and by definition of a basic transition there is a transition $\delta = (s, \ell) \xrightarrow{op\ A} (s', \ell') \in \Delta$ with $v' = ((s', \ell', f_1, f_2, pl(s'), r), \overline{\gamma_1}', \overline{\gamma_2}, 1, ph) = \pi(u')$ where u' is the successor of u reached by transition δ.

If $\hat{\rho}' \neq \varepsilon$, then the last node of $\hat{\rho}'$ is in $\mathfrak{I}_?$ because a node in S_{sim} can only be reached from another node in S_{sim} or a node in $\mathfrak{I}_?$, and ρ' does not have any node in S_{sim} by hypothesis. For any node of the form $((\text{np}(s, f_2, j, r)), \overline{\gamma_1}, \overline{\gamma_2}, 1, ph)$, the available successors depend only on the top of the first stack $\overline{\gamma_1}$. If there are stack letters from Γ on the top, then only a state of the form $\text{np}^B(s, f_2, j, r)$ can be reached, from which the only successor is again $\text{np}(s, f_2, j, r)$. Therefore if γ is on top of $\overline{\gamma_1}$, then all plays will start by $\mathit{transfer}_\gamma^{NP}$. If there is a local state (ℓ, f_1) on the top, there are two available successors: $\text{np}^\ell(s, f_2, j, r)$ and $?(s, \ell, f_1, f_2, j, r)$. The second case leaves $\mathfrak{I}_{\text{NP}}$ so it will only occurs once per process change, but the first case loops back to $\text{np}(s, f_2', j, r)$ so it may happen multiple times (this corresponds to a $\mathit{transfer}_\ell^{NP}$). Finally if the first stack is empty, then only the successor $?(s, \ell_{\mathsf{in}}, \checkmark, f_2, j, r)$ is available, so for the same reason as above this can only occur once per process change.

If $\hat{\rho}'$ goes in $\mathfrak{I}_{\text{NP}}$, then by the remarks above it can be decomposed into $\hat{\rho}' = \hat{\rho}'_0 \cdots \hat{\rho}'_{m-1} \cdot v_?$ such that $v_? \in \mathfrak{I}_?$ and for all $0 \leq i \leq m-1$ the last node of ρ'_i is of the form $((\text{np}(s, f_2^i, j, r)), \overline{\gamma_1^i}, \overline{\gamma_2^i}, 1, ph)$ with a local state on top of $\overline{\gamma_1^i}$. Moreover, as $\overline{\gamma_1} = \gamma_p \cdot (\ell_{p+1}, g_{p+1}) \cdot \gamma_{p+1} \cdots \gamma k$, each ρ'_i is of the form $\mathit{transfer}_{\ell_{p+i}}^{NP} \cdot \mathit{transfer}_{\gamma_{p+i}}^{NP}$. Thus $\rho' = \text{next}_v^{p+m}$. The proof is similar for a round change, except the roles of $\overline{\gamma_1}$ and $\overline{\gamma_2}$ are reversed.

Finally, we have an edge between $v_?$ and v, there is an edge between $\hat{\pi}(v_?)$ and $u' = \hat{\pi}(v)$. As $\hat{\pi}(v_?) = (c, p', r')$ and $\hat{\pi}(v) = (c', p', r')$ for some configuration c', then there is an edge between $u = (c, p, r)$ and u'.

Now we extend the definition of π on plays. We define

$$\pi(((s_{\mathsf{in}}), 0, 1) = (s'_{\mathsf{in}}, \varepsilon, \varepsilon, 1, 1)$$

and for all transitions $(s_{\mathsf{in}}, \ell_{\mathsf{in}}) \xrightarrow{op\ A} (s, \ell)$ in Δ, we have

$$\pi(((s_{\mathsf{in}}), 0, 1) \cdot ((s, (\ell, \gamma_1)), 1, 1)) = (s'_{\mathsf{in}}, \varepsilon, \varepsilon, 1, 1) \cdot ((s, \ell, \checkmark, \checkmark, pl(s), 1), \gamma_1, \varepsilon, 1, 1)$$

