Formal Methods in System Design
https://doi.org/10.1007/s10703-023-00431-0

ORIGINAL ARTICLE


# Round- and context-bounded control of dynamic pushdown systems

**Benedikt Bollig**[1] · **Mathieu Lehaut**[2] · **Nathalie Sznajder**[2]

Received: 12 October 2022 / Accepted: 5 June 2023


**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.

**Keywords** Parameterized systems · Control · Underapproximation · Pushdown systems · Games

## 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 [20] for an overview. Furthermore, as the number of processes involved in the system is unbounded, one can distinguish two cases: either for each execution there is a finite (albeit unknown) set of processes involved (*parameterized* case), or potentially infinitely many processes can be generated during the execution (*dynamic* case). In this paper, we focus on the latter.

✉ Mathieu Lehaut
mathieu.lehaut@gmail.com

Benedikt Bollig
bollig@lsv.fr

Nathalie Sznajder
nathalie.sznajder@lip6.fr

[1] CNRS, LMF & ENS Paris-Saclay, Université Paris-Saclay, Gif-sur-Yvette, France

[2] LIP6, CNRS, Sorbonne Université, 75005 Paris, France

Published online: 07 July 2023

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 [29], but defined in a way that reveals similarities with data automata/class-memory automata, a certain automata model over infinite alphabets [8, 9]. More precisely, we consider *dynamic pushdown systems*, as each process has a dedicated stack to model recursion. A dynamic pushdown system distinguishes between a finite-state *global process* (sometimes 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 verification of round-bounded parameterized systems, which restricts system executions to a bounded number of round-robin schedules while still allowing an unbounded number of processes to be involved in the execution [29]. 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 dynamic 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, 36]. 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 [28], too, to solve verification problems involving queues. Finally, we also prove that relaxing the round-robin schedule restriction by allowing processes to act in a different order in every round makes the problem undecidable again.

*Related Work* Underapproximate verification goes back to Qadeer and Rehof [34]. 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.

As already mentioned, there is a large body of literature on parameterized verification, mostly focusing on closed systems (e.g., [2, 4, 19, 20]). As said earlier, our model is similar to the one from [29], which is also essentially the one from [12]. Contrary to the latter, we do not model local transitions explicitly. They correspond to transitions that do not change the

global state. In [12], context-bounded verification is studied. In a context, only one process can access the global state, but any other process can execute local transitions. Then, the number of context switches of the whole system is bounded. In [6], it is argued that this restriction is ill-suited for parameterized systems, and they change it to a per-process basis. We use a slightly different but closely related definition for contexts in Sect. 5 to show an undecidability result in the game-based setting.

Infinite-state games have been extensively studied over vector addition systems with states (VASS) (e.g., [3, 7, 13, 15, 24]). 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 [22].

We believe that our results will fertilize *synthesis* of parameterized systems [23] and more classical questions whose theoretical foundations go back to the 50 s and Church's synthesis problem. Let us cite Brütsch and Thomas, who observed a lack of approaches to synthesis over infinite alphabets [14]: "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 $\omega$-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 also inspired by data automata/class-memory automata [8, 9], which were originally defined as language acceptors over infinite alphabets. The automata studied in [14] are quite different. The synthesis problem over infinite alphabets has also been studied in [17, 21, 26].

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.

This work is an extended version of [10] and [11], adding the result of Sect. 5 to the previous works and improving the readability of proofs in Sect. 4.

*Outline* We define parameterized pushdown systems in Sect. 2, where we also recall known results on reachability questions. Rounds are defined in Sect. 3, and we discuss round-bounded reachability problems. The control problem is addressed in Sect. 4, and we discuss a possible relaxation of our restriction in Sect. 5. Finally we conclude in Sect. 6.

## 2 Dynamic pushdown systems

We start with some preliminary definitions. For $n \in \mathbb{N}$, we let $[n] = \{1, \ldots, n\}$.

*Words* Let $\Sigma$ be a (possibly infinite) set. A *word* $w$ over $\Sigma$ is a finite or (countably) infinite sequence $a_0 a_1 a_2 \ldots$ of elements $a_i \in \Sigma$. Let $\Sigma^*$ denote the set of finite words over $\Sigma$, $\Sigma^\omega$ 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 $\varepsilon$ 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 may also write $u \to v$. We call $v$ a *successor* of $u$.

A *partial run* of $\mathcal{T}$ is a non-empty, finite or infinite sequence $\rho = v_0 v_1 v_2 \ldots \in V^\infty$ 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 $\rho$ a *run*. A (partial) run from $u$ to $v$ is a finite (partial) run of the form $v_0 v_1 \ldots v_n$ with

$v_0 = u$ and $v_n = v$. In particular, $u$ is a partial run (of length 1) from $u$ to $u$. By $Runs(\mathcal{T})$, we denote the set of runs of $\mathcal{T}$.

A set of nodes $\mathcal{F} \subseteq V$, interpreted as a reachability condition, induces the set of *accepting runs*

$$Acc(\mathcal{T}, \mathcal{F}) = \{\rho = v_0 v_1 \ldots \in Runs(\mathcal{T}) \mid v_i \in \mathcal{F} \text{ for some } 0 \leq i < |\rho|\}.$$

### 2.1 Dynamic pushdown systems

We introduce now dynamic pushdown systems, that model 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 2.1** A *dynamic pushdown system* (DPS) 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* $s_{\mathsf{in}}$,
- $L$ is the finite set of *local states*, including the *initial local state* $\ell_{\mathsf{in}}$,
- $\Gamma$ 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 $\mathsf{int}$ 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 $(s_{\mathsf{in}})$ 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 set of final configurations of $\mathcal{P}$ is denoted by $\mathcal{F}_{\mathcal{P}}$. We will interpret this as a reachability condition.

The *size* $|c|$ of a configuration $c$ is the number $k$ of processes in $c$.

The semantics of a DPS $\mathcal{P}$ 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 $\gamma_p$.

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 $k \geq 0$,

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

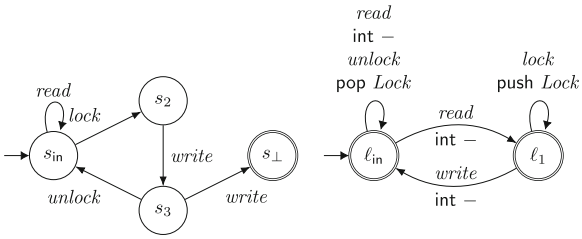


**Fig. 1** A DPS represented as a global automaton (left) and local automaton (right) synchronized with labels (in gray)

if and only if $p = k + 1$ and there are $op \in Act$ and $A \in \Gamma$ such that $((s, \ell_{\text{in}}), (op, A), (s', \ell_{k+1})) \in \Delta$ 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 $[\![\mathcal{P}]\!]$. It is *accepting* if it is contained in $Acc([\![\mathcal{P}]\!], \mathcal{F}_{\mathcal{P}})$, i.e., if it contains some final configuration.

A *dynamic finite-state system (DFS)* is simply a DPS without stacks. That is, a DFS is a tuple $\mathcal{P} = (S, L, s_{\text{in}}, \ell_{\text{in}}, \Delta, F_{\text{glob}}, F_{\text{loc}})$ where $\Delta \subseteq (S \times L) \times (S \times L)$ and the rest is defined as in DPS. Configurations in $C_{\mathcal{P}}$ are tuples $c = (s, \ell_1, \ldots, \ell_k)$ with $k \geq 0$. The semantics of $\mathcal{P}$ is $[\![\mathcal{P}]\!] = (C_{\mathcal{P}}, E, (s_{\text{in}}))$ 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 $((s, \ell_{\text{in}}), (s', \ell_{k+1})) \in \Delta$. Runs and accepting runs are defined accordingly.

***Example 2.2*** Let us study an example for a model of a lock system for a resource shared by multiple processes. This resource can be read by any process, but must be locked before being written on so that only one process can modify it at a time. We give the following implementation of this system (see Fig. 1 for an illustration) and then we will check whether it is a correct one.

Let $\mathcal{P} = (S, L, \Gamma, s_{\text{in}}, \ell_{\text{in}}, \Delta, F_{\text{glob}}, F_{\text{loc}})$ with $S = \{s_{\text{in}}, s_2, s_3, s_\perp\}$ three global states symbolizing the state of the shared resource, as well as an error state $s_\perp$, $L = \{\ell_{\text{in}}, \ell_1\}$ are two local states, $\Gamma = \{Lock\}$ a stack alphabet whose only letter is used to store locks, $F_{\text{glob}} = \{s_\perp\}$ and $F_{\text{loc}} = L$ are the accepting states, and the list of transitions is illustrated in Fig. 1 in the following way: if there is a transition $s \to s'$ in the global automaton and a transition $\ell \xrightarrow{op\ A} \ell'$ in the local automaton with the same label among $\{read, lock, write, unlock\}$, then $((s, \ell), (op, A), (s', \ell')) \in \Delta$. We use this slightly different presentation to help with readability in this example.

The idea is that the global state $s_\perp$ can only be reached if two *write* actions are made sequentially, which should not happen as only one process at a time should be able to modify the shared resource. Therefore, there is a bug in the implementation if one can find an accepting run of this DPS. An example of such a run is given in Fig. 2.

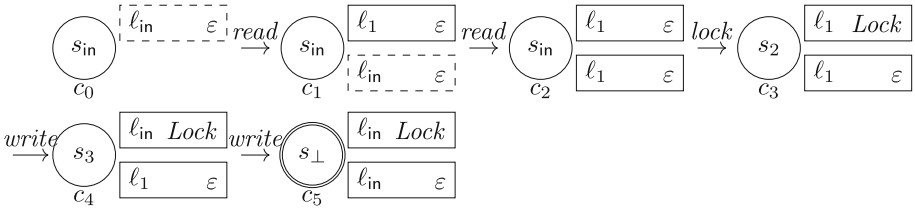


**Fig. 2** An example of an accepting finite run involving two processes

**Table 1** Reachability problems

| DPS-Reachability | DPS-Reachability$_{\text{rb}}$ |
|---|---|
| **I:** DPS $\mathcal{P}$ | **I:** DPS $\mathcal{P}$; $B \geq 1$ (given in unary) |
| **Q:** $Acc([\![\mathcal{P}]\!], \mathcal{F}_{\mathcal{P}}) \neq \emptyset$ ? | **Q:** $Acc([\![\mathcal{P}]\!]_{B\text{-rb}}, \mathcal{F}_{\mathcal{P}}^{B}) \neq \emptyset$ ? |

| DFS-Reachability | DFS-Reachability$_{\text{rb}}$ |
|---|---|
| **I:** DFS $\mathcal{P}$ | **I:** DFS $\mathcal{P}$; $B \geq 1$ (given in unary) |
| **Q:** $Acc([\![\mathcal{P}]\!], \mathcal{F}_{\mathcal{P}}) \neq \emptyset$ ? | **Q:** $Acc([\![\mathcal{P}]\!]_{B\text{-rb}}, \mathcal{F}_{\mathcal{P}}^{B}) \neq \emptyset$ ? |

### 2.2 Reachability problems

Consider the problems defined in the left part of Table 1. The problem DPS- Reachability (respectively, DFS- Reachability) consists in deciding if, in a given DPS (respectively, DFS), 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:

**Theorem 2.3** *The problem* DPS- Reachability *is undecidable, while the problem* DFS-Reachability *is decidable (and inherently non-elementary).*

The undecidability result is folklore (cf. also [35]), as two stacks are already sufficient to simulate a Turing machine. For the decidability result, we observe that dynamic systems without stacks are essentially Petri nets (cf. [9]), whose reachability problem is decidable [32] and not elementary [16].

## 3 Round-bounded reachability in dynamic pushdown systems

To regain decidability in the case of DPS, we restrict ourselves to runs that are *round bounded*.

The notion of *round-bounded* behaviors was introduced in [29]. 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 $B$*-bounded* if it uses at most $B$ rounds. Formally, given a natural number $B \geq 1$ and a DPS $\mathcal{P} = (S, L, \Gamma, s_{\text{in}}, \ell_{\text{in}}, \Delta, F_{\text{glob}}, F_{\text{loc}})$, we define the $B$*-bounded semantics* of $\mathcal{P}$ as the transition system $[\![\mathcal{P}]\!]_{B\text{-rb}} = (V^B, E^B, v_{\text{in}}^B)$ where

- Nodes are *extended 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 $v_{\mathsf{in}}^B = ((s_{\mathsf{in}}), 0, 1)$, 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 $B$-bounded semantics of a DFS is defined accordingly.

A *$B$-bounded run* (or simply *bounded run* if $B$ is understood) of $\mathcal{P}$ is a run of $[\![\mathcal{P}]\!]_{B\text{-rb}}$. As before, a $B$-bounded run is called *accepting* if it is contained in $Acc([\![\mathcal{P}]\!]_{B\text{-rb}}, \mathcal{F}_{\mathcal{P}}^B)$ where $\mathcal{F}_{\mathcal{P}}^B$ is the set of extended configurations $(c, p, r)$ such that $c \in \mathcal{F}_{\mathcal{P}}$.

***Example 3.1*** In the run of Fig. 2, we use 2 rounds:

Round 1: Process 1 performs a *read*-transition, then Process 2 takes a *read*-transition.
Round 2: Process 1 takes a *lock*-transition followed by a *write*-transition, then process 2 takes a *write*-transition.

Therefore, this run is a $B$-bounded run for any $B \geq 2$. The corresponding run in $[\![\mathcal{P}]\!]_{B\text{-rb}}$ is: $(c_0, 0, 1) \to (c_1, 1, 1) \to (c_2, 2, 1) \to (c_3, 1, 2) \to (c_4, 1, 2) \to (c_5, 2, 2)$ where the $c_i$ refer to the configuration depicted in Fig. 2.

Consider the problems on the right-hand side of Table 1 (note that $B$ is encoded in unary). For both DPS and DFS, deciding the existence of an accepting $B$-bounded run is PSPACE-complete:

**Theorem 3.2** DPS- Reachability$_{rb}$ *and* DFS- Reachability$_{rb}$ *are both PSPACE-complete.*

The upper bound has been shown in [29] and the lower bound in [30], for a model that is very similar to ours. Their proof adapted to our setting is rather straightforward, and can be found in [11].

## 4 Round-bounded control of parameterized systems

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

### 4.1 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.

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.

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 $f_j(v_0 \ldots v_{i-1}) = v_i$. Note that $\rho$ is uniquely determined by $(f_0, f_1)$.

It remains to specify when a strategy is winning. For $\mathcal{F} \subseteq V$, we say that Player 0's strategy $f_0$ is *$\mathcal{F}$-winning* if, for all strategies for Player 1 $f_1$, the unique $(f_0, f_1)$-play is $\mathcal{F}$-winning, i.e., it is contained in $Acc(\mathcal{G}, \mathcal{F})$. Dually, strategy $f_1$ of Player 1 is *$\mathcal{F}$-winning* if there is no strategy $f_0$ of Player 0 such that the unique $(f_0, f_1)$-play is in $Acc(\mathcal{G}, \mathcal{F})$.

We will add another type of winning condition. A *parity condition* is given by a ranking function $\alpha : V \to Col$ where $Col \subseteq \mathbb{N}$ is a finite set of colors. It induces the set

$$Parity(\mathcal{G}, \alpha) = \{\rho \in Runs(\mathcal{G}) \cap V^\omega \mid \min(\mathrm{Inf}_\alpha(\rho)) \text{ is even}\}$$

where $\mathrm{Inf}_\alpha(v_0 v_1 v_2 \ldots) = \{m \in Col \mid m \text{ appears infinitely often in the sequence } \alpha(v_0)\alpha(v_1)\alpha(v_2)\ldots\}$. That is, $Parity(\mathcal{G}, \alpha)$ contains an infinite run if and only if the minimal color seen infinitely often is even. We say that Player 0's strategy $f_0$ is *$\alpha$-winning* if, for all strategies $f_1$, the unique $(f_0, f_1)$-play is $\alpha$-winning, i.e., it is contained in $Parity(\mathcal{G}, \alpha)$. Finally, Player 1's strategy $f_1$ is $\alpha$-winning if there is no strategy $f_0$ such that the unique $(f_0, f_1)$-play is in $Parity(\mathcal{G}, \alpha)$.

Given one of the above winning conditions, 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 4.1** (cf. [18, 40]) *Games with a reachability or 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 DPS.

A round-bounded parameterized pushdown game is described by a DPS $\mathcal{P} = (S, L, \Gamma, s_{\mathsf{in}}, \ell_{\mathsf{in}}, \Delta, F_{\mathsf{glob}}, F_{\mathsf{loc}})$ 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 $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$ given by the transition system $[\![\mathcal{P}]\!]_{B\text{-rb}} = (V^B, E^B, v^B_{\mathsf{in}})$ 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 $s \in S_j$.