Let ρ be a finite play of $\mathcal{G}_{\mathcal{P}}^B$ ending in $u = (c, p, r)$, with $p \neq 0$, and $u' = (c', p', r')$ a successor of u.

$$\pi(\rho \cdot u') = \begin{cases} \pi(\rho) \text{ if } \rho \text{ is winning, else} \\ \pi(\rho) \cdot \text{next}_{\pi(u)}^{p'} \cdot \pi(u') \cdot (\text{win}_0, \overline{\gamma_1}', \overline{\gamma_2}', 1, ph')^\omega \text{ if } u' \in \mathcal{F}, \text{ else} \\ \pi(\rho) \cdot \text{next}_{\pi(u)}^{p'} \cdot \pi(u') \cdot \text{Ch}_0 \cdot \text{next}_{\pi(u')}^{p'+1} \cdot (\text{win}_1, \overline{\gamma_1}', \overline{\gamma_2}', 1, ph')^\omega \text{ if } u' \text{ has no successor, else} \\ \pi(\rho) \cdot \text{next}_{\pi(u)}^{p'} \cdot \pi(u'). \end{cases}$$

where $\pi(u') = ((s', \ell', f_1', f_2', j', r'), \overline{\gamma_1}', \overline{\gamma_2}', 1, ph')$ and $\text{Ch}_0 = ((s', \ell', f_1', f_2', 0, r'), \overline{\gamma_1}', \overline{\gamma_2}', 1, ph')$ if $j' = 1$, ε otherwise. Assume that $\pi(\rho)$ is a finite play of $\mathcal{G}_{\mathcal{M}}$. By construction,

$\pi(\rho) \cdot \text{next}^{p'}_{\pi(u)} \cdot \pi(u')$ is a play of $\mathcal{G}_\mathcal{M}$. If $u' \in \mathcal{F}$, then since $\pi(u')$ is well-defined, $\pi(u') \in \mathfrak{F}$ and $\pi(\rho \cdot u')$ is a play of $\mathcal{G}_\mathcal{M}$. Otherwise, since it is always possible to go to a state win_1 from any $((s', \ell', f'_1, f'_2, 0, r'), \overline{\gamma_1}', \overline{\gamma_2}', 1, ph')$, $\pi(\rho \cdot u')$ is also a play of $\mathcal{G}_\mathcal{M}$.

If ρ is an infinite play, we let $\pi(\rho) = \lim_{\rho' \sqsubseteq \rho} \pi(\rho')$.

Conversely, for all plays $\hat{\rho}'$ and nodes v in $\mathcal{G}_\mathcal{M}$, we define

$$\hat{\pi}(\hat{\rho} \cdot v) = \begin{cases} \hat{\pi}(\hat{\rho}) \cdot \hat{\pi}(v) \text{ if } v \text{ is a well-defined node in } S_{\text{sim}} \\ \hat{\pi}(\hat{\rho}) \text{ otherwise.} \end{cases}$$

Let $f_\mathcal{M}$ be a winning strategy of $\mathcal{G}_\mathcal{M}$ and ρ be a finite play of $\mathcal{G}^B_\mathcal{P}$ ending in $u \in V_0$. If $\pi(\rho)$ is a finite $f_\mathcal{M}$-play, then we show that there is a unique $u'(\rho, f_\mathcal{M})$ successor of u such that $\pi(\rho \cdot u'(\rho, f_\mathcal{M}))$ is also a $f_\mathcal{M}$-play. First observe that if $\pi(\rho)$ is finite, it ends in $\pi(u)$.

If $f_\mathcal{M}(\pi(\rho)) = v$ with v in S_{sim}, then $u'(\rho, f_\mathcal{M}) = \hat{\pi}(v)$ and obviously, $\pi(\rho \cdot u'(\rho, f_\mathcal{M}))$ is a $f_\mathcal{M}$-play. Since in that case, $\pi(\rho \cdot u'(\rho, f_\mathcal{M})) = \pi(\rho) \cdot v$, v is a successor of $\pi(u)$ in $\mathcal{G}_\mathcal{M}$, and by Lemma 5, $\hat{\pi}(v)$ is a successor of u. Hence $\rho \cdot u'(\rho, f_\mathcal{M})$ is a play of $\mathcal{G}^B_\mathcal{P}$. Otherwise, $v \in \mathfrak{I}$. Indeed, if it is win_1, $f_\mathcal{M}$ cannot be winning, and it cannot be win_0, since $\pi(\rho)$ is finite, $u \notin \mathcal{F}$, hence, no successor of u can have its state in win_0. Assume now that there is a $f_\mathcal{M}$-play starting in $\pi(\rho) \cdot v$ that never visits again a node whose state is in S_{sim}. Then this run would stay forever in $\mathfrak{I}$ or would end in win_1, which is impossible since $f_\mathcal{M}$ is winning. Then any $f_\mathcal{M}$-run starting in $\pi(\rho) \cdot v$ is of the form $\pi(\rho) \cdot \hat{\rho}' \cdot v' \cdot \hat{\rho}''$, with $v' = ((s', \ell', f'_1, f'_2, j', r'), \overline{\gamma_1}', \overline{\gamma_2}', 1, ph')$. By Lemma 5, $\hat{\rho}'$ is of the form $\text{next}^{p'}_{\pi(u)}$ and $\hat{\pi}(v')$ is a successor of u. As every node of ρ' belongs to Player 0, this prefix is the same for every maximal extension of $\pi(\rho)$, thus we define $u'(\rho, f_\mathcal{M}) = \hat{\pi}(v')$.

We build the following strategy $f_\mathcal{P}$ for Player 0 in $\mathcal{G}^B_\mathcal{P}$: if ρ is a play of $\mathcal{G}^B_\mathcal{P}$ ending in a node $u \in V_0$, then

$$f_\mathcal{P}(\rho) = \begin{cases} u'(\rho, f_\mathcal{M}) \text{ if } \pi(\rho) \text{ is a finite } f_\mathcal{M}\text{-play,} \\ \text{any successor of } u \text{ otherwise.} \end{cases}$$

By induction, if ρ is a $f_\mathcal{P}$-play, then $\pi(\rho)$ is a $f_\mathcal{M}$-play. Assume there is a maximal $f_\mathcal{P}$-play ρ that is not winning. Then if ρ is finite, $\hat{\pi}(\rho)$ cannot be winning, and since $\pi(\rho)$ is a $f_\mathcal{M}$-run, it is impossible. If ρ is infinite, by construction, $\hat{\pi}(\rho)$ never visits a node whose state is in win_0, hence again $\pi(\rho)$ is not winning, which is impossible. Hence, $f_\mathcal{P}$ is a winning strategy in $\mathcal{G}^B_\mathcal{P}$.

Let $f_\mathcal{P}$ be a winning strategy of $\mathcal{G}^B_\mathcal{P}$ and $\hat{\rho}$ a play of $\mathcal{G}_\mathcal{M}$ ending in a node v of Player 0. Suppose that $\hat{\pi}(\hat{\rho})$ is a finite $f_\mathcal{P}$-play ending in u. If $\hat{\pi}(\hat{\rho})$ is not winning, u has at least one successor because $f_\mathcal{P}$ is a winning strategy.

Let $\hat{\rho}'$ be the smallest prefix of $\hat{\rho}$ whose last node belongs to Player 0 such that $\hat{\pi}(\hat{\rho}) = \hat{\pi}(\hat{\rho}')$ (which means its last node is in S_{sim}). If u has at least a successor, let $u' = f_\mathcal{P}(\hat{\pi}(\hat{\rho}))$ if $u \in V_0$ and some successor of u if $u \in V_1$. By construction, $\hat{\pi}(\hat{\rho}) \cdot u'$ is a $f_\mathcal{P}$-play, thus there is a play $\hat{\rho} \cdot \text{next}^{p'}_{\pi(u)} \cdot \pi(u')$ in $\mathcal{G}^B_\mathcal{P}$ where $u' = (c', p', r')$. Let $\text{next}^{p'}_{\pi(u)} = v_1 \dots v_m$.