Theorem 4.1 implies that $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$ is determined.

***Example 4.2*** Let us take again the example from Fig. 1, and partition the global states in $S_0 = \{s_{\mathsf{in}}, s_\perp\}$ and $S_1 = \{s_2, s_3\}$. Intuitively, this means that Player 0 is in charge of reads and locks, while Player 1 can decide which process is allowed to write and unlock the resource after it has been locked. In that case, it is easy to see that for any bound $B$, there are no winning strategy for Player 0: any time a process $p$ makes a *lock* transition, Player 1 can make a *write* transition immediately followed by a *unlock* transition with the same process $p$, thus preventing the system to ever reach $s_\perp$.

If we take the partition $S_0 = \{s_{\mathsf{in}}, s_3, s_\perp\}$ and $S_1 = \{s_2\}$ instead, then we have a winning strategy iff $B \geq 2$ that is as follows: First make a *read* transition with two different processes, and make a *lock* transition with the second. Player 1 is then forced to make a *write* transition with either of the processes. After that, Player 0 can make a *write* transition with the other one, reaching $s_\perp$ and thus winning. One can check that the run created uses 2 rounds whatever Player 1's choice is.

Parameterized games on DFS are defined similarly as for DPS. Note that, without a bound on the number of rounds, games on DFS 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:

DPS- CONTROL$_{\text{rb}}$

**I:** DPS $\mathcal{P} = (S_0 \uplus S_1, L, \Gamma, s_{\mathsf{in}}, \ell_{\mathsf{in}}, \Delta, F_{\mathsf{glob}}, F_{\mathsf{loc}})$; $B \geq 1$
**Q:**Does Player 0 have an $\mathcal{F}^B_{\mathcal{P}}$-winning strategy in $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$ ?

The problem DFS- CONTROL$_{\text{rb}}$ is defined accordingly, the only difference being that the input $\mathcal{P}$ is a DFS.

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

- DPS- CONTROL$_{\text{rb}}$ is decidable, and
- DFS- CONTROL$_{\text{rb}}$ is inherently non-elementary.

### 4.2 Upper bound for round-bounded control

**Theorem 4.3** DPS- CONTROL$_{rb}$ *is decidable.*

Decidability of DPS- CONTROL$_{\text{rb}}$ comes from decidability of games on phase-bounded multi-pushdown systems (short: multi-pushdown games), which were first studied in [36] and rely on the phase-bounded multi-pushdown automata from [27].

*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 4.4** 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*, $\Gamma$ 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 $[\![\mathcal{M}]\!] = (V, E, v_{\mathsf{in}})$, with $V = V_0 \uplus V_1$, 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 $\sigma$, 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 $v = (s, \gamma_1, \ldots, \gamma_N, st, ph) \in V$ 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 $\sigma = st$ ($\sigma$ is the current active stack), and $st = st'$ and $ph = ph'$,
  - or $st = 0$, and $st' = \sigma$ and $ph' = ph = 1$,

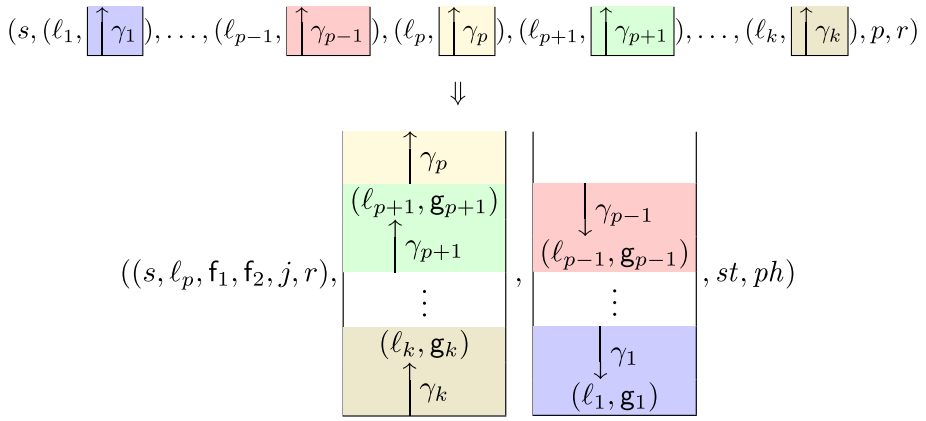


**Fig. 3** Encoding of a configuration in $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$ by a configuration in $\mathcal{G}_{\mathcal{M}}$

- or $st \notin \{0, \sigma\}$ and $ph < \kappa$, and $st' = \sigma$ and $ph' = ph + 1$.

Observe that, if $st = 0$ then, by definition, $ph = 1$, and $\sigma$ is the first active stack to be declared. Moreover, if $\sigma$ was not the current active stack, then a new phase starts (if possible).

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 MPS-Control, is defined as follows: Given an MPS $\mathcal{M}$, does Player 0 have an $\overline{\alpha}$-winning strategy in $\mathcal{G}_{\mathcal{M}}$?

**Theorem 4.5** *([5, 36]) MPS-*Control *is decidable, and is non-elementary in the number of phases.*

The upper bound was first shown in [36] by adopting the technique from [39], 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 $\kappa$-phase game. Similarly, we could try a direct proof of our Theorem 4.3 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.

We will reduce the problem DPS- $\text{Control}_{\text{rb}}$ to MPS-Control. 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 DPS and $B \geq 1$. We will build an MPS $\mathcal{M}$ such that Player 0 has a winning strategy in $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$ 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 of $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$ as a configuration in $\mathcal{G}_{\mathcal{M}}$ as depicted in Fig. 3.

Stack contents of process $p$ and of all processes $p' > p$ are stored in the first stack of the MPG, while the stack contents of processes $p' < p$ are stored *in reverse order* on the second stack. Component $j \in \{0, 1\}$ of the node's state denotes the current player. By default, it is $pl(s)$, that is, Player 0 controls the node of $\mathcal{G}_{\mathcal{M}}$ if it simulates a configuration controlled by Player 0 in $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$, and vice-versa (we will describe an exception to this rule later). We explain $\mathsf{f}_1$ and $\mathsf{f}_2$ further below.

The process $p$ that has moved last is considered as the *active* process whose local state $\ell_p$ is kept in the state of $\mathcal{G}_\mathcal{M}$ along with $s$, and whose stack content $\gamma_p$ 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).

If a player decides to take a transition for some process $p' > p$, she will store $\ell_p$ 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 Fig. 4 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 will store $\ell_p$ on stack 1 and shift the contents of stack 2 onto stack 1 to recover the local state $\ell_{p'}$ and stack contents $\gamma_{p'}$ (see Fig. 5 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 $\mathcal{G}^\mathcal{P}_{B\text{-rb}}$, which means keeping track of the local states piled in the stacks. To this aim, the state component $(s, \ell_p, \mathsf{f}_1, \mathsf{f}_2, j, r)$ of $\mathcal{M}$ contains the flags $\mathsf{f}_1, \mathsf{f}_2 \in \{\checkmark, \times\}$ where, as an invariant, we maintain $\mathsf{f}_1 = \checkmark$ if and only if $\{\ell_{p+1}, \ldots, \ell_k\} \subseteq F_{\mathsf{loc}}$ and $\mathsf{f}_2 = \checkmark$ 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 $(s, \ell, \mathsf{f}_1, \mathsf{f}_2, j, r)$ such that $s \in F_{\mathsf{glob}}$, $\ell \in F_{\mathsf{loc}}$, and $\mathsf{f}_1 = \mathsf{f}_2 = \checkmark$. To faithfully maintain the invariant, every local state $\ell_q$ that is pushed on one of the two stacks, comes with an additional flag $\mathsf{g}_q \in \{\checkmark, \times\}$, which is $\checkmark$ if and only if all local states *strictly below* on the stack are contained in $F_{\mathsf{loc}}$. 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 the multi-pushdown system $\mathcal{M}$ that take care of the various stack shifts necessary to change the active process (see gadgets pictured in Figs. 4 and 5). The problem here is that once Player $j$ commits to taking a transition by entering a gadget, she is not allowed to get stuck. Otherwise, the simulation would end abruptly and Player 1 would win the game (because the play is finite), while it does not necessarily means that Player 1 wins in $\mathcal{P}$. To ensure progress, there are transitions from inside a gadget to a state $\mathrm{win}_{1-j}$ that is winning for Player $1 - j$.

Third, suppose that, in a non-final configuration of $\mathcal{G}^\mathcal{P}_{B\text{-rb}}$, 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 $\mathcal{G}^\mathcal{P}_{B\text{-rb}}$? 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 $\kappa = 2B - 1$ (the maximal number of phases needed), $N = 2$ (the number of stacks), and $\Gamma' = \Gamma \uplus (L \times \{\checkmark, \times\})$.

*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 $s'_{\mathsf{in}}$ is the initial state. Moreover, $S_{\mathrm{sim}} = S \times L \times \{\checkmark, \times\}^2 \times \{0, 1\} \times \{1, \ldots, B\}$. A state $(s, \ell, \mathsf{f}_1, \mathsf{f}_2, j, r) \in S_{\mathrm{sim}}$ stores the global state $s$ and the local state $\ell$ of the last process $p$ that executed a transition. The third and fourth component $\mathsf{f}_1$ and $\mathsf{f}_2$ 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 as we said there will be deviations). Finally, $r$ is the current round that is simulated. Recall that $(s, \ell, \mathsf{f}_1, \mathsf{f}_2, j, r)$

represents a final configuration if and only if $s \in F_{\mathsf{glob}}$, $\ell \in F_{\mathsf{loc}}$, and $\mathsf{f}_1 = \mathsf{f}_2 = \checkmark$. Let $\mathfrak{F} \subseteq S_{\mathrm{sim}}$ be the set of such states. The states $\mathrm{win}_0$ and $\mathrm{win}_1$ 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 $S_{\mathrm{sim}}$, we let $(s, \ell, \mathsf{f}_1, \mathsf{f}_2, j, r) \in S'_j$. The states $\mathrm{win}_0$ and $\mathrm{win}_1$ both belong to Player 0 (but this does not really matter). Membership of intermediate states is defined below. The ranking function $\alpha$ maps $\mathrm{win}_0$ 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 $(s, \ell, \mathsf{f}_1, \mathsf{f}_2, j, r) \in \mathfrak{F}$, we will have a transition $(s, \ell, \mathsf{f}_1, \mathsf{f}_2, j, r) \xrightarrow{\mathsf{int}} \mathrm{win}_0$ (we omit the stack symbol, as it is meaningless), which will be the only transition outgoing from $(s, \ell, \mathsf{f}_1, \mathsf{f}_2, j, r)$. 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 active process. For all $(s, \ell, \mathsf{f}_1, \mathsf{f}_2, j, r) \in S_{\mathrm{sim}} \setminus \mathfrak{F}$ and transitions $(s, \ell) \xrightarrow{(op,A)} (s', \ell')$ from $\Delta$ (in $\mathcal{P}$), the MPS $\mathcal{M}$ has a transition $(s, \ell, \mathsf{f}_1, \mathsf{f}_2, j, r) \xrightarrow{(op,1,A)} (s', \ell', \mathsf{f}_1, \mathsf{f}_2, pl(s'), r)$.

*Change of Player* As we have said, when a player enters a gadget to simulate a change of process, she is committed to complete the change. If no transition in the original game is available from a configuration belonging to Player 1, in the multi-pushdown game, that same player will have no choice but eventually taking a transition leading to $\mathrm{win}_0$, allowing Player 0 to win the game $\mathcal{G}_{\mathcal{M}}$. However, if the blocking configuration was not winning in $\mathcal{P}$, Player 1 should win the game. To get around this discrepancy, when Player 1 thinks he does not have an outgoing transition (in $\mathcal{P}$), he can force Player 0 to play instead of himself. That is, for all $(s, \ell, \mathsf{f}_1, \mathsf{f}_2, 1, r) \in S_{\mathrm{sim}} \setminus \mathfrak{F}$, we introduce the transition $(s, \ell, \mathsf{f}_1, \mathsf{f}_2, 1, r) \xrightarrow{\mathsf{int}} (s, \ell, \mathsf{f}_1, \mathsf{f}_2, 0, r)$. Note that if Player 1 forced a change of player when there was actually an outgoing transition in $\mathcal{P}$, then Player 0 wins immediately after simulating such a transition, as we will describe a bit later.

*Transitions for Process Change* We now introduce the transitions needed to accomplish a process change in the same round. In doing so, we will introduce the sets of intermediate states $\mathfrak{I}_?$, $\mathfrak{I}_{\mathrm{NP}}$, $\mathfrak{I}_{\mathrm{NR}}$.

### Within the same round

For all $(s, \ell, \mathsf{f}_1, \mathsf{f}_2, j, r) \in S_{\mathrm{sim}} \setminus \mathfrak{F}$, we introduce, in $\mathcal{M}$, the gadget given in Fig. 4. As we move to another process, the current local state $\ell$ is pushed on stack 2, along with flag $\mathsf{f}_2$, which tells us whether, henceforth, all states on stack 2 *below* the new stack symbol are local accepting states. Afterwards, the value of $\mathsf{f}_2$ kept in the global state has to be updated, depending on whether $\ell \in F_{\mathsf{loc}}$ or not. Actually, maintaining the value of $\mathsf{f}_2$ is done in terms of additional (but finitely many) states. For the sake of readability, however, we rather consider that $\mathsf{f}_2$ is a variable and use $\mathsf{upd}(\mathsf{f}_2, \ell)$ to update its value. We continue shifting the contents of stack 1 onto stack 2 (updating $\mathsf{f}_2$ when retrieving a local state). Now, there are two possibilities. We may eventually pop a new current local state $\hat{\ell}$ and then simulate

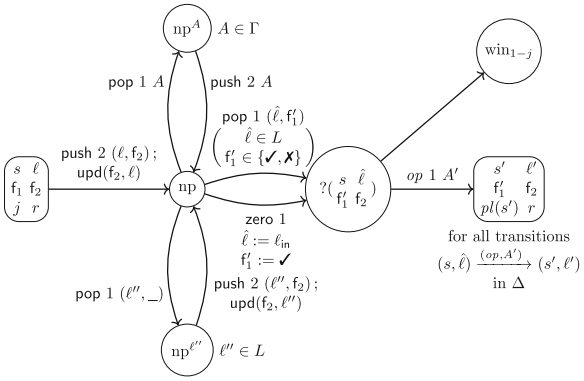


**Fig. 4** Change from process $p$ to some process $p' > p$ (staying in the same round). All intermediate states belong to Player $j$; from every intermediate state, there is an outgoing internal transition to $\mathrm{win}_{1-j}$. Moreover, $\mathsf{upd}(\mathsf{f}_2, \bar{\ell})$ stands for the update rule IF $(f_2 = \checkmark \wedge \bar{\ell} \in F_{\mathsf{loc}})$ THEN $\mathsf{f}_2 := \checkmark$ ELSE $\mathsf{f}_2 := \boldsymbol{\times}$

the transition of the corresponding existing process. Or, when there are no more symbols on stack 1, we create a new process.

Formally, we define the set $\mathfrak{I}_{\mathrm{NP}}$ of intermediate states for a process change. For every $A \in \Gamma$, $s \in S$, $\ell \in L$, $\mathsf{f}_1, \mathsf{f}_2, \mathsf{g} \in \{\checkmark, \boldsymbol{\times}\}$, $j \in \{0, 1\}$, and $r \leq B$, we include $\mathrm{np}(s, \mathsf{f}_1, \mathsf{f}_2, j, r)$, $\mathrm{np}^A(s, \mathsf{f}_1, \mathsf{f}_2, j, r)$, and $\mathrm{np}^\ell(s, \mathsf{f}_1, \mathsf{f}_2, j, r)$ in $\mathfrak{I}_{\mathrm{NP}}$, and we add the transitions

1. $(s, \ell, \mathsf{f}_1, \mathsf{f}_2, j, r) \xrightarrow{\mathsf{push}\ 2\ (\ell, \mathsf{f}_2)} \mathrm{np}(s, \mathsf{f}_1, \mathsf{upd}(\mathsf{f}_2, \ell), j, r)$,
2. $\mathrm{np}(s, \mathsf{f}_1, \mathsf{f}_2, j, r) \xrightarrow{\mathsf{pop}\ 1\ A} \mathrm{np}^A(s, \mathsf{f}_1, \mathsf{f}_2, j, r)$,
3. $\mathrm{np}^A(s, \mathsf{f}_1, \mathsf{f}_2, j, r) \xrightarrow{\mathsf{push}\ 2\ A} \mathrm{np}(s, \mathsf{f}_1, \mathsf{f}_2, j, r)$,
4. $\mathrm{np}(s, \mathsf{f}_1, \mathsf{f}_2, j, r) \xrightarrow{\mathsf{pop}\ 1\ (\ell, \mathsf{g})} \mathrm{np}^\ell(s, \mathsf{g}, \mathsf{f}_2, j, r)$,
5. $\mathrm{np}^\ell(s, \mathsf{f}_1, \mathsf{f}_2, j, r) \xrightarrow{\mathsf{push}\ 2\ (\ell, \mathsf{f}_2)} \mathrm{np}(s, \mathsf{f}_1, \mathsf{upd}(\mathsf{f}_2, \ell), j, r)$,
6. $\mathrm{np}(s, \mathsf{f}_1, \mathsf{f}_2, j, r) \xrightarrow{\mathsf{pop}\ 1\ (\ell, \mathsf{g})} ?(s, \ell, \mathsf{g}, \mathsf{f}_2, j, r)$, and
7. $\mathrm{np}(s, \mathsf{f}_1, \mathsf{f}_2, j, r) \xrightarrow{\mathsf{zero}\ 1} ?(s, \ell_{\mathsf{in}}, \checkmark, \mathsf{f}_2, j, r)$

where $\mathsf{upd}(\mathsf{f}_2, \ell) = \checkmark$ iff $\mathsf{f}_2 = \checkmark \wedge \ell \in F_{\mathsf{loc}}$. States of the form $?(s, \ell, \mathsf{g}, \mathsf{f}_2, j, r)$ are used to exit this gadget, and will be defined shortly after.