We define $f_{\mathcal{M}}$ as follows:

$$\begin{cases} f_{\mathcal{M}}(\hat{\rho}) = \text{ any successor of } v \text{ is } u \text{ has no successor, else} \\ f_{\mathcal{M}}(\hat{\rho}' \cdot v_1 \cdots v_k) = v_{k+1} \text{ for } 0 \leq k < m, \\ f_{\mathcal{M}}(\hat{\rho}' \cdot v_1 \cdots v_m) = \pi(u'). \end{cases}$$

By induction, if $\hat{\rho}$ is a $f_{\mathcal{M}}$-play, then $\hat{\pi}(\hat{\rho})$ is a $f_{\mathcal{P}}$-play. Suppose there is a $f_{\mathcal{M}}$-play $\hat{\rho}$ that is maximal (i.e infinite) and not winning. That means either at some point $\hat{\rho}$ reached win_1 which is a sink state, or $\hat{\rho}$ visits S_{sim} infinitely often.

The only way the first case can happen is if the last node of $\hat{\pi}(\hat{\rho})$ has no successor. Since $\hat{\rho}$ is not winning, so is $\hat{\pi}(\hat{\rho})$. Hence we have a $f_{\mathcal{P}}$-play that is maximal and not winning, which is a contradiction.

In the second case, this means that $\hat{\pi}(\hat{\rho})$ is also infinite. Furthermore, $\hat{\pi}(\hat{\rho})$ does not visit a final configuration, otherwise $\hat{\rho}$ would end in win_0. Again, this means that $\hat{\pi}(\hat{\rho})$ is a $f_{\mathcal{P}}$-play that is maximal and not winning, thus we have a contradiction. Hence, $f_{\mathcal{M}}$ is a winning strategy.

F Proof Details for Lower Bound of Theorem 4

Formally let φ be a formula, $\text{Cl}(\varphi)$ the set of subformulas (non-strict) of φ, and $\text{Var}_\varphi \subset \text{Var}$ the set of variables appearing in φ. We define $B = |\text{Var}_\varphi| + 2$ and $\mathcal{P} = (S_0 \uplus S_1, L, s_{\mathsf{in}}, \ell_{\mathsf{in}}, \Delta, F_{\mathsf{glob}}, F_{\mathsf{loc}})$ as follows:

- $S = \{s_{\mathsf{in}}, \text{Guess}, \text{Win}\} \cup \{\text{Win-if-}x \mid x \in \text{Var}_\varphi\} \cup \{\psi \mid \psi \in \text{Cl}(\varphi)\} \cup \{\boxed{\psi} \mid \psi \in \text{Cl}(\varphi)\}$
  A state is in S_1 if it is of the form $\psi \wedge \psi'$ or $\forall x.\psi$, otherwise it is in S_0.
  The initial state is s_{in}, and $F_{\mathsf{glob}} = \{\text{Win}\}$.
- $L = \{\ell_{\mathsf{in}}, \text{first}\} \cup \left(\Sigma \times 2^{\text{Var}_\varphi}\right)$, with initial state ℓ_{in} and $F_{\mathsf{loc}} = L$.
- Δ contains the following transitions:
  1. $(s_{\mathsf{in}}, \ell_{\mathsf{in}}) \to (\text{Guess}, \text{first})$
  2. $(\text{Guess}, \ell_{\mathsf{in}}) \to (\text{Guess}, (a, \emptyset))$
  3. $(\text{Guess}, \text{first}) \to (\varphi, \text{first})$
  4. $(\boxed{\psi}, \text{first}) \to (\psi, \text{first})$
  5. $(?x.\psi, (a, S)) \to (\boxed{\psi}, (a, S \cup \{x\}))$ for $? \in \{\exists, \forall\}$
  6. $(\psi_1 ? \psi_2, \text{first}) \to (\psi_i, \text{first})$ for $? \in \{\vee, \wedge\}$ and $i \in \{1, 2\}$
  7. $(a(x), (a, S)) \to (\text{Win}, (a, S))$ if $x \in S$
  8. $(\neg a(x), (b, S)) \to (\text{Win}, (b, S))$ if $x \in S$ and $b \neq a$
  9. $(x < y, (a, S)) \to (\text{Win-if-}y, (a, \emptyset))$ if $x \in S$
  10. $(\neg(x < y), (a, S)) \to (\text{Win-if-}x, (a, S))$ if $y \in S$
  11. $(\text{Win-if-}x, (a, S)) \to (\text{Win}, (a, S))$ if $x \in S$

Lemma 6. *There is a winning strategy for Player 0 in $\mathcal{P}$ if and only if φ is satisfiable.*

Proof. Given a configuration $c = (s, \text{first}, (a_1, S_1), \dots, (a_n, S_n))$ of size $n+1$, we define the associated (partial) valuation $\nu(c)(x) = i$ if $x \in S_i$, which is well defined as there is no possible way in the game to have a single variable x in S_i and S_j if $i \neq j$. Conversely, given a state s, a word $w = a_1 \dots a_n$ and a valuation ν, the associated configuration is $c(s, w, \nu) = (s, \text{first}, (a, S_1), \dots, (a, S_n))$ where $S_i = \{x \mid \nu(x) = i\}$.

Let f be a winning strategy for Player 0. From the initial node (which belongs to Player 0), f will necessarily first do one transition of type 1, then n transitions of type 2 (for $n \in \mathbb{N}$), then one transition of type 3, reaching a node of the form $v_\varphi = ((\varphi, \text{first}, (a_1, \emptyset), \dots, (a_n, \emptyset)), 1, 2)$. We fix $w_f = a_1 \dots a_n$. Let f' be a strategy for Player 1, and ρ the winning $(f, f')-$play. We show by recursion on the subformula ψ that for all nodes $v_\psi = (c_\psi, 1, r)$ with $c_\psi = (\psi, \text{first}, (a_1, S_1), \dots, (a_n, S_n))$ visited during ρ, we have $w_f, \nu(c_\psi) \models \psi$.

First, note that if ψ is a term (or negated term), then necessarily v_ψ is reached during the last round $r = B = |\text{Var}_\varphi| + 2$ as it takes one round to reach v_φ and then $|\text{Var}_\varphi|$ rounds to go through every quantifier of φ.