### In the next round

For all $(s, \ell, \mathsf{f}_1, \mathsf{f}_2, j, r) \in S_{\mathrm{sim}} \backslash \mathfrak{F}$ such that $r < B$, we introduce, in $\mathcal{M}$, the gadget given in Fig. 5. It is similar to the previous gadget. However, we now shift symbols from stack 2 onto stack 1 and have to update $\mathsf{f}_1$ accordingly.

Formally, let $\mathfrak{I}_{\mathrm{NR}}$ the set of intermediate states for a round change. For every $A \in \Gamma$, $s \in S$, $\ell \in L$, $\mathsf{f}_1, \mathsf{f}_2, \mathsf{g} \in \{\checkmark, \boldsymbol{\times}\}$, $j \in \{0, 1\}$, and $r \leq B$, we include $\mathrm{nr}(s, \mathsf{f}_1, \mathsf{f}_2, j, r)$, $\mathrm{nr}^A(s, \mathsf{f}_1, \mathsf{f}_2, j, r)$, and $\mathrm{nr}^\ell(s, \mathsf{f}_1, \mathsf{f}_2, j, r)$ in $\mathfrak{I}_{\mathrm{NR}}$, together with the following transitions:

1. $(s, \ell, \mathsf{f}_1, \mathsf{f}_2, j, r) \xrightarrow{\mathsf{push}\ 1\ (\ell, \mathsf{f}_1)} \mathrm{nr}(s, \mathsf{upd}(\mathsf{f}_1, \ell), \mathsf{f}_2, j, r)$,
2. $\mathrm{nr}(s, \mathsf{f}_1, \mathsf{f}_2, j, r) \xrightarrow{\mathsf{pop}\ 2\ A} \mathrm{nr}^A(s, \mathsf{f}_1, \mathsf{f}_2, j, r)$,
3. $\mathrm{nr}^A(s, \mathsf{f}_1, \mathsf{f}_2, j, r) \xrightarrow{\mathsf{push}\ 1\ A} \mathrm{nr}(s, \mathsf{f}_1, \mathsf{f}_2, j, r)$,
4. $\mathrm{nr}(s, \mathsf{f}_1, \mathsf{f}_2, j, r) \xrightarrow{\mathsf{pop}\ 2\ (\ell, \mathsf{g})} \mathrm{nr}^\ell(s, \mathsf{f}_1, \mathsf{g}, j, r)$,
5. $\mathrm{nr}^\ell(s, \mathsf{f}_1, \mathsf{f}_2, j, r) \xrightarrow{\mathsf{push}\ 1\ (\ell, \mathsf{f}_1)} \mathrm{nr}(s, \mathsf{upd}(\mathsf{f}_1, \ell), \mathsf{f}_2, j, r)$, and
6. $\mathrm{nr}(s, \mathsf{f}_1, \mathsf{f}_2, j, r) \xrightarrow{\mathsf{pop}\ 2\ (\ell, \mathsf{g})} ?(s, \ell, \mathsf{f}_1, \mathsf{g}, j, r + 1)$.

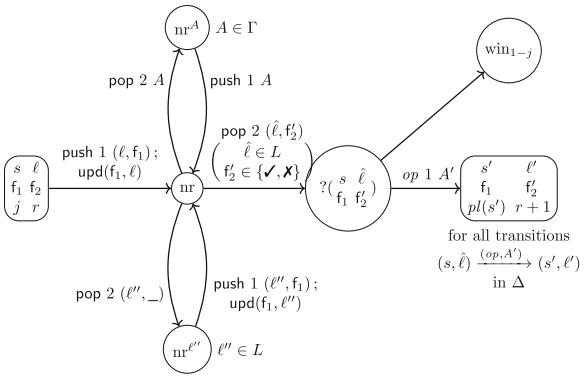


**Fig. 5** Going from a process $p$ to some process $p' < p$ (involving a round change). All intermediate states belong to Player $j$; from every intermediate state, there is an outgoing internal transition to $\mathrm{win}_{1-j}$. Moreover, $\mathsf{upd}(\mathsf{f}_1, \bar{\ell})$ stands for the update rule IF $(f_1 = \checkmark \wedge \bar{\ell} \in F_{\mathsf{loc}})$ THEN $\mathsf{f}_1 := \checkmark$ ELSE $\mathsf{f}_1 := \times$

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 a dummy push and pop transition on stack 1.

### Exiting the gadgets

First, for all $s \in S, \ell \in L, \mathsf{f}_1, \mathsf{f}_2 \in \{\checkmark, \times\}$, $j \in \{0, 1\}$, and $r \leq B$, we have $?(s, \ell, \mathsf{f}_1, \mathsf{f}_2, j, r) \in \mathfrak{I}_?$. For all such states, there is a transition

1. $?(s, \ell, \mathsf{f}_1, \mathsf{f}_2, j, r) \xrightarrow{\mathsf{int}} \mathrm{win}_{1-j}$.

We also add

2. $(s, \ell, \mathsf{f}_1, \mathsf{f}_2, j, r) \xrightarrow{\mathsf{int}} \mathrm{win}_{1-j}$ for all $(s, \ell, \mathsf{f}_1, \mathsf{f}_2, j, r) \in S_{\mathrm{sim}} \setminus \mathfrak{F}$.

These two transitions force a player to lose if there is no other transition available.

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

3. $?(s, \ell, \mathsf{f}_1, \mathsf{f}_2, j, r) \xrightarrow{op\,1\,A} (s', \ell', \mathsf{f}_1, \mathsf{f}_2, pl(s'), r)$

which completes the simulation of a transition from $\mathcal{P}$.

Finally, when $j \neq pl(s)$, for all transitions $(s, \ell) \xrightarrow{(op,A)} (s', \ell')$ in $\Delta$, there are two additional transitions

4. $?(s, \ell, \mathsf{f}_1, \mathsf{f}_2, j, r) \xrightarrow{(op,A)} \mathrm{win}_j$, and
5. $(s, \ell, \mathsf{f}_1, \mathsf{f}_2, j, r) \xrightarrow{(op,A)} \mathrm{win}_j$.

These last transitions allow Player 0 to win immediately in the case of a change of player, if Player 1 forced the change despite having an outgoing transition in $\mathcal{P}$. Otherwise, Player 0 will have no other choice but to take the transition leading to $\mathrm{win}_1$.

We can now state correctness of our construction.

**Lemma 4.6** *Player 0 has a winning strategy in $\mathcal{G}_{\mathcal{M}}$ if and only if Player 0 has a winning strategy in $\mathcal{G}^{\mathcal{P}}_{B\text{-}rb}$.*

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

Let $[\![\mathcal{P}]\!]_{B\text{-rb}} = (V^B, E^B, v_{\text{in}}^B)$ and $[\![\mathcal{M}]\!] = (V, E, v_{\text{in}})$. By construction, every play of $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$ 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 $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$. Let us formalize this intuition by giving a mapping $\pi$ between plays of $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$ and plays of $\mathcal{G}_{\mathcal{M}}$.

In the base game $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$, 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 $p' > p$.

We first define $\pi$ on nodes of $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$. Let $c = (s, (\ell_1, \gamma_1) \dots (\ell_k, \gamma_k))$ be a configuration, and $p \in \{1, \dots, k\}$ a process. We define the following flags

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

and the corresponding two stacks of the multi-pushdown game

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

where $\widetilde{\gamma}$ is the mirror word of $\gamma$.

Given a round $r$, we then let

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

Now, let $(c', p', r')$ be a successor of $(c, p, r)$ in $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$. If $p' \neq p$, $\pi(c', p', r')$ will not be a successor of $\pi(c, p, r)$ in $\mathcal{G}_{\mathcal{M}}$, because of the mechanism of process or round change that will introduce several intermediate configurations. We denote by $\text{next}_p^{p'}(\pi(c, p, r))$ the part of the play between $\pi(c, p, r)$ and $\pi(c', p', r')$ in $\mathcal{G}_{\mathcal{M}}$. Its exact form depends on whether it involves a round change or not, and its length depends on the number of processes between $p$ and $p'$. To define it precisely, we introduce $\text{loop}_i^{\text{np}}$ and $\text{loop}_i^{\text{nr}}$ that describe the partial runs that occur inside the process change gadgets (staying in the same round and involving a round change respectively).

We define recursively the function $\text{loop}_i^{\text{np}} : \mathfrak{I}_{\text{NP}} \times (\Gamma'^*)^2 \times \{1\} \times \mathbb{N} \to V^*$, for $i \in \mathbb{N}^*$ that gives the portion of the run executed in $\mathcal{G}_{\mathcal{M}}$ when we transfer stack contents of $i$ processes from stack 1 to stack 2. Let $\text{np}(s, \mathsf{f}_1, \mathsf{f}_2, j, r) \in \mathfrak{I}_{\text{NP}}$, $A \in \Gamma$, $\ell \in L$, $\mathsf{g} \in \{\checkmark, \times\}$, and $\overline{\gamma_1}, \overline{\gamma_2} \in \Gamma'^*$.

Then,

$$\begin{aligned}
&\text{loop}_i^{\text{np}}(\text{np}(s, \mathsf{f}_1, \mathsf{f}_2, j, r), A \cdot \overline{\gamma_1}, \overline{\gamma_2}, 1, ph) \\
&\quad = (\text{np}^A(s, \mathsf{f}_1, \mathsf{f}_2, j, r), \overline{\gamma_1}, \overline{\gamma_2}, 1, ph) \cdot (\text{np}(s, \mathsf{f}_1, \mathsf{f}_2, j, r), \overline{\gamma_1}, A \cdot \overline{\gamma_2}, 1, ph) \\
&\quad \cdot \text{loop}_i^{\text{np}}(\text{np}(s, \mathsf{f}_1, \mathsf{f}_2, j, r), \overline{\gamma_1}, A \cdot \overline{\gamma_2}, 1, ph) \qquad (1)
\end{aligned}$$

describes the transfer of the first element of the stack from stack 1 to stack 2.

$$\begin{aligned}&\mathrm{loop}_i^{\mathrm{np}}(\mathrm{np}(s,\mathsf{f}_1,\mathsf{f}_2,j,r),(\ell,\mathsf{g})\cdot\overline{\gamma_1},\overline{\gamma_2},1,ph)\\&\quad=(\mathrm{np}^{\ell}(s,\mathsf{g},\mathsf{f}_2,j,r)),\overline{\gamma_1},\overline{\gamma_2},1,ph)\cdot(\mathrm{np}(s,\mathsf{g},\mathsf{upd}(\mathsf{f}_2,\ell),j,r),\overline{\gamma_1},(\ell,\mathsf{f}_2)\cdot\overline{\gamma_2},1,ph)\\&\quad\cdot\mathrm{loop}_{i-1}^{\mathrm{np}}(\mathrm{np}(s,\mathsf{g},\mathsf{upd}(\mathsf{f}_2,\ell),j,r),\overline{\gamma_1},(\ell,\mathsf{f}_2)\cdot\overline{\gamma_2},1,ph)\text{ if }i>1,\end{aligned}\tag{2}$$

describes the transfer of the next local state to stack 2 when $i > 1$, i.e. when the process we want to simulate is not the next one.

$$\mathrm{loop}_1^{\mathrm{np}}(\mathrm{np}(s,\mathsf{f}_1,\mathsf{f}_2,j,r),(\ell,\mathsf{g})\cdot\overline{\gamma_1},\overline{\gamma_2},1,ph)=(?(s,\ell,\mathsf{g},\mathsf{f}_2,j,r),\overline{\gamma_1},\overline{\gamma_2},1,ph)\tag{3}$$

describes the fact that the transfer of contents between stack 1 and stack 2 is finished and that we have reached the configuration corresponding to "$(c, p', r)$".

$$\mathrm{loop}_1^{\mathrm{np}}(\mathrm{np}(s,\mathsf{f}_1,\mathsf{f}_2,j,r),\varepsilon,\overline{\gamma_2},1,ph)=(?(s,\ell_{\mathsf{in}},\checkmark,\mathsf{f}_2,j,r),\varepsilon,\overline{\gamma_2},1,ph)\tag{4}$$

describes the case where a new process is created.

If $i$ is smaller than the number of elements from $L \times \{\checkmark, \times\}$ in $\gamma_1$, it is easy to see that $\mathrm{loop}_i^{\mathrm{np}}(\mathrm{np}(s,\mathsf{f}_1,\mathsf{f}_2,j,r),\overline{\gamma_1},\overline{\gamma_2},1,ph)$ is well defined, since the number of elements of $\overline{\gamma_1}$ strictly decreases at each iteration, and since the index $i$ decreases at each popping of an element of $L \times \{\checkmark, \times\}$.

Similarly, we define recursively $\mathrm{loop}_i^{\mathrm{nr}} : \mathfrak{I}_{\mathrm{NP}} \times (\Gamma'^*)^2 \times \{2\} \times \mathbb{N} \to V^*$, for $i \in \mathbb{N}^*$ to describe the portion of the play executed in $\mathcal{G}_{\mathcal{M}}$ when we transfer stack contents of $i$ processes from stack 2 to stack 1, simulating a round change. Let $\mathrm{nr}(s,\mathsf{f}_1,\mathsf{f}_2,j,r) \in \mathfrak{I}_{\mathrm{NP}}$, $A \in \Gamma$, $\ell \in L$, $\mathsf{g} \in \{\checkmark, \times\}$, $\overline{\gamma_1}, \overline{\gamma_2} \in \Gamma'^*$. Then,

$$\begin{aligned}&\mathrm{loop}_i^{\mathrm{nr}}(\mathrm{nr}(s,\mathsf{f}_1,\mathsf{f}_2,j,r),\overline{\gamma_1},A\cdot\overline{\gamma_2},st,ph)\\&\quad=(\mathrm{nr}^{A}(s,\mathsf{f}_1,\mathsf{f}_2,j,r),\overline{\gamma_1},\overline{\gamma_2},2,2r)\cdot(\mathrm{nr}(s,\mathsf{f}_1,\mathsf{f}_2,j,r),A\cdot\overline{\gamma_1},\overline{\gamma_2},2,2r)\\&\quad\cdot\mathrm{loop}_i^{\mathrm{nr}}(\mathrm{nr}(s,\mathsf{f}_1,\mathsf{f}_2,j,r),A\cdot\overline{\gamma_1},\overline{\gamma_2},2,2r)\end{aligned}\tag{5}$$

with either $st = 1$ and $ph = 2r - 1$ or $st = 2$ and $ph = 2r$.

$$\begin{aligned}&\mathrm{loop}_i^{\mathrm{nr}}(\mathrm{nr}(s,\mathsf{f}_1,\mathsf{f}_2,j,r),\overline{\gamma_1},(\ell,\mathsf{g})\cdot\overline{\gamma_2},st,ph)\\&\quad=(\mathrm{nr}^{\ell}(s,\mathsf{f}_1,\mathsf{g},j,r)),\overline{\gamma_1},\overline{\gamma_2},2,2r)\cdot(\mathrm{nr}(s,\mathsf{upd}(\mathsf{f}_1,\ell),\mathsf{g},j,r),(\ell,\mathsf{f}_1)\cdot\overline{\gamma_1},\overline{\gamma_2},2,2r)\\&\quad\cdot\mathrm{loop}_{i-1}^{\mathrm{nr}}(\mathrm{nr}(s,\mathsf{upd}(\mathsf{f}_1,\ell),\mathsf{g},j,r),(\ell,\mathsf{f}_1)\cdot\overline{\gamma_1},\overline{\gamma_2},2,2r)\text{ if }i>1\end{aligned}\tag{6}$$

with either $st = 1$ and $ph = 2r - 1$ or $st = 2$ and $ph = 2r$.

$$\begin{aligned}&\mathrm{loop}_1^{\mathrm{nr}}(\mathrm{nr}(s,\mathsf{f}_1,\mathsf{f}_2,j,r),\overline{\gamma_1},(\ell,\mathsf{g})\cdot\overline{\gamma_2},st,ph)\\&\quad=(?(s,\ell,\mathsf{f}_1,\mathsf{g},j,r+1),\overline{\gamma_1},\overline{\gamma_2},1,2r+1)^{1}\end{aligned}\tag{7}$$

with either $st = 1$ and $ph = 2r - 1$ or $st = 2$ and $ph = 2r$.[1]

Again, if $i$ is smaller than the number of elements from $L \times \{\checkmark, \times\}$ in $\gamma_2$, one can see that $\mathrm{loop}_i^{\mathrm{nr}}(\mathrm{nr}(s,\mathsf{f}_1,\mathsf{f}_2,j,r),\overline{\gamma_1},\overline{\gamma_2},2,2r+1)$ is well defined as the number of elements of $\overline{\gamma_2}$ strictly decreases at each iteration and the index $i$ decreases when an element from $L \times \{\checkmark, \times\}$ is popped.

We first observe the following property:

[1] Normally, the last two elements of the tuple (active stack and current phase) should be 2 and $2r$. But as mentioned above, we assume that after the round change, the active stack is again stack 1, which can be achieved by pushing some element on stack 1 and popping it back immediately. We omit the necessary intermediate states to alleviate notations.

**Lemma 4.7** *For all node* $u = (np(s, \mathsf{f}_1, \mathsf{f}_2, j, r), A_1 \cdots A_n \cdot \overline{\gamma_1}, \overline{\gamma_2}, 1, 2r-1)$*, with* $A_1 \cdots A_n \in \Gamma^*$*,* $\overline{\gamma_1} \in ((L \times \{\checkmark, \times\}).\Gamma^*)^*$ *and* $\overline{\gamma_2} \in \Gamma'^*$*, and for all* $i \in \mathbb{N}^*$*, there exists* $\rho$ *such that* $u \cdot loop_i^{np}(u) = u \cdot \rho \cdot loop_i^{np}(v)$*, where* $u \cdot \rho$ *is a partial play ending in* $v$*, with* $v = (np(s, \mathsf{f}_1, \mathsf{f}_2, j, r), \overline{\gamma_1}, A_n \cdots A_1 \cdot \overline{\gamma_2}, 1, 2r-1)$.

*For all* $u = (nr(s, \mathsf{f}_1, \mathsf{f}_2, j, r), \overline{\gamma_1}, A_1 \cdots A_n \cdot \overline{\gamma_2}, st, ph)$*, with* $\overline{\gamma_1} \in (\Gamma')^*$*,* $A_1 \cdots A_n \in \Gamma^*$*, and* $\overline{\gamma_2} \in ((L \times \{\checkmark, \times\}).\Gamma^*)^*$*, and for all* $i \in \mathbb{N}^*$*, there exists* $\rho$ *such that* $u \cdot loop_i^{nr}(u) = u \cdot \rho \cdot loop_i^{nr}(v)$*, where* $u \cdot \rho$ *is a partial play ending in* $v$*, with* $v = (nr(s, \mathsf{f}_1, \mathsf{f}_2, j, r), A_n \cdots A_1 \cdot \overline{\gamma_1}, \overline{\gamma_2}, 2, 2r)$.

***Proof*** We prove the first part of the lemma. This can be easily shown by induction on the size of $A_1 \ldots A_n$. If $n = 0$, it is trivial as it means that $v = u$ and $\rho = \varepsilon$. If $n > 0$, then by definition of $\text{loop}_i^{\text{np}}$,

$$\begin{aligned}\text{loop}_i^{\text{np}}(u) =& (\text{np}^{A_1}(s, \mathsf{f}_1, \mathsf{f}_2, j, r), A_2 \cdots A_n \cdot \overline{\gamma_1}, \overline{\gamma_2}, 1, 2r-1) \\ & \cdot (\text{np}(s, \mathsf{f}_1, \mathsf{f}_2, j, r), A_2 \cdots A_n \cdot \overline{\gamma_1}, A_1 \cdot \overline{\gamma_2}, 1, 2r-1) \\ & \cdot \text{loop}_i^{\text{np}}(\text{np}(s, \mathsf{f}_1, \mathsf{f}_2, j, r), A_2 \cdots A_n \cdot \overline{\gamma_1}, A_1 \cdot \overline{\gamma_2}, 1, 2r-1)\end{aligned}$$

which by induction hypothesis can be rewritten as

$$\begin{aligned}\text{loop}_i^{\text{np}}(u) =& (\text{np}^{A_1}(s, \mathsf{f}_1, \mathsf{f}_2, j, r), A_2 \cdots A_n \cdot \overline{\gamma_1}, \overline{\gamma_2}, 1, 2r-1) \\ & \cdot (\text{np}(s, \mathsf{f}_1, \mathsf{f}_2, j, r), A_2 \cdots A_n \cdot \overline{\gamma_1}, A_1 \cdot \overline{\gamma_2}, 1, 2r-1) \\ & \cdot \rho' \cdot \text{loop}_i^{\text{np}}(v)\end{aligned}$$

with $(\text{np}(s, \mathsf{f}_1, \mathsf{f}_2, j, r), A_2 \cdots A_n \cdot \overline{\gamma_1}, A_1 \cdot \overline{\gamma_2}, 1, 2r-1) \cdot \rho'$ a partial play ending in $v$.

If we let

$$\begin{aligned}\rho = & (\text{np}^{A_1}(s, \mathsf{f}_1, \mathsf{f}_2, j, r), A_2 \cdots A_n \cdot \overline{\gamma_1}, \overline{\gamma_2}, 1, 2r-1) \\ & \cdot (\text{np}(s, \mathsf{f}_1, \mathsf{f}_2, j, r), A_2 \cdots A_n \cdot \overline{\gamma_1}, A_1 \cdot \overline{\gamma_2}, 1, 2r-1) \cdot \rho'\end{aligned}$$

we have that $u \cdot \rho$ is a valid run of $\mathcal{G}_{\mathcal{M}}$ (because of transitions of type 2 and 3 given in the definition of $\mathfrak{I}_{\text{NP}}$) that ends in node $v$, so the property is proved. □

Then we can define $\text{next}_p^{p'}$ as follows:

$$\text{next}_p^{p'}((s, \ell, \mathsf{f}_1, \mathsf{f}_2, j, r), \overline{\gamma_1}, \overline{\gamma_2}, 1, ph)) =$$

$$\begin{cases} \varepsilon & \text{if } p = p' \\ (\text{np}(s, \mathsf{f}_1, \mathsf{upd}(\mathsf{f}_2, \ell), j, r), \overline{\gamma_1}, (\ell, \mathsf{f}_2) \cdot \overline{\gamma_2}, 1, ph) \cdot & \\ \quad \text{loop}_{p'-p}^{\text{np}}(\text{np}(s, \mathsf{f}_1, \mathsf{upd}(\mathsf{f}_2, \ell), j, r), \overline{\gamma_1}, (\ell, \mathsf{f}_2) \cdot \overline{\gamma_2}, 1, ph) & \text{if } p < p' \\ (\text{nr}(s, \mathsf{upd}(\mathsf{f}_1, \ell), \mathsf{f}_2, j, r), (\ell, \mathsf{f}_1) \cdot \overline{\gamma_1}, \overline{\gamma_2}, 2, ph+1)^2 \cdot & \\ \quad \text{loop}_{p-p'}^{\text{nr}}(\text{nr}(s, \mathsf{upd}(\mathsf{f}_1, \ell), \mathsf{f}_2, j, r), (\ell, \mathsf{f}_1) \cdot \overline{\gamma_1}, \overline{\gamma_2}, 2, ph+1) & \text{if } p' < p \end{cases}$$

We are now ready to extend the definition of $\pi$ on plays.[2] We define

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

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

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

[2] Similarly and for simplicity, here we assume that the second stack becomes active.

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

$$\pi(\rho \cdot u') = \begin{cases} \pi(\rho) & \text{if } \rho \text{ is winning, else} \\ \pi(\rho) \cdot \text{next}_p^{p'}(\pi(u)) \cdot \pi(u') \cdot W_0^\omega & \text{if } u' \in \mathcal{F}, \text{ else} \\ \pi(\rho) \cdot \text{next}_p^{p'}(\pi(u)) \cdot \pi(u') \cdot \text{Ch}_0(u') \cdot W_1^\omega & \text{if } u' \notin \mathcal{F} \text{ and has no} \\ & \text{successor, else} \\ \pi(\rho) \cdot \text{next}_p^{p'}(\pi(u)) \cdot \pi(u'). \end{cases}$$

$$\text{where } \text{Ch}_0(u') = \begin{cases} ((s, \ell, \mathsf{f}_1, \mathsf{f}_2, 0, r), \overline{\gamma_1}, \overline{\gamma_2}, 1, ph) & \text{if } pl(s) = 1 \text{ and} \\ & \pi(u') = ((s, \ell, \mathsf{f}_1, \mathsf{f}_2, 1, r), \overline{\gamma_1}, \overline{\gamma_2}, 1, ph) \\ \varepsilon & \text{otherwise.} \end{cases}$$

and $W_i = (\text{win}_i, \tau_1(c', p'), \tau_2(c', p'), 1, 2r' - 1)$ for $i \in \{0, 1\}$.

We extend the mapping to infinite plays in the following way: if $\rho$ is an infinite play, we let $\pi(\rho) = \lim_{\rho' \sqsubseteq \rho} \pi(\rho')$.

The correctness of this mapping relies on the fact that $\text{next}_p^{p'}(\pi(u)) \cdot \pi(u')$ is indeed a partial play of $\mathcal{G}_{\mathcal{M}}$. This is ensured by the following lemma:

**Lemma 4.8** *Given a configuration $c = (s, (\ell_1, \gamma_1) \dots (\ell_k, \gamma_k))$ and two processes $p \in \{1, \dots, k\}$, $p' \in \{1, \dots, k+1\}$ such that $p \neq p'$, $next_p^{p'}(\pi(c, p, r))$ and $next_p^{p'}(Ch_0(c, p, r))$ are partial plays of $\mathcal{G}_{\mathcal{M}}$ ending respectively in the state*

$$(?(s, \ell_{p'}, \mathsf{g}^{>}(c, p'), \mathsf{g}^{<}(c, p'), pl(s), r'), \tau_1(c, p'), \tau_2(c, p'), 1, 2r' - 1).$$

*and*

$$(?(s, \ell_{p'}, \mathsf{g}^{>}(c, p'), \mathsf{g}^{<}(c, p'), 0, r'), \tau_1(c, p'), \tau_2(c, p'), 1, 2r' - 1).$$

*Moreover, if $p < p'$, then $r' = r$, if $p > p'$ then $r' = r + 1$, and the entire portion of the play belongs to the same player.*

***Proof*** We show the proof in the case $p < p'$, and of $\pi(c, p, r)$, the other cases being extremely similar.

Let $c, p, p', r$ defined as stated in the lemma with $p < p' \leq k + 1$, and let $\pi(c, p, r) = ((s, \ell_p, \mathsf{f}_1, \mathsf{f}_2, pl(s), r), \overline{\gamma_1}, \overline{\gamma_2}, 1, 2r - 1)$. By definition, we have that $\mathsf{f}_1 = \mathsf{g}^{>}(c, p)$, $\mathsf{f}_2 = \mathsf{g}^{<}(c, p)$, $\overline{\gamma_1} = \tau_1(c, p)$, and $\overline{\gamma_2} = \tau_2(c, p)$. If we let $u = (\text{np}(s, \mathsf{f}_1, \mathsf{upd}(\mathsf{f}_2, \ell_p), j, r), \overline{\gamma_1}, (\ell_p, \mathsf{f}_2) \cdot \overline{\gamma_2}, 1, 2r - 1)$ then $\text{next}_p^{p'}(\pi(c, p, r)) = u \cdot \text{loop}^{\text{np}}_{p'-p}(u)$.

We show that for all $0 \leq i < p' - p$, if we let

$$\begin{aligned} u_i = (&\text{np}(s, \mathsf{g}^{>}(c, p + i), \mathsf{g}^{<}(c, p + i + 1), j, r), \\ &\tau_1(c, p + i), (\ell_{p+i}, \mathsf{g}^{<}(c, p + i)) \cdot \tau_2(c, p + i), 1, 2r - 1), \end{aligned}$$

then $u_i \cdot \text{loop}^{\text{np}}_{p'-p-i}(u_i)$ is a valid partial play that ends in node

$$(?(s, \ell_{p'}, \mathsf{g}^{>}(c, p'), \mathsf{g}^{<}(c, p'), pl(s), r), \tau_1(c, p'), \tau_2(c, p'), 1, 2r - 1).$$

Note that the node $u$ defined above is $u_0$, so proving that this property holds for $i = 0$ proves the Lemma.

Suppose that $i = p' - p - 1$, i.e $p + i$ is the process immediately preceding $p'$. There are two different cases to study depending on whether $p' = k + 1$ or not.

1) If $p' = k+1$, then $p+i = k$ is the last process of $c$ and therefore $\tau_1(c, p+i) = \gamma_k$. In that case, $u_i = (\mathrm{np}(s, \mathsf{g}^{>}(c,k), \mathsf{g}^{<}(c,k+1), j, r), \gamma_k, (\ell_k, \mathsf{g}^{<}(c,k)) \cdot \tau_2(c,k), 1, 2r-1)$.

Thus, from Lemma 4.7, $u_i \cdot \mathrm{loop}_1^{\mathrm{np}}(u_i) = u_i \cdot \rho \cdot \mathrm{loop}_1^{\mathrm{np}}(v_i)$, with $v_i = (\mathrm{np}(s, \mathsf{g}^{>}(c,k), \mathsf{g}^{<}(c,k+1), j, r), \varepsilon, \widetilde{\gamma_k} \cdot (\ell_k, \mathsf{g}^{<}(c,k)) \cdot \tau_2(c,k), 1, 2r-1)$, and $u_i \cdot \rho$ a partial play ending in $v_i$. By definition, $\tau_2(c, k+1) = \widetilde{\gamma_k} \cdot (\ell_k, \mathsf{g}^{<}(c,k)) \cdot \tau_2(c,k)$, then $v_i = (\mathrm{np}(s, \mathsf{g}^{>}(c,k), \mathsf{g}^{<}(c,k+1), j, r), \varepsilon, \tau_2(c,k+1), 1, 2r-1)$, and $\mathrm{loop}_1^{\mathrm{np}}(v_i) = (?(s, \ell_{\mathsf{in}}, \checkmark, \mathsf{g}^{<}(c,k+1), j, r), \varepsilon, \tau_2(c,k+1), 1, 2r-1)$, by definition (4) describing $\mathrm{loop}_1^{\mathrm{np}}$. Hence, $u_i \cdot \mathrm{loop}_1^{\mathrm{np}}(u_i) = u_i \cdot \rho \cdot \mathrm{loop}_1^{\mathrm{np}}(v_i)$ is indeed a partial play, because transition 7 from states from $\mathfrak{I}_{\mathrm{NP}}$ allows to go from $v_i$ to $\mathrm{loop}_1^{\mathrm{np}}(v_i)$.

Moreover,

- $\ell_{\mathsf{in}} = \ell_{k+1}$ as every new process starts in state $\ell_{\mathsf{in}}$,
- $\mathsf{g}^{>}(c, k+1) = \checkmark$ since $k+1$ is a new process so there are no processes above it,
- $\varepsilon = \tau_1(c, k+1)$ as every new process starts with an empty stack.

Then, $u_i \cdot \mathrm{loop}_1^{\mathrm{np}}(u_i)$ ends indeed in

$$(?(s, \ell_{k+1}, \mathsf{g}^{>}(c,k+1), \mathsf{g}^{<}(c,k+1), pl(s), r), \tau_1(c,k+1), \tau_2(c,k+1), 1, 2r-1).$$

2) If $p' < k+1$, then $\tau_1(c, p+i) = \tau_1(c, p'-1) = \gamma_{p'-1} \cdot (\ell_{p'}, \mathsf{g}^{>}(c,p')) \cdot \tau_1(c,p')$. In that case, $u_i = (\mathrm{np}(s, \mathsf{g}^{>}(c,p'-1), \mathsf{g}^{<}(c,p'), j, r), \gamma_{p'-1} \cdot (\ell_{p'}, \mathsf{g}^{>}(c,p')) \cdot \tau_1(c,p'), (\ell_{p'-1}, \mathsf{g}^{<}(c,p'-1)) \cdot \tau_2(c,p'-1), 1, 2r-1)$.

By Lemma 4.7, $u_i \cdot \mathrm{loop}_1^{\mathrm{np}}(u_i) = u_i \cdot \rho \cdot \mathrm{loop}_1^{\mathrm{np}}(v_i)$, with $u_i \cdot \rho$ a partial play ending in $v_i$ and

$$\begin{aligned} v_i &= (\mathrm{np}(s, \mathsf{g}^{>}(c,p'-1), \mathsf{g}^{<}(c,p'), j, r), (\ell_{p'}, \mathsf{g}^{>}(c,p')) \cdot \tau_1(c,p'), \\ &\quad \widetilde{\gamma_{p'-1}} \cdot (\ell_{p'-1}, \mathsf{g}^{<}(c,p'-1)) \cdot \tau_2(c,p'-1), 1, 2r-1) \\ &= (\mathrm{np}(s, \mathsf{g}^{>}(c,p'-1), \mathsf{g}^{<}(c,p'), j, r), (\ell_{p'}, \mathsf{g}^{>}(c,p')) \cdot \tau_1(c,p'), \\ &\quad \tau_2(c,p'), 1, 2r-1). \end{aligned}$$

By definition,

$$\mathrm{loop}_1^{\mathrm{np}}(v_i) = (?(s, \ell_{p'}, \mathsf{g}^{>}(c,p'), \mathsf{g}^{<}(c,p'), j, r), \tau_1(c,p'), \tau_2(c,p'), 1, 2r-1).$$

and

$u_i \cdot \mathrm{loop}_1^{\mathrm{np}}(u_i)$ is a partial play as announced.

Now suppose the property holds for some $0 < i \leq p - p' - 1$, and show that it remains true for $i-1$. Necessarily, since there is at least one process between $p+(i-1)$ and $p'$, $\tau_1(c, p+(i-1)) = \gamma_{p+(i-1)} \cdot (\ell_{p+i}, \mathsf{g}^{>}(c, p+i)) \cdot \tau_1(c, p+i)$. Then

$$u_{i-1} \cdot \mathrm{loop}_{p'-p-(i-1)}^{\mathrm{np}}(u_{i-1}) = u_{i-1} \cdot \rho \cdot \mathrm{loop}_{p'-p-(i-1)}^{\mathrm{np}}(v_{i-1})$$

with $u_{i-1} \cdot \rho$ a partial play ending in

$$\begin{aligned} v_{i-1} = (\mathrm{np}(s, \mathsf{g}^{>}(c, p+i-1), \mathsf{g}^{<}(c, p+i), j, r), (\ell_{p+i}, \mathsf{g}^{>}(c, p+i)) \cdot \tau_1(c, p+i), \\ \tau_2(c, p+i), 1, 2r-1). \end{aligned}$$

Moreover, by definition,

$$\begin{aligned} &\mathrm{loop}_{p'-p-(i-1)}^{\mathrm{np}}(v_{i-1}) = \\ &(\mathrm{np}^{\ell_{p+i}}(s, \mathsf{g}^{>}(c, p+i), \mathsf{g}^{<}(c, p+i), j, r)), \tau_1(c, p+i), \tau_2(c, p+i), 1, 2r-1) \cdot u_i \\ &\cdot \mathrm{loop}_{p'-p-i}^{\mathrm{np}}(u_i) \end{aligned}$$

Using the induction hypothesis and the definitions of the transitions, we obtain that $u_{i-1} \cdot \mathrm{loop}^{\mathrm{np}}_{p'-p-(i-1)}(u_{i-1})$ is indeed a partial play ending in

$$(?(s, \ell_{p'}, \mathsf{g}^{>}(c, p'), \mathsf{g}^{<}(c, p'), pl(s), r), \tau_1(c, p'), \tau_2(c, p'), 1, 2r - 1).$$

The proof for $\mathrm{loop}^{\mathrm{nr}}$ is almost symmetrical and so it will be omitted. □

We are now ready to establish that:

**Lemma 4.9** *If $\rho$ is a play of $\mathcal{G}^{\mathcal{P}}_{B\text{-}rb}$, then $\pi(\rho)$ is a play of $\mathcal{G}_{\mathcal{M}}$.*

***Proof*** We show it by induction on the size of $\rho$. If $\rho = ((s_{\mathsf{in}}), 0, 1)$, then $\pi(\rho) = (s'_{\mathsf{in}}, \varepsilon, \varepsilon, 1, 1)$ which is a play of $\mathcal{G}_{\mathcal{M}}$. If $\rho = ((s_{\mathsf{in}}), 0, 1) \cdot ((s, (\ell, \gamma_1)), 1, 1)$, then $\pi(((s_{\mathsf{in}}), 0, 1) \cdot ((s, (\ell, \gamma_1)), 1, 1)) = (s'_{\mathsf{in}}, \varepsilon, \varepsilon, 0, 1) \cdot ((s, \ell, \checkmark, \checkmark, pl(s), 1), \gamma_1, \varepsilon, 1, 1)$ which is also a play of $\mathcal{G}_{\mathcal{M}}$. Assume now that $\rho$ is a finite play ending in $u = (c, p, r)$, with $c = (s, (\ell_1, \gamma_1) \dots (\ell_k, \gamma_k))$ and suppose that $\pi(\rho)$ is a play of $\mathcal{G}_{\mathcal{M}}$. Let $u' = (c', p', r')$ with $c = (s', (\ell'_1, \gamma'_1) \dots (\ell'_k, \gamma'_k))$ be such that $\rho \cdot u'$ is a play of $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$, which means that $(c, c') \in E_{p'}$. We have then $(s, \ell_p) \xrightarrow{(op, A)} (s', \ell'_p)$, and $\ell_i = \ell'_i$ and $\gamma_i = \gamma'_i$ for all $i \neq p'$.

If $u \in \mathcal{F}$, then $\pi(\rho \cdot u') = \pi(\rho)$ and by induction hypothesis, $\pi(\rho \cdot u')$ is a play of $\mathcal{G}_{\mathcal{M}}$. Otherwise, $\pi(\rho \cdot u')$ starts with $\pi(\rho) \cdot \mathrm{next}^{p'}_p(\pi(u)) \cdot \pi(u')$ and $\pi(\rho)$ ends in $\pi(u)$. Let

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

and

$$\pi(u') = ((s', \ell'_{p'}, \mathsf{g}^{>}(c', p'), \mathsf{g}^{<}(c', p'), pl(s'), r'), \tau_1(c', p'), \tau_2(c', p'), 1, 2r' - 1).$$

There are three cases to consider: either $p = p'$, $p < p'$, or $p > p'$.

- If $p = p'$, by definition, $\mathrm{next}^{p'}_p(\pi(u)) = \varepsilon$. Moreover, $(s, \ell_p) \xrightarrow{(op, A)} (s', \ell'_p)$, hence we have a transition

  $$(s, \ell_p, \mathsf{g}^{>}(c, p), \mathsf{g}^{<}(c, p), pl(s), r) \xrightarrow{(op, 1, A)} (s', \ell'_p, \mathsf{g}^{>}(c, p), \mathsf{g}^{<}(c, p), pl(s'), r).$$

  Observe that in that case $\mathsf{g}^{>}(c', p') = \mathsf{g}^{>}(c, p)$ and $\mathsf{g}^{<}(c', p') = \mathsf{g}^{<}(c, p)$, hence $\pi(u')$ is indeed a successor of $\pi(u)$ in $\mathcal{G}_{\mathcal{M}}$.
- If $p < p'$, then $r = r'$ and $\mathrm{next}^{p'}_p(\pi(u))$ starts with

  $$(\mathrm{np}(s, \mathsf{g}^{>}(c, p), \mathsf{upd}(\mathsf{g}^{<}(c, p), \ell_p), pl(s), r), \tau_1(c, p), (\ell_p, \mathsf{g}^{<}(c, p)) \cdot \tau_2(c, p), 1, 2r - 1)$$

  which is a successor of $\pi(u)$. By Lemma 4.8, $\mathrm{next}^{p'}_p(\pi(u))$ is a play of $\mathcal{G}_{\mathcal{M}}$ that ends in

  $$v = (?(s, \ell_{p'}, \mathsf{g}^{>}(c, p'), \mathsf{g}^{<}(c, p'), pl(s), r), \tau_1(c, p'), \tau_2(c, p'), 1, 2r - 1).$$

  Observe that by definition, $\ell'_i = \ell_i$ for all $i \neq p'$, so $\mathsf{g}^{>}(c, p') = \mathsf{g}^{>}(c', p')$ and $\mathsf{g}^{<}(c, p') = \mathsf{g}^{<}(c', p')$, and by definition $\tau_2(c, p') = \tau_2(c', p')$. Then $\pi(u')$ is indeed a successor of $v$.
- If $p > p'$, then $r' = r + 1$ and $\mathrm{next}^{p'}_p(\pi(u))$ starts with

  $$(\mathrm{nr}(s, \mathsf{upd}(\mathsf{g}^{>}(c, p), \ell_p), \mathsf{g}^{<}(c, p), pl(s), r), (\ell_p, \mathsf{g}^{>}(c, p)) \cdot \tau_1(c, p), \tau_2(c, p), 2, 2r)$$

  which is also a successor of $\pi(u)$. Again by Lemma 4.8, $\mathrm{next}^{p'}_p(\pi(u))$ is a play of $\mathcal{G}_{\mathcal{M}}$ that ends in

  $$v = (?(s, \ell_{p'}, \mathsf{g}^{>}(c, p'), \mathsf{g}^{<}(c, p'), pl(s), r + 1), \tau_1(c, p'), \tau_2(c, p'), 1, 2r + 1).$$

Again, one can check that $\pi(u')$ is a successor of $v$.

In any case, $\pi(\rho) \cdot \text{next}_p^{p'}(\pi(u)) \cdot \pi(u')$ is a play of $\mathcal{G}_{\mathcal{M}}$. Then there are two special cases to consider:

- If now $u' \in \mathcal{F}$, then by construction, $\mathsf{g}^{>}(c', p') = \mathsf{g}^{<}(c', p') = \checkmark$, $s' \in F_{\mathsf{glob}}$, $\ell_{p'} \in F_{\mathsf{loc}}$, hence $(s', \ell'_{p'}, \mathsf{g}^{>}(c', p'), \mathsf{g}^{<}(c', p'), pl(s'), r') \in \mathfrak{F}$ and thus $(\text{win}_0, \tau_1(c', p'), \tau_2(c', p'), 1, 2r' - 1)$ is a successor of $\pi(u')$ and $\pi(\rho \cdot u')$ is a play of $\mathcal{G}_{\mathcal{M}}$.
- Otherwise, if $u'$ has no successor, $\pi(\rho) \cdot \text{next}_p^{p'}(\pi(u)) \cdot \pi(u') \cdot \text{Ch}_0$ is a play that ends in a state $v \in V_0' \setminus \mathcal{F}$, hence $(\text{win}_1, \tau_1(c', p'), \tau_2(c', p'), 1, 2r' - 1)$ is a successor of $v$ and $\pi(\rho \cdot u')$ is a play of $\mathcal{G}_{\mathcal{M}}$.

So for any finite play $\rho$ in $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$, $\pi(\rho)$ is a play of $\mathcal{G}_{\mathcal{M}}$. If $\rho$ is an infinite play, then $\pi(\rho)$ is also a play of $\mathcal{G}_{\mathcal{M}}$, otherwise we can find a finite prefix $\rho'$ of $\rho$ such that $\pi(\rho')$ is not a play of $\mathcal{G}_{\mathcal{M}}$. □

We describe now how to transform a play of $\mathcal{G}_{\mathcal{M}}$ into a play of $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$, through the mapping $\hat{\pi}$. We let $V_{\text{sim}} = \{(s_{\mathcal{M}}, \overline{\gamma_1}, \overline{\gamma_2}, st, ph) \mid s_M \in S_{\text{sim}}\}$, $V_? = \{(s_{\mathcal{M}}, \overline{\gamma_1}, \overline{\gamma_2}, st, ph) \mid s_M \in \mathfrak{I}_?\}$ and $V_{\text{win}} = V_{\text{win}0} \cup V_{\text{win}1}$, with $V_{\text{win}i} = \{(\text{win}_i, \overline{\gamma_1}, \overline{\gamma_2}, st, ph)\}$.

Let $v = (s_{\mathcal{M}}, \overline{\gamma_1}, \overline{\gamma_2}, 1, ph) \in V_{\text{sim}} \cup V_?$ be a node of $\mathcal{G}_{\mathcal{M}}$, with

$$\begin{aligned} s_M &= (s, \ell, \mathsf{f}_1, \mathsf{f}_2, j, r) \text{ or } s_M = ?(s, \ell, \mathsf{f}_1, \mathsf{f}_2, j, r) \\ \overline{\gamma_1} &= \gamma_p \cdot (\ell_{p+1}, \mathsf{g}_{p+1}) \cdot \gamma_{p+1} \cdots (\ell_k, \mathsf{g}_k) \cdot \gamma_k \\ \overline{\gamma_2} &= \widetilde{\gamma}_{p-1} \cdot (\ell_{p-1}, \mathsf{g}_{p-1}) \cdots \widetilde{\gamma}_1 \cdot (\ell_1, \mathsf{g}_1) \end{aligned}$$

for some $\ell_1, \ldots, \ell_k \in L$, $\gamma_1, \ldots, \gamma_k \in \Gamma^*$, and $\mathsf{g}_1, \ldots, \mathsf{g}_k \in \{\checkmark, \text{✗}\}$, where $\widetilde{\gamma}$ denotes the mirror word of $\gamma$.

We let $\text{size}(\overline{\gamma_1}) = k - p$ and $\text{size}(\overline{\gamma_2}) = p - 1$ be the number of elements from $L \times \{\checkmark, \text{✗}\}$ in each stack.

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

1. For all $1 \leq i \leq p-1$, $\mathsf{g}_i = \checkmark$ if and only if for all $1 \leq p' < i$, $\ell_{p'} \in F_{\mathsf{loc}}$,
2. For all $p+1 \leq i \leq k$, $\mathsf{g}_i = \checkmark$ if and only if for all $i < p' \leq k$, $\ell_{p'} \in F_{\mathsf{loc}}$,
3. $\mathsf{f}_1 = \checkmark$ if and only if for all $p < p' \leq k$, $\ell_{p'} \in F_{\mathsf{loc}}$,
4. $\mathsf{f}_2 = \checkmark$ if and only if for all $1 \leq p' < p$, $\ell_{p'} \in F_{\mathsf{loc}}$,
5. if $\ell = \ell_{\text{in}}$, then $\overline{\gamma_1} = \varepsilon$,
6. $ph = 2r - 1$.

In that case, we define

$$\hat{\pi}(v) = ((s, (\ell_1, \gamma_1) \ldots (\ell_k, \gamma_k)), p, r).$$

Note that $p = \text{size}(\overline{\gamma_2}) + 1$.

If $v$ additionnally meets the following requirement:

7. $j = pl(s)$

we say that $v$ is $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$-*real*.

***Remark 4.10*** It is easy to see that for any $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$-real node $v \in V_{\text{sim}}$, then $\pi(\hat{\pi}(v)) = v$, and for any node $u \in V^B$ in $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$, $\hat{\pi}(\pi(u)) = u$. Moreover, $v \in \mathfrak{F}$ if and only if $\hat{\pi}(v) \in \mathcal{F}$.

We also let

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

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

$$\hat{\pi}(\hat{\rho} \cdot v) = \begin{cases} \hat{\pi}(\hat{\rho}) \cdot \hat{\pi}(v) \text{ if } v \text{ is a } \mathcal{G}^{\mathcal{P}}_{B\text{-rb}}\text{-real node in } V_{\mathrm{sim}}, \\ \hat{\pi}(\hat{\rho}) \text{ otherwise.} \end{cases}$$

To show that the mapping $\hat{\pi}$ builds a correct play of $\mathcal{G}_{\mathcal{M}}$, we first show that a play between a node $u \in V_{\mathrm{sim}}$ and a node in $V_?$ is necessarily of the form $\mathrm{next}^{p}_{p'}(u)$.

**Lemma 4.11** *Let $\hat{\rho} = \hat{\rho}_1 \cdot u \cdot \hat{\rho}_2 \cdot v$ be a finite play of $\mathcal{G}_{\mathcal{M}}$ such that $u$ is the last state in $V_{sim}$ and $v \in V_?$. Then, there exists two distinct $p$ and $p'$ such that $\hat{\rho}_2 \cdot v = next^{p'}_{p}(u)$. Moreover, if $u$ is well-defined (resp. $\mathcal{G}^{\mathcal{P}}_{B\text{-}rb}$-real) and such that $\hat{\pi}(u) = (c, p, r)$, so is $v$ and $\hat{\pi}(v) = (c, p', r')$ with $r' = r$ if $p < p'$ and $r' = r + 1$ if $p > p'$.*

***Proof*** Let $u = ((s, \ell, \mathsf{f}_1, \mathsf{f}_2, j, r), \overline{\gamma_1}, \overline{\gamma_2}, 1, ph) \in V_{\mathrm{sim}}$, and $p = \mathrm{size}(\overline{\gamma_2}) + 1$, and let $v = (?(s^v, \ell^v, \mathsf{f}^v_1, \mathsf{f}^v_2, j^v, r^v), \overline{\gamma^v_1}, \overline{\gamma^v_2}, st, ph^v)$. By the transition relation of $\mathcal{G}_{\mathcal{M}}$, in order to reach $V_?$ one must go through states either in $\mathfrak{I}_{\mathrm{NP}}$ or $\mathfrak{I}_{\mathrm{NR}}$. Since by hypothesis $\hat{\rho}_2$ contains no node in $V_{\mathrm{sim}}$, all of $\hat{\rho}_2$ must occur in either $\mathfrak{I}_{\mathrm{NP}}$ or $\mathfrak{I}_{\mathrm{NR}}$.

Suppose the former is true (the other case will, again, be extremely similar). Necessarily, the first node in $\hat{\rho}_2$ is $w = (\mathrm{np}(s, \mathsf{f}_1, \mathsf{upd}(\mathsf{f}_2, \ell), j, r), \overline{\gamma_1}, (\ell, \mathsf{f}_2) \cdot \overline{\gamma_2}, 1, ph)$, as this is the only transition from $u$ that goes in $\mathfrak{I}_{\mathrm{NP}}$ (a transition of type 1 in the description of $\mathfrak{I}_{\mathrm{NP}}$).

Let us show that for all (not strict) suffixes $\hat{\rho}'$ of $\hat{\rho}_2$ such that $\hat{\rho}'$ starts in $u' = (\mathrm{np}(s, \mathsf{f}'_1, \mathsf{f}'_2, j, r), \overline{\gamma_1}', \overline{\gamma_2}', 1, ph)$ for some $\mathsf{f}'_1, \mathsf{f}'_2, \overline{\gamma_1}', \overline{\gamma_2}'$ that meet the requirements 1-6 of the definition of well-defined, $(i)$ there exists a $k \geq 1$ such that $\hat{\rho}' \cdot v = u' \cdot \mathrm{loop}^{\mathrm{np}}_k(u')$, $(ii)$ $v$ is well-defined, and $(iii)$ $\widetilde{\overline{\gamma_2}'} \cdot \overline{\gamma_1}' = \widetilde{\overline{\gamma^v_2}} \cdot \gamma$, with $\gamma = (\ell^v, \mathsf{f}^v_1) \cdot \overline{\gamma^v_1}$, if $\ell^v \neq \ell_{\mathsf{in}}$, and $\gamma = \varepsilon$ otherwise. Suppose that $\hat{\rho}' = u'$, i.e. the next node is $v$. Then either a transition of type 6 or 7 has been used to go from $u'$ to $v$. However, the kind of transition depends only on $\overline{\gamma_1}'$. If $\overline{\gamma_1}' = (\ell, \mathsf{g}) \cdot \overline{\gamma_1}''$ then only a transition of type 6 can be used, and $\hat{\rho}' \cdot v = u' \cdot \mathrm{loop}^{\mathrm{np}}_1(u')$. Moreover, $\ell^v = \ell$, $\mathsf{f}^v_1 = \mathsf{g}$, $\mathsf{f}^v_2 = \mathsf{f}'_2$, $\overline{\gamma^v_1} = \overline{\gamma_1}''$, and $\overline{\gamma^v_2} = \overline{\gamma_2}'$. Immediately, $v$ is then well-defined, and $\widetilde{\overline{\gamma_2}'} \cdot \overline{\gamma_1}' = \widetilde{\overline{\gamma^v_2}} \cdot (\ell^v, \mathsf{f}^v_1) \cdot \overline{\gamma^v_1}$. If $\overline{\gamma_1}' = \varepsilon$, only the transition of type 7 can be used and we again have that $\hat{\rho}' \cdot v = u' \cdot \mathrm{loop}^{\mathrm{np}}_1(u')$. Moreover, $\ell^v = \ell_{\mathsf{in}}$, $\mathsf{f}^v_1 = \checkmark$, $\mathsf{f}^v_2 = \mathsf{f}'_2$, $\overline{\gamma^v_1} = \overline{\gamma_1}' = \varepsilon$, and $\overline{\gamma^v_2} = \overline{\gamma_2}'$. Immediately, $v$ is then well-defined, and $\widetilde{\overline{\gamma_2}'} \cdot \overline{\gamma_1}' = \widetilde{\overline{\gamma^v_2}}$.

Now suppose that $|\hat{\rho}'| > 1$ and that it starts with $u'$. Then $\overline{\gamma_1}'$ cannot be empty, otherwise the only transition available would directly lead to $v$ and $\hat{\rho}'$ would not be longer than 1. Therefore there are two cases to consider:

- If $\overline{\gamma_1}' = A \cdot \overline{\gamma_1}''$, then the only available transitions (type 2) give that $\hat{\rho}' = u' \cdot (\mathrm{np}^A(s, \mathsf{f}'_1, \mathsf{f}'_2, j, r), \overline{\gamma_1}'', \overline{\gamma_2}', 1, ph) \cdot \hat{\rho}''$ with $\hat{\rho}''$ starting with $u'' = (\mathrm{np}(s, \mathsf{f}'_1, \mathsf{f}'_2, j, r), \overline{\gamma_1}'', A \cdot \overline{\gamma_2}', 1, ph)$. By induction hypothesis, there exists some $k \geq 1$ such that $\hat{\rho}'' \cdot v = u'' \cdot \mathrm{loop}^{\mathrm{np}}_k(u'')$. Then, by definition of $\mathrm{loop}^{\mathrm{np}}_k$, $\hat{\rho}' \cdot v = u' \cdot \mathrm{loop}^{\mathrm{np}}_k(u')$. Neither $\mathsf{f}'_1$ nor $\mathsf{f}'_2$ has changed, nor any of the $\mathsf{g}_i$, so if $u'$ meets all the requirements 1 to 6, so does $u''$, and by induction hypothesis, $v$ is well-defined. Moreover, $\widetilde{\overline{\gamma_2}'} \cdot \overline{\gamma_1}' = \widetilde{\overline{\gamma_2}'} \cdot A \cdot \overline{\gamma_1}'' = A \cdot \widetilde{\overline{\gamma_2}'} \cdot \overline{\gamma_1}'' = \widetilde{\overline{\gamma^v_2}} \cdot \gamma$ by induction hypothesis.
- If $\overline{\gamma_1}' = (\ell, \mathsf{g}) \cdot \overline{\gamma_1}''$, then, since $|\hat{\rho}'| > 1$, the only possible transition is of type 4. Hence, $\hat{\rho}' = u' \cdot (\mathrm{np}^{\ell}(s, \mathsf{g}, \mathsf{f}_2, j, r), \overline{\gamma_1}'', \overline{\gamma_2}', 1, ph) \cdot \hat{\rho}''$ with $\hat{\rho}''$ starting with $u'' = (\mathrm{np}(s, \mathsf{g}, \mathsf{upd}(\mathsf{f}'_2, \ell), j, r), \overline{\gamma_1}'', (\ell, \mathsf{f}'_2) \cdot \overline{\gamma_2}', 1, ph)$. By induction hypothesis, $\hat{\rho}'' \cdot v =$

$u'' \cdot \text{loop}_k^{\text{np}}(u'')$ for some $k \geq 1$, so $\hat{\rho}' \cdot v = u' \cdot (\text{np}^{\ell}(s, \mathsf{g}, \mathsf{f}_2, j, r), \overline{\gamma_1}'', \overline{\gamma_2}', 1, ph) \cdot u'' \cdot \text{loop}_k^{\text{np}}(u'') = u' \cdot \text{loop}_{k+1}^{\text{np}}(u')$. Moreover, if $u'$ meets the requirements 1 to 5 of being well-defined, so is $u''$, by construction, and by induction hypothesis $v$ is well-defined.

Finally, $\overset{\sim}{\overline{\gamma_2'}} \cdot \overline{\gamma_1'} = \overset{\sim}{\overline{\gamma_2'}} \cdot (\ell, \mathsf{g}) \cdot \overline{\gamma_1''} = \overset{\sim}{(\ell, \mathsf{g})} \cdot \overline{\gamma_2'} \cdot \overline{\gamma_1''} = \overset{\sim}{\overline{\gamma_2^v}} \cdot \gamma$ by induction hypothesis.

Therefore this property holds for $\hat{\rho}_2$, i.e. $\hat{\rho}_2 \cdot v = w \cdot \text{loop}_k^{\text{np}}(w)$ for some $k \geq 1$, i.e. by definition $\hat{\rho}_2 \cdot v = \text{next}_p^{p+k}(u)$. Moreover, since it is easy to show that if $u$ is well-defined, $w$ meets the requirements 1 to 6 of being well-defined, then $v$ is well-defined. If $u$ is $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$-real, since $j$ and $s$ does not change during this part of the run, so is $v$.

Now if $\hat{\pi}(u) = (c, p, r)$, then since $\overset{\sim}{\overline{\gamma_2}} \cdot (\ell, \mathsf{f}_2) \cdot \overline{\gamma_1} = \overset{\sim}{\overline{\gamma_2^v}} \cdot \gamma$, then necessarily, $\hat{\pi}(v) = (c, p', r')$ since the sequence of symbols from $\Gamma$ and $L$ stayed unchanged. Moreover, $\text{size}(\overline{\gamma_2^v}) > \text{size}(\overline{\gamma_2})$ then $p' > p$, and $r' = r$ because $ph$ has not changed during this part of the play, and $u$ and $v$ are well-defined.

If all of $\hat{\rho}_2$ must occur in $\mathfrak{I}_{\text{NR}}$, the proof is similar, but $p' < p$ and $r' = r + 1$. □

**Corollary 4.12** *Every reachable node $v \in V_{sim} \cup V_?$ is well-defined.*

***Proof*** We prove it by induction on the play $\hat{\rho} \cdot v$. If $\hat{\rho} = (s'_{\text{in}}, \varepsilon, \varepsilon, 0, 1)$ then by construction, $v$ is well-defined. Otherwise, assume that for any finite prefix of $\hat{\rho}$ ending in a node of $V_{\text{sim}} \cup V_?$, the property holds true. Let $\hat{\rho} = \hat{\rho}_1 \cdot v_1$ and $v$ be a successor of $v_1$.

- If $v \in V_{\text{sim}}$, then by construction, $v_1 \in V_{\text{sim}} \cup V_?$. In both cases, the transition involves no phase change, no modification of $\mathsf{f}_1$, $\mathsf{f}_2$, nor of any $(\ell_i, \mathsf{g}_i)$, so if the property holds for $v_1$, it holds for $v$.
- If $v \in V_?$, then $\hat{\rho} \cdot v = \hat{\rho}_1 \cdot u \cdot \hat{\rho}_2 \cdot v$ with $u$ the last node in $V_{\text{sim}}$. By Lemma 4.11, if $u$ is well-defined, so is $v$. □

**Lemma 4.13** *If $\hat{\rho}$ is a play of $\mathcal{G}_{\mathcal{M}}$ then $\hat{\pi}(\hat{\rho})$ is a play of $\mathcal{G}^{\mathcal{P}}_{B\text{-}rb}$.*

***Proof*** Let $\hat{\rho}$ be a finite prefix of a play of $\mathcal{G}_{\mathcal{M}}$. If $\hat{\rho} = (s'_{\text{in}}, \varepsilon, \varepsilon, 0, 1)$, then $\hat{\pi}(\hat{\rho})$ is indeed a play of $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$, by construction. Otherwise, assume that $\hat{\pi}(\hat{\rho})$ is a play of $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$. Let $v$ be such that $\hat{\rho} \cdot v$ is a play of $\mathcal{G}_{\mathcal{M}}$. If $v \notin V_{\text{sim}}$, or if $v$ is not $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$-real, then $\hat{\pi}(\hat{\rho} \cdot v) = \hat{\pi}(\hat{\rho})$ and it is then a play of $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$. Otherwise, let $v = ((s, \ell, \mathsf{f}_1, \mathsf{f}_2, pl(s), r), \overline{\gamma_1}, \overline{\gamma_2}, 1, ph)$. Since it is in $V_{\text{sim}}$, by definition of the transition relation of $\mathcal{G}_{\mathcal{M}}$, $\hat{\rho}$ necessarily ends in $u \in V_{\text{sim}} \cup V_?$. Let $\hat{\pi}(v) = ((s, (\ell_1, \gamma_1) \dots (\ell_k, \gamma_k)), p, r)$, with $\ell = \ell_p$.

If $u \in V_{\text{sim}}$, then $u = ((s', \ell', \mathsf{f}_1, \mathsf{f}_2, j, r), \overline{\gamma_1'}, \overline{\gamma_2}, 1, ph)$. There exists a transition $(s', \ell', \mathsf{f}_1, \mathsf{f}_2, j, r) \xrightarrow{(op,1,A)} (s, \ell, \mathsf{f}_1, \mathsf{f}_2, pl(s), r)$. By definition, there is a transition $(s', \ell') \xrightarrow{op,A} (s, \ell)$ in $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$. Then, if $op = \mathsf{push}$, $\overline{\gamma_1} = A \cdot \overline{\gamma_1'}$, and if $op = \mathsf{pop}$, then $\overline{\gamma_1'} = A \cdot \overline{\gamma_1}$. By Corollary 4.12, $u$ is well-defined, and $\hat{\pi}(u) = ((s', (\ell'_1, \gamma'_1), \dots, (\ell'_k, \gamma'_k), p, r)$ with $\ell'_i = \ell_i$, $\gamma'_i = \gamma_i$, for all $i \neq p$. Moreover, if $op = \mathsf{push}$, $\gamma_p = A \cdot \gamma'_p$, and if $op = \mathsf{pop}$, then $\gamma'_p = A \cdot \gamma_p$. Hence, $\hat{\pi}(v)$ is indeed a successor of $\hat{\pi}(u)$ in $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$. If $u$ is $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$-real, then if we let $\hat{\rho} = \hat{\rho}' \cdot u$, then $\hat{\pi}(\hat{\rho} \cdot v) = \hat{\pi}(\hat{\rho}') \cdot \hat{\pi}(u) \cdot \hat{\pi}(v)$, which is a play of $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$. Otherwise, it is because $j = 0$ and $pl(s') = 1$. In that case, the predecessor of $u$ is necessarily $u' = ((s', \ell', \mathsf{f}_1, \mathsf{f}_2, 1, r), \overline{\gamma_1'}, \overline{\gamma_2}, 1, ph)$, which is $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$-real, and $\hat{\pi}(u') = \hat{\pi}(u)$. Hence, as above, $\hat{\pi}(v)$ is a successor or $\hat{\pi}(u')$. Moreover, if we let $\hat{\rho} = \hat{\rho}' \cdot u' \cdot u$, we have $\hat{\pi}(\hat{\rho} \cdot v) = \hat{\pi}(\hat{\rho}') \cdot \hat{\pi}(u') \cdot \hat{\pi}(v)$, which is then a play of $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$.

In case $u \in V_?$, $u = (?(s', \ell', \mathsf{f}_1, \mathsf{f}_2, j, r), \overline{\gamma_1'}, \overline{\gamma_2}, 1, ph)$ and there exists a transition $(s', \ell') \xrightarrow{op,A} (s, \ell)$. Then again, if $op = \mathsf{push}$, $\overline{\gamma_1'} = A \cdot \overline{\gamma_1}$, then $\overline{\gamma_p'} = A \cdot \overline{\gamma_p}$, and

if $op = \mathsf{pop}$, $\overline{\gamma_1} = A \cdot \overline{\gamma_1'}$, then $\overline{\gamma_p} = A \cdot \overline{\gamma_p'}$. Let $\hat{\rho} = \hat{\rho}_1 \cdot u_1 \cdot \hat{\rho}_2 \cdot u$ with $u_1$ the last node in $V_{\text{sim}}$. By construction, $\hat{\pi}(u) = ((s', (\ell_1', \gamma_1'), \ldots, (\ell_k', \gamma_k'), p, r)$ with $\ell_i' = \ell_i$, $\gamma_i' = \gamma_i$, for all $i \neq p$. Moreover, by Lemma 4.11, there exists $p' \neq p$, and $r'$ such that $\hat{\pi}(u_1) = ((s', (\ell_1', \gamma_1'), \ldots, (\ell_k', \gamma_k'), p', r')$ and $\hat{\pi}(v)$ is also a successor of $\hat{\pi}(u_1)$. If $u_1$ is $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$-real, $\hat{\pi}(\hat{\rho} \cdot v) = \hat{\pi}(\hat{\rho}_1) \cdot \hat{\pi}(u_1) \cdot \hat{\pi}(v)$. Otherwise, as above, the predecessor of $u_1$ is necessarily the $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$-real node $u'$, and $\hat{\pi}(u') = \hat{\pi}(u_1)$. Then, $\hat{\pi}(\hat{\rho} \cdot v) = \hat{\pi}(\hat{\rho}_1 \cdot u_1 \cdot \hat{\rho}_2 \cdot u \cdot v) = \hat{\pi}(\hat{\rho}_1' \cdot u' \cdot u_1 \cdot \hat{\rho}_2 \cdot u \cdot v) = \hat{\pi}(\hat{\rho}_1') \cdot \hat{\pi}(u') \cdot \hat{\pi}(v)$, which proves the result. □

From the preceding results, we deduce the following correspondence between the plays of the two games:

**Corollary 4.14** *Let the run $\hat{\rho} \cdot v$ be a run ending in a $\mathcal{G}^{\mathcal{P}}_{B\text{-}rb}$-real node in $V_{sim}$. If $v \notin \mathfrak{F}$, and if $\hat{\pi}(v)$ has a successor in $\mathcal{G}^{\mathcal{P}}_{B\text{-}rb}$, then $\pi(\hat{\pi}(\hat{\rho} \cdot v)) = \hat{\rho} \cdot v$.*

***Proof*** We show it by induction. If $\hat{\rho}$ is reduced to the initial configuration of the game, then the result comes directly from the definitions. Consider now a play $\hat{\rho}_0 \cdot v_0 \cdot \hat{\rho}_1 \cdot v_1$ with $v_0, v_1$ $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$-real nodes in $V_{\text{sim}}$, and $\hat{\rho}_1$ containing no $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$-real node in $V_{\text{sim}}$. Assume also that $v_1 \notin \mathfrak{F}$. By construction of the game, $v_0 \notin \mathfrak{F}$, otherwise the only continuation of $\hat{\rho}_0 \cdot v_0$ would be in $V_{\text{win}}$. Also, $\hat{\pi}(\hat{\rho}_0 \cdot v_0 \cdot \hat{\rho}_1 \cdot v_1) = \hat{\pi}(\hat{\rho}_0) \cdot \hat{\pi}(v_0) \cdot \hat{\pi}(v_1)$. Since it is a play of $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$ by Lemma 4.13, then $\hat{\pi}(v_0)$ has a successor in $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$. Let $(c, p, r) = \hat{\pi}(v_0)$ and $(c', p', r') = \hat{\pi}(v_1)$. Then $\pi(\hat{\pi}(\hat{\rho}_0 \cdot v_0 \cdot \hat{\rho}_1 \cdot v_1)) = \pi(\hat{\pi}(\hat{\rho}_0 \cdot v_0) \cdot \hat{\pi}(v_1)) = \pi(\hat{\pi}(\hat{\rho}_0 \cdot v_0)) \cdot \text{next}_p^{p'}(v_0) \cdot \pi(\hat{\pi}(v_1))$. By induction hypothesis, $\pi(\hat{\pi}(\hat{\rho}_0 \cdot v_0)) = \hat{\rho}_0 \cdot v_0$. Moreover, for $\hat{\rho}_0 \cdot v_0 \cdot \hat{\rho}_1 \cdot v_1$ to be a play, $\hat{\rho}_1$ is either $\varepsilon$, or ends in $V_?$. Then, by Lemma 4.11, $\hat{\rho}_1 = \text{next}_p^{p'}(v_0)$. Hence, $\pi(\hat{\pi}(\hat{\rho}_0 \cdot v_0 \cdot \hat{\rho}_1 \cdot v_1)) = \hat{\rho}_0 \cdot v_0 \cdot \hat{\rho}_1 \cdot v_1$. □

Now that we have finally defined the mappings $\pi$ and $\hat{\pi}$ between plays, we are ready to prove the correctness of our construction, that is Lemma 4.6.

$\boxed{\Rightarrow}$ Let $f_{\mathcal{M}}$ be a memoryless winning strategy of $\mathcal{G}_{\mathcal{M}}$. Observe that, starting from $v \in V_0'$, there is a unique partial $f_{\mathcal{M}}$-play $v \cdot \rho \cdot v'$ with $\rho \in V_?^* \cap V_0'^*$ and $v' \in V_{\text{sim}} \cup V_{\text{win}}$. We then let $\overline{f_{\mathcal{M}}}(v) = v'$.

We can hence define a strategy for Player 0 in $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$ as follows.

$$f_{\mathcal{P}}((s_{\text{in}}), 0, 1) = \hat{\pi}(f_{\mathcal{M}}(s_{\text{in}}', \varepsilon, \varepsilon, 0, 1))$$

Let $(c, p, r) \in V_0$,

$$f_{\mathcal{P}}(c, p, r) = \begin{cases} \hat{\pi}(\overline{f_{\mathcal{M}}}(\pi(c, p, r))) & \text{if } \overline{f_{\mathcal{M}}}(\pi(c, p, r)) \in V_{\text{sim}} \\ \text{any sucessor of } (c, p, r) & \text{otherwise.} \end{cases}$$

By Corollary 4.12, if $\overline{f_{\mathcal{M}}}(\pi(c, p, r)) \in V_{\text{sim}}$, it is well-defined, and $\hat{\pi}(\overline{f_{\mathcal{M}}}(\pi(c, p, r))$ is correctly defined.

We show that if $\rho$ is an $f_{\mathcal{P}}$-play of $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$, then $\pi(\rho)$ is an $f_{\mathcal{M}}$-play. In fact, by Lemma 4.9, $\pi(\rho)$ is a $\mathcal{G}_{\mathcal{M}}$-run. We then show by induction that for any finite prefix $\rho$ of an $f_{\mathcal{P}}$-play ending in $u \in V_0$, if $f_{\mathcal{M}}$ is winning, then $\pi(\rho \cdot f_{\mathcal{P}}(u))$ is an $f_{\mathcal{M}}$-play.

If $\rho$ consists in one node, it is obvious. Let now $\rho$ be an $f_{\mathcal{P}}$-play ending in $u \in V_0$, and assume that $\pi(\rho)$ is an $f_{\mathcal{M}}$-play. If $\pi(\rho) \in V'^* \cdot V_{\text{win}_1}^{\omega}$, then by definition of $\pi$, $u \notin \mathcal{F}$ and has no successor, hence $f_{\mathcal{P}}(u)$ is not defined, and $\rho$ is maximal. If $\pi(\rho) \in V'^* \cdot V_{\text{win}_0}^{\omega}$, then $\rho$ is winning, and $\pi(\rho \cdot f_{\mathcal{P}}(u)) = \pi(\rho)$ and it is then an $f_{\mathcal{M}}$-play by induction hypothesis. Otherwise, $\pi(\rho)$ ends in $\pi(u)$ and $u \notin \mathcal{F}$ and has at least one successor. Let $f_{\mathcal{P}}(u) = u'$. Then, $\pi(\rho \cdot u') = \pi(\rho) \cdot \text{next}_p^{p'}(\pi(u)) \cdot \pi(u') \cdot \Gamma$, where $\Gamma \in \{\varepsilon\} \cup V_{\text{win}}^{\omega} \cup V_1 V_{\text{win}}^{\omega}$ (by definition of $\pi$).

- If $\overline{f_{\mathcal{M}}}(\pi(u)) \in V_{\text{win}}$, then if $\overline{f_{\mathcal{M}}}(\pi(u)) \in V_{\text{win}1}$, it implies that $f_{\mathcal{M}}$ is not winning. Otherwise, $\overline{f_{\mathcal{M}}}(\pi(u)) \in V_{\text{win}0}$, which is a contradiction with $\pi(\rho)$ finite, our last assumption. Indeed, if $f_{\mathcal{M}}(\pi(u)) \neq \overline{f_{\mathcal{M}}}(\pi(u))$, then the predecessor $v$ of $\overline{f_{\mathcal{M}}}(\pi(u))$ in the corresponding $f_{\mathcal{M}}$-play is in $V_?$, and, by Lemma 4.11, $v$ is $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$-real. Since $\pi(u) \in V'_0$, so is $v$. The definition of $\mathcal{G}_{\mathcal{M}}$ implies that $\overline{f_{\mathcal{M}}}(\pi(u))$ cannot be in $V_{\text{win}0}$, a contradiction. Hence, $\overline{f_{\mathcal{M}}}(\pi(u)) = f_{\mathcal{M}}(\pi(u)) \in V_{\text{win}0}$, and $\pi(u) \in \mathfrak{F}$ (again, because $\pi(u)$ is $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$-real). As a consequence, $u \in \mathcal{F}$ and $\pi(\rho)$ is infinite.
- If $\overline{f_{\mathcal{M}}}(\pi(u)) \in V_{\text{sim}}$, it is necessarily $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$-real. Indeed, if a well-defined node of $V_{\text{sim}}$ is not $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$-real, its predecessor is in $V_{\text{sim}} \cap V'_1$. The predecessor of $\overline{f_{\mathcal{M}}}(\pi(u))$ is either $\pi(u)$ or a node of $V_?$. Since we have assumed that $u \in V_0$, then $\pi(u) \in V'_0$ by definition of $\pi$, a contradiction with $\overline{f_{\mathcal{M}}}(\pi(u)) \in V_{\text{sim}}$ not being $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$-real. Then, by definition of $f_{\mathcal{P}}$, $u' = \hat{\pi}(\overline{f_{\mathcal{M}}}(\pi(u)))$, and $\pi(u') = \pi(\hat{\pi}(\overline{f_{\mathcal{M}}}(\pi(u)))) = \overline{f_{\mathcal{M}}}(\pi(u))$, and we can write that $\pi(\rho \cdot u') = \pi(\rho) \cdot \text{next}^{p'}_p(\pi(u)) \cdot \overline{f_{\mathcal{M}}}(\pi(u)) \cdot \Gamma$, with $\pi(\rho) \cdot \text{next}^{p'}_p(\pi(u)) \cdot \overline{f_{\mathcal{M}}}(\pi(u))$ being an $f_{\mathcal{M}}$-play by definition of $\overline{f_{\mathcal{M}}}$.
  - If $u' \in \mathcal{F}$, $\Gamma \in V^{\omega}_{\text{win}0}$. Moreover, $\pi(u') \in \mathfrak{F}$, so its only successor in $\mathcal{G}_{\mathcal{M}}$ is in $V_{\text{win}0}$ and $\pi(\rho \cdot u')$ is an $f_{\mathcal{M}}$-play.
  - If $u' \notin \mathcal{F}$ and has no successor, then we can show that $\overline{f_{\mathcal{M}}}(\pi(u')) \in V_{\text{win}1}$. Indeed, assume that $\overline{f_{\mathcal{M}}}(\pi(u')) = v \in V_{\text{sim}}$. Then $v$ is $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$-real since $u'$ is $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$-real. Hence, let $\rho'$ such that $\pi(u') \cdot \rho' \cdot v$ is a partial $f_{\mathcal{M}}$-run. In that case, since $\pi(\rho) \cdot \text{next}^{p'}_p(\pi(u)) \cdot \pi(u') \cdot \rho' \cdot v$ is an $f_{\mathcal{M}}$-run, by Lemma 4.13, $\hat{\pi}(\pi(\rho) \cdot \text{next}^{p'}_p(\pi(u)) \cdot \pi(u') \cdot \rho' \cdot v) = \rho \cdot u' \cdot \hat{\pi}(v)$ is a play of $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$, a contradiction with the fact that $u'$ has no successor. So $\overline{f_{\mathcal{M}}}(\pi(u')) \in V_{\text{win}}$. Since $u' \notin \mathcal{F}$, $\pi(u') \notin \mathfrak{F}$, and $\overline{f_{\mathcal{M}}}(\pi(u')) \in V_{\text{win}1}$, which implies that $f_{\mathcal{M}}$ is not winning.
  - Otherwise, $\pi(\rho \cdot u') = \pi(\rho) \cdot \text{next}^{p'}_p(\pi(u)) \cdot \pi(u')$ which is an $f_{\mathcal{M}}$-play.

Now let $\rho$ be a maximal $f_{\mathcal{P}}$-play. Then $\pi(\rho)$ is an $f_{\mathcal{M}}$-play. Assume that $\rho$ is not winning. If it is finite, it ends in a node $u \notin \mathcal{F}$ without any successor, and $\pi(\rho) \in V'^* V^{\omega}_{\text{win}1}$, a contradiction with $f_{\mathcal{M}}$ being winning. If it is infinite, then it never visits $\mathcal{F}$, and by construction, $\pi(\rho)$ is an infinite play that never visits $V_{\text{win}0}$. Hence, $\pi(\rho)$ is not winning, which again contradicts the fact that $f_{\mathcal{M}}$ is winning. Then $\rho$ is winning, and $f_{\mathcal{P}}$ is a winning strategy for Player 0 in $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$.

$\boxed{\Leftarrow}$ Let $f_{\mathcal{P}}$ be a winning strategy of $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$, and let us build a strategy $f_{\mathcal{M}}$ for $\mathcal{G}_{\mathcal{M}}$. Let $\hat{\rho}$ be a play of $\mathcal{G}_{\mathcal{M}}$ ending in a node of Player 0, and $\hat{v}$ be the last $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$-*real* node in $V_{\text{sim}}$ of $\hat{\rho}$: $\hat{\rho} = \hat{\rho}_1 \cdot \hat{v} \cdot \hat{\rho}_2$.

By Lemma 4.13, $\hat{\pi}(\hat{\rho})$ is a play of $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$ ending in a node $v = \hat{\pi}(\hat{v})$. Let $v = (c, p, r)$. Observe that if $v \in V^B_1$, since the original play ends in $V_0$, it implies that there has been a change of player. Hence, the successor of $\hat{v}$ in that case is $\text{Ch}_0(v)$. Given such a partial play $\hat{\rho}$, we give the definition of *next*$(\hat{\rho})$, which describes how to extend $\hat{\rho}$ to the next $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$-*real* node in $V_{\text{sim}}$, or to $V_{\text{win}}$, by simulating the strategy $f_{\mathcal{P}}$ on $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$. Given a node $u \in V$, we write $\text{win}_i(u)$ the successor of $u$ in $V_{\text{win}i}$.

We define $next(\hat{\rho})$ as follows.

$$next(\hat{\rho}) = \begin{cases} \hat{\rho}_1 \cdot \hat{v} \cdot \mathrm{win}_0(\hat{v}) & \text{if } v \in \mathcal{F} \\ \hat{\rho}_1 \cdot \hat{v} \cdot \mathrm{Ch}_0(v) \cdot \mathrm{win}_1(\mathrm{Ch}_0(v)) & \text{if } v \in V_1^B \text{ without any successor in } \mathcal{G}^{\mathcal{P}}_{B\text{-rb}} \\ \hat{\rho}_1 \cdot \hat{v} \cdot \mathrm{Ch}_0(v) \cdot \mathrm{next}_p^{p'}(\mathrm{Ch}_0(v)).\mathrm{win}_0(v_?) & \text{if } v \in V_1^B \text{ and there exists } v' \in V^B, (v, v') \in E^B_{p'}, \text{ with } v_? \in V_? \text{ the node described in Lemma 4.8.} \\ \hat{\rho}_1 \cdot \hat{v} \cdot \mathrm{win}_1(\hat{v}) & \text{if } v \in V_0^B \text{ and has no successor in } \mathcal{G}^{\mathcal{P}}_{B\text{-rb}} \\ \pi(\hat{\pi}(\hat{\rho}) \cdot f_{\mathcal{P}}(\hat{\pi}(\hat{\rho})) & \text{otherwise.} \end{cases}$$

Observe that $next(\hat{\rho})$ is a partial play of $\mathcal{G}_{\mathcal{M}}$, in which all the nodes visited after $\hat{v}$ are in $V_0$.

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

$$f_{\mathcal{M}}(\hat{\rho} \cdot \hat{v}) = \begin{cases} \hat{v} & \text{if } \hat{v} \in V_{\mathrm{win}} \\ \hat{v'} & \text{if } \hat{\rho} \cdot \hat{v} \cdot \hat{v'} \text{ is a prefix of } next(\hat{\rho} \cdot \hat{v}) \\ w & \text{with } w \text{ any successor of } \hat{v} \text{ in } \mathcal{G}_{\mathcal{M}} \text{ otherwise.} \end{cases}$$

We show by induction on the length of $\hat{\rho}$ that if it is a (partial) $f_{\mathcal{M}}$-play, then $\hat{\pi}(\hat{\rho})$ is a (partial) $f_{\mathcal{P}}$-play. When $\hat{\rho}$ is restricted to the initial node, it is trivial. Let $\hat{\rho}$ be an $f_{\mathcal{M}}$-play and suppose that $\hat{\pi}(\hat{\rho})$ is an $f_{\mathcal{P}}$-play. Consider now $\hat{\rho} \cdot v$ an $f_{\mathcal{M}}$-play. If $\hat{\pi}(\hat{\rho} \cdot v) = \hat{\pi}(\hat{\rho})$, then obviously $\hat{\pi}(\hat{\rho} \cdot v)$ is an $f_{\mathcal{P}}$-play. If $\hat{\pi}(\hat{\rho} \cdot v) = \hat{\pi}(\hat{\rho}) \cdot \hat{\pi}(v)$ and $\hat{\pi}(\hat{\rho})$ ends in $V_1^B$, obvisouly $\hat{\pi}(\hat{\rho} \cdot v)$ is an $f_{\mathcal{P}}$-play. Assume then that $\hat{\pi}(\hat{\rho} \cdot v) = \hat{\pi}(\hat{\rho}) \cdot \hat{\pi}(v)$, i.e. $v$ is a $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$-real node in $V_{\mathrm{sim}}$, and $\hat{\pi}(\hat{\rho})$ ends in $V_0^B$. Let $v_1$ be the last $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$-real node of $\hat{\rho}$ in $V_{\mathrm{sim}}$: $\hat{\rho} = \hat{\rho}_1 \cdot v_1 \cdot \hat{\rho}_2$. Then observe that $\hat{\pi}(\hat{\rho}) = \hat{\pi}(\hat{\rho}_1) \cdot \hat{\pi}(v_1)$. Then, since $\hat{\pi}(v_1) \in V_0^B$, $v_1 \in V_0$, and by construction of the game $\mathcal{G}_{\mathcal{M}}$, $\hat{\rho}_2 \in V_0^*$. Then, $\hat{\rho}$ ends in $V_0$, and $v = f_{\mathcal{M}}(\hat{\rho})$. We need to show that $\hat{\pi}(v) = f_{\mathcal{P}}(\hat{\pi}(\hat{\rho}))$.

By definition, $next(\hat{\rho}) = next(\hat{\rho}_1 \cdot v_1)$. We show that any prefix $\hat{\rho}_1 \cdot v_1 \cdot \hat{\rho}'_2$ of $\hat{\rho}$ is a prefix of $next(\hat{\rho}_1 \cdot v_1)$. By Corollary 4.14, if $next(\hat{\rho}_1 \cdot v_1) = \pi(\hat{\pi}(\hat{\rho}_1 \cdot v_1) \cdot f_{\mathcal{P}}(\hat{\pi}(\hat{\rho}_1 \cdot v_1)))$ then $next(\hat{\rho}_1 \cdot v_1) = \hat{\rho}_1 \cdot v_1 \cdot \mathrm{next}_p^{p'}(v_1) \cdot \pi(f_{\mathcal{P}}(\hat{\pi}(\hat{\rho}_1 \cdot v_1)))$, with $f_{\mathcal{P}}(\hat{\pi}(\hat{\rho}_1 \cdot v_1)) = (c', p', r')$. So if $\hat{\rho}'_2 = \varepsilon$, $\hat{\rho}_1 \cdot v_1 \cdot \hat{\rho}'_2$ is obviously a prefix of $next(\hat{\rho}_1 \cdot v_1)$.

Let now $\hat{\rho}_1 \cdot v_1 \cdot \hat{\rho}'_2$ be a prefix of $\hat{\rho}$, and assume that it is also a prefix of $next(\hat{\rho}_1 \cdot v_1)$. Let $\hat{\rho}_1 \cdot v_1 \cdot \hat{\rho}'_2 \cdot v_2$ be a prefix of $\hat{\rho}$. Since $\hat{\rho}_2 \in V_0^*$, $v_2 = f_{\mathcal{M}}(\hat{\rho}_1 \cdot v_1 \cdot \hat{\rho}'_2)$. We know that no node of $\hat{\rho}_2$ is in $V_{\mathrm{win}}$, otherwise, $\hat{\rho} \cdot v$ with $v \in V_{\mathrm{sim}}$ would not be a play of $\mathcal{G}_{\mathcal{M}}$. So, by induction hypothesis and by definition of $f_{\mathcal{M}}$, $\hat{\rho}_1 \cdot v_1 \cdot \hat{\rho}'_2 \cdot v_2$ is a prefix of $next(\hat{\rho}_1 \cdot v_1)$.

Then $v = f_{\mathcal{M}}(\hat{\rho}_1 \cdot v_1 \cdot \hat{\rho}_2)$ is such that $\hat{\rho}_1 \cdot v_1 \cdot \hat{\rho}_2 \cdot v$ is a prefix of $next(\hat{\rho})$. Since $v_1 \in V_0$ and $v \in V_{\mathrm{sim}}$, the only possibility for $next(\hat{\rho})$ is that $next(\hat{\rho}) = \pi(\hat{\pi}(\hat{\rho}) \cdot f_{\mathcal{P}}(\hat{\pi}(\hat{\rho}))) = \hat{\rho}_1 \cdot v_1 \cdot \mathrm{next}_p^{p'}(v_1) \cdot \pi(f_{\mathcal{P}}(\hat{\pi}(\hat{\rho}_1 \cdot v_1)))$. Then, $v = \pi(f_{\mathcal{P}}(\hat{\pi}(\hat{\rho}_1 \cdot v_1)))$, and hence $\hat{\pi}(v) = \hat{\pi}(\pi(f_{\mathcal{P}}(\hat{\pi}(\hat{\rho}_1 \cdot v_1))) = f_{\mathcal{P}}(\hat{\pi}(\hat{\rho}_1 \cdot v_1)) = f_{\mathcal{P}}(\hat{\pi}(\hat{\rho}))$.

Hence if $\hat{\rho}$ is a finite (partial) $f_{\mathcal{M}}$-play, $\hat{\pi}(\hat{\rho})$ is a finite (partial) $f_{\mathcal{P}}$-play. Moreover, if $\hat{\rho}$ is an infinite $f_{\mathcal{M}}$-play, then $\hat{\pi}(\hat{\rho})$ is a $f_{\mathcal{P}}$-play (otherwise, we could find a finite prefix $\hat{\rho}'$ of $\hat{\rho}$ such that $\hat{\pi}(\hat{\rho}')$ is not a $f_{\mathcal{P}}$-play).

Suppose there is an $f_{\mathcal{M}}$-play $\hat{\rho}$ that is maximal (i.e infinite) and not winning. That means that either it ends in $V_{\text{win}1}$, which is a sink state, or $\hat{\rho}$ visits $V_{\text{sim}}$ infinitely often (as it is not possible for either player to stay in $V_?$ indefinitely). In both cases, $\hat{\rho}$ does not visit any node in $\mathfrak{F}$, because the only successor of such nodes are in $V_{\text{win}0}$, contradicting the fact that $\hat{\rho}$ is not winning. If $\hat{\rho}$ visits infinitely often nodes in $V_{\text{sim}}$, then $\hat{\pi}(\hat{\rho})$ is also infinite, and does not visit any node in $\mathcal{F}$. Since $\hat{\pi}(\hat{\rho})$ is an $f_{\mathcal{P}}$-play, it contradicts the assumption that $f_{\mathcal{P}}$ is winning.

Since $V_{\text{win}1}$ is only accessible from nodes in $V_0$, if $\hat{\rho}$ ends in $V_{\text{win}1}$, there is some prefix $\hat{\rho}'$ of $\hat{\rho}$ ending in $V_0$ and $f_{\mathcal{M}}(\hat{\rho}') \in V_{\text{win}1}$, meaning that $next(\hat{\rho}')$ ends in $V_{\text{win}1}$. By definition of *next*, it means that $\hat{\pi}(\hat{\rho}')$ ends in a node without any successor in $\mathcal{G}^{\mathcal{P}}_{B\text{-rb}}$. Hence, $\hat{\pi}(\hat{\rho}')$ is a maximal $f_{\mathcal{P}}$-play, that is not winning, which again is a contradiction. Hence, $f_{\mathcal{M}}$ is a winning strategy. This concludes the proof of Lemma 4.6 and this section.

### 4.3 Lower bound for round-bounded control

**Theorem 4.15** DFS- CONTROL$_{rb}$ *is inherently non-elementary.*

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 [37]. Note that the lower bound already holds for DFS.

Let Var be a countably infinite set of variables and $\Sigma$ a finite alphabet. Formulas $\varphi$ 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 $x, y \in \text{Var}$ 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 $\nu : \text{Var} \to \{0, \dots, n-1\}$. 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 [38] for details. A formula $\varphi$ without free variables is *satisfiable* if there is a word $w$ such that $w, \emptyset \models \varphi$. We suppose that $\varphi$ is given in prenex normal form and that all the quantified variables are pairwise distinct.

We build a DFS-based round-bounded game that is winning for Player 0 if and only if $\varphi$ 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 $\varphi$, the following rounds are devoted to the valuation of the variables appearing in $\varphi$, $\nu(x) = i$ being represented by memorizing the variable $x$ in the local state of the $i^{\text{th}}$ 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 $\varphi_1$ or $\varphi_2$; 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. Here is the formalization and proof of this idea.

Let $\varphi$ be a formula, $\text{Cl}(\varphi)$ the set of subformulas (non-strict) of $\varphi$, and $\text{Var}_\varphi \subset \text{Var}$ the set of variables appearing in $\varphi$. We define $B = |\text{Var}_\varphi| + 2$ and $\mathcal{P} = (S_0 \uplus S_1, L, s_{\text{in}}, \ell_{\text{in}}, \Delta, F_{\text{glob}}, F_{\text{loc}})$ 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 $s_{\mathsf{in}}$, and $F_{\mathsf{glob}} = \{\text{Win}\}$.
- $L = \{\ell_{\mathsf{in}}, \text{first}\} \cup \left(\Sigma \times 2^{\text{Var}_\varphi}\right)$, with initial state $\ell_{\mathsf{in}}$ and $F_{\mathsf{loc}} = L$.
- $\Delta$ contains the following transitions:

1. $(s_{\mathsf{in}}, \ell_{\mathsf{in}}) \to (\text{Guess}, \text{first})$
2. $(\text{Guess}, \ell_{\mathsf{in}}) \to (\text{Guess}, (a, \emptyset))$, for $a \in \Sigma$
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. $(x < y, (a, S)) \to (\text{Win-if-}y, (a, \emptyset))$ if $x \in S$
9. $(\neg(x < y), (a, S)) \to (\text{Win-if-}x, (a, S))$ if $y \in S$
10. $(\text{Win-if-}x, (a, S)) \to (\text{Win}, (a, S))$ if $x \in S$

The first process to be activated (which will stay in the local state first all along) constrains the evolution of the play: in a first part, when the global state is in Guess, the different processes will guess the different letters of the word, then the first process will force the system to start the verification of the formula on the word (with transition 3.). Then the different transitions of type 5. will build the valuation of the formula by associating each variable to a position in the word. Transition 4. ensures that the first process plays after each transition 5., ensuring a change of round each time.

**Lemma 4.16** *There is a winning strategy for Player 0 in $\mathcal{P}$ if and only if $\varphi$ is satisfiable.*

***Proof*** Given a configuration $c = (s, \text{first}, (a_1, S_1), \ldots, (a_n, S_n))$ 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 $i \neq j$: indeed we have assumed that the quantified variables were pairwise distinct, and a variable can be added to a local state $S_i$ only upon taking a transition of type 5., which modify the global state to the immediate subformula. Observe also that along a run the local state of a process is only slightly modified: once the letter is chosen, it never changes, and the set S can only grow along the run, until the last round where it can be set to $\emptyset$ again. Hence, if $c$ and $c'$ are two configurations occurring along the same play $\rho$, if $\nu(c)(x)$ and $\nu(c')(x)$ are both defined, then $\nu(c)(x) = \nu(c')(x)$. Conversely, given a state $s \in S$, a word $w = a_1 \ldots a_n$ and a valuation $\nu$, the associated configuration is $c(s, w, \nu) = (s, \text{first}, (a, S_1), \ldots, (a, S_n))$ where $S_i = \{x \mid \nu(x) = i\}$.

$\boxed{\Rightarrow}$ Let $f$ be a winning strategy for Player 0. We fix $w_f = a_1 \ldots a_n$. We show by recursion on the subformula $\psi$, that for all nodes $v_\psi = (c_\psi, 1, r)$ with $c_\psi = (\psi, \text{first}, (a_1, S_1), \ldots, (a_n, S_n))$ visited during an $f$-play, we have $w_f, \nu(c_\psi) \models \psi$.

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

- If $\psi = a(x)$, then if $v_\psi$ occurs in an $f$-play, since $f$ is winning, the transition 7. can be taken, and there is a process with local state $(a_i, S_i)$ with $a_i = a$ and $x \in S_i$. In other words $\nu(c_\psi)(x) = i$, so we have $w_f, \nu(c_\psi) \models \psi$.
- If $\psi = x < y$, any $f$-play will continue with transitions 8. then 10., leading to $v_1 = ((\text{Win-if-}y, (a_1, S_1), \ldots, (a_i, \emptyset), \ldots, (a_n, S_n)), i + 1, B)$ and $v_2 =$

$((\text{Win}, (a_1, S_1), \ldots, (a_i, \emptyset), \ldots, (a_n, S_n))), j+1, B)$. 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, since with transition 9., $S_j$ is not emptied.
- If $\psi = \psi_1 \vee \psi_2$, then $v_\psi \in S_0$, and let $\rho$ be an $f$-play starting with $\rho_1 v_\psi$. Let $v_{\psi_i}$ be the successor of $v_\psi$ in $\rho$ (by transition 6.). Since $\psi_i$ is a subformula of $\psi$, by induction hypothesis, $w_f, \nu(c_{\psi_i}) \models \psi_i$. Moreover, $\nu(c_{\psi_i}) = \nu(c_\psi)$ (as no local state is changed during the transition), then $w_f, \nu(c_\psi) \models \psi_i$, which in turn means that $w_f, \nu(c_\psi) \models \psi$.
- If $\psi = \psi_1 \wedge \psi_2$, then $v_\psi \in S_1$. Let $\rho \cdot v_\psi$ be a prefix of an $f$-play. Then both $\rho \cdot v_\psi \cdot v_{\psi_1}$ and $\rho \cdot v_\psi \cdot v_{\psi_2}$ are prefixes of an $f$-play. By induction hypothesis, we obtain that $w_f, \nu(c_{\psi_i}) \models \psi_i$ for $i \in \{1, 2\}$. Moreover, $\nu(\psi) = \nu(\psi_i)$ for $i \in \{1, 2\}$, then $w_f, \nu(c_\psi) \models \psi$.
- If $\psi = \exists x.\psi'$, then $v_\psi \in S_0$. Let $\rho$ be an $f$-play starting with $\rho_1 \cdot v_\psi$, and let $v_1 = ((\boxed{\psi'}, \text{first}, (a_1, S_1), \ldots, (a_i, S_i \cup \{x\}), \ldots, (a_n, S_n)), i+1, r)$ with $1 \leq i$ the successor node of $v_\psi$ in $\rho$, and $v_{\psi'} = ((\psi', \ldots), 1, r+1)$ the successor of $v_1$ (transitions 5. then 4.). By induction hypothesis, $w_f, \nu(c_{\psi'}) \models \psi'$. Moreover, $\nu(c_{\psi'}) = \nu(c_\psi) \uplus \{x \to i\}$ by construction. Thus $w_f, \nu(c_\psi) \models \psi$.
- If $\psi = \forall x.\psi'$, then $v_\psi \in S_1$. Let $\rho \cdot v_\psi$ be the prefix of an $f$-play. For all $1 \leq i \leq n$, we let $v_1^i = ((\boxed{\psi'}, \text{first}, (a_1, S_1), \ldots, (a_i, S_i \cup \{x\}), \ldots, (a_n, S_n)), i+1, r)$, $c^i_{\psi'} = (\psi', \text{first}, (a_1, S_1), \ldots, (a_i, S_i \cup \{x\}), \ldots, (a_n, S_n))$, $v^i_{\psi'} = (c^i_{\psi'}, 1, r+1)$. Then for all $1 \leq i \leq n$, $\rho \cdot v_\psi \cdot v_1^i \cdot v^i_{\psi'}$ is the prefix of an $f$-play. By induction hypothesis, $w_f, \nu(c^i_{\psi'}) \models \psi'$. Moreover, $\nu(c^i_{\psi'}) = \nu(c_\psi) \uplus \{x \to i\}$. 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 $\varphi$.

$\boxed{\Leftarrow}$ Now suppose that $a_1 \ldots a_n, \emptyset \models \varphi$. We build a memoryless strategy $f_\varphi$ as follows. Let $\rho$ be a run ending in $v$.

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

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

By definition of $f_\varphi$ 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 $k \in \{n+1, n+3, \ldots, n+2 \cdot (|\text{Var}_\varphi| - 1)\}$. 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 $f_\varphi$ 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 facts that $v_{n+1} = (c(\varphi, w, \