- If $\psi = a(x)$, then as ρ is winning there is a process with local state (a_i, S_i) with $a_i = a$ and $x \in S_i$ (transition of type 7), in other words $\nu(c_\psi)(x) = i$, so we have $w_f, \nu(c_\psi) \models \psi$.
- If $\psi = x < y$, let $v_1 = ((\text{Win-if-}y, \dots, (a_i, \emptyset), \dots), i+1, B)$ and $v_2 = ((\text{Win}, \dots), j+1, B)$ such that ρ ends in $v_\psi v_1 v_2$ (transitions 9 then 11). Then we have $\nu(c_\psi)(x) = i$ and $\nu(c_\psi)(y) = j$. Since v_1 and v_2 are visited in the same round (see note above), then $i \leq j$. And since after v_1 $S_i = \emptyset$, we know that $i \neq j$, thus $i < j$.
- Similarly for $\psi = \neg(x < y)$, we have $\nu(c_\psi)(x) = i$ and $\nu(c_\psi)(y) = j$ with $j \leq i$ but this time no strict inequality.
- If $\psi = \psi_1 \vee \psi_2$, then let $i \in \{1, 2\}$ such that the next node has global state ψ_i (transition 6), as we know recursively that $w_f, \nu(c_{\psi_i}) \models \psi_i$ and that $\nu(c_{\psi_i}) = \nu(c_\psi)$ (as no local state is changed during the transition), then $w_f, \nu(c_\psi) \models \psi_i$, which in turn means that $w_f, \nu(c_\psi) \models \psi$.
- Similarly if $\psi = \psi_1 \wedge \psi_2$, for every $i \in \{1, 2\}$ representing the choice of Player 1 we have $w_f, \nu(c_{\psi_i}) \models \psi_i$ and $\nu(c_{\psi_i}) = \nu(c_\psi)$, thus $w_f, \nu(c_\psi) \models \psi$.
- If $\psi = \exists x.\psi'$, let $v_1 = ((\boxed{\psi'}, \dots), i+1, r)$ with $1 \leq i$ the successor node of v_ψ in ρ, and $v_2 = ((\psi', \dots), 1, r+1)$ the successor of v_1 (transitions 5 then 4). By recursion we have that $w_f, \nu(c_{\psi'}) \models \psi'$, and $\nu(c_{\psi'}) = \nu(c_\psi) \uplus \{x \to i\}$ by construction. Thus $w_f, \nu(c_\psi) \models \psi$.
- If $\psi = \forall x.\psi'$, for all $1 \leq i \leq n$ we let $v_1^i = ((\boxed{\psi'}, \dots), i+1, r)$ and $v_2^i = (c_{\psi'}^i, 1, r+1)$ the i possible successors of v_ψ corresponding to Player 1's choice. Similarly, we know that for all i $w_f, \nu(c_{\psi'}^i) \models \psi'$ and $\nu(c_{\psi'}^i) = \nu(c_\psi) \uplus \{x \to i\}$. Thus $w_f, \nu(c_\psi) \models \psi$.

From this we conclude that $w_f, \nu(c_\varphi) \models \varphi$ and as $\nu(c_\varphi)$ is the empty valuation, then w_f satisfies φ.

Now suppose that φ is satisfied by $w = a_1 \dots a_n$. Let ρ be a run ending in v. We build f_φ as follows:

- If $v = ((\text{Guess}, \text{first}, (a_1, \emptyset), \dots, (a_k, \emptyset)), k+1, 1)$ for $0 \leq k < n$ then $f_\varphi(\rho) = ((\text{Guess}, \text{first}, (a_1, \emptyset) \dots, (a_k, \emptyset), (a_{k+1}, \emptyset)), k+2, 1)$. If $k = n$, then $f_\varphi(\rho) = ((\varphi, \text{first}, (a_1, \emptyset), \dots, (a_k, \emptyset)), 1, 2)$
- If $v = (c_\psi, 1, r)$ with $c_\psi = (\exists x.\psi', \text{first}, (a_1, S_1), \dots, (a_n, S_n))$, then assuming that $w, \nu(c_\psi) \models \psi$ we pick one $i \leq n$ such that $w, \nu(c_\psi) \uplus \{x \to i\} \models \psi'$ and we define $f_\varphi(\rho) = ((\boxed{\psi'}, (a_1, S'_1), \dots, (a_n, S'_n)), i+1, r)$ with $S'_i = S_i \uplus \{x\}$ and $S'_j = S_j$ for $j \neq i$.
- If $v = (c_\psi, 1, r)$ with $c_\psi = (\psi_1 \vee \psi_2, \dots)$, then assuming $w, \nu(c_\psi) \models \psi$ we know that there is at least one $i \in \{1, 2\}$ such that $w, \nu(c_\psi) \models \psi_i$. We pick one such i, and define $f_\varphi(\rho) = ((\psi_i, \dots), 1, r)$.
- For all other cases, there is at most one transition available so f_φ is defined unambiguously.

Let f' be a strategy for Player 1, and $\rho = v_0 \dots v_m$ the resulting (f_φ, f')-play. We show that ρ is winning.

By definition of f_φ and since v_0 to v_n are owned by Player 0, we have that $v_{n+1} = (c(\varphi, w, \emptyset), 1, 2)$. Let $k \in \{n+1, n+3, \dots, n+2 \cdot (|\text{Var}_\varphi| - 1)\}$. We have the following two properties:

1. If $v_k = (c(\exists x.\psi', w, \nu), 1, r)$ such that $w, \nu \models \exists x.\psi'$, then by construction of f_φ we have $v_{k+1} = (c(\boxed{\psi'}, w, \nu'), i+1, r)$ and $v_{k+2} = (c(\psi', w, \nu'), 1, r+1)$ for some $1 \leq i \leq n$ and $\nu' = \nu \uplus \{x \to i\}$, and furthermore $w, \nu' \models \psi'$.
2. If $v_k = (c(\forall x.\psi', w, \nu), 1, r)$ such that $w, \nu \models \forall x.\psi'$, then for some $1 \leq i \leq n$ (defined by f') we have $\nu' = \nu \uplus \{x \to i\}$ such that $v_{k+1} = (c(\boxed{\psi'}, w, \nu'), i+1, r)$ and $v_{k+2} = (c(\psi', w, \nu'), 1, r+1)$. By definition since $w, \nu \models \forall x.\psi'$, we deduce that $w, \nu' \models \psi'$.

By those two properties, combined with the fact that $v_{n+1} = (c(\varphi, w, \emptyset), 1, 2)$, we deduce that $v_{n+2\cdot|\text{Var}_\varphi|} = (c(\psi', w, \nu), 1, B)$ where ψ' has no quantifiers and $w, \nu \models \psi'$.

Let $k \geq n + 2 \cdot |\text{Var}_\varphi|$, again we have two similar-looking properties:

1. If $v_k = (c(\psi_1 \vee \psi_2, w, \nu), 1, B)$ and $w, \nu \models \psi_1 \vee \psi_2$ then by definition of f_φ, $v_{k+1} = (c(\psi_i, w, \nu), 1, B)$ with $i \in \{1, 2\}$ and $w, \nu \models \psi_i$.
2. If $v_k = (c(\psi_1 \wedge \psi_2, w, \nu), 1, B)$ and $w, \nu \models \psi_1 \wedge \psi_2$ then for some $i \in \{1, 2\}$ defined by f', $v_{k+1} = (c(\psi_i, w, \nu), 1, B)$. By definition of satisfiability, we also have that $w, \nu \models \psi_i$.

Using those two properties, we deduce that there exists $m' \geq n + 2 \cdot |\text{Var}_\varphi|$ such that $v_{m'} = (c(t, w, \nu), 1, B)$ where t is a term or a negated term such that $w, \nu \models t$. Here m' depends not only on φ but also on both strategies f_φ and f'. There are 4 possible cases for t:

1. $t = a(x)$: as $w, \nu \models t$ we know that $a_{\nu(x)} = a$, and $v_{m'} = (c(t, w, \nu), 1, B)$ so $v_{m'+1} = (c(\text{Win}, w, \nu), \nu(x) + 1, B)$.

2. $t = \neg a(x)$: similarly, we have $v_{m'+1} = (c(\text{Win}, w, \nu), \nu(x) + 1, B)$.
3. $t = x < y$: we know that $\nu(x) < \nu(y)$ so $v_{m'+1} = (c(\text{Win-if-}y, w, \nu'), \nu(x), B)$ where $\nu' = \nu - \{\{x' \to \nu(x)\} \mid x' \in \text{Var}_\varphi\}$. Since $\nu(x) \neq \nu(y)$, $\nu'(y) = \nu(y) > \nu(x)$. So $v_{m'+2} = (c(\text{Win}, w, \nu'), \nu(y), B)$.
4. $t = \neg(x < y)$: in this case $\nu(y) \leq \nu(x)$ and we have $v_{m'+1} = (c(\text{Win-if-}x, w, \nu), \nu(y), B)$ and $v_{m'+2} = (c(\text{Win}, w, \nu), \nu(x), B)$.

Every case ends in an accepting node, therefore ρ is winning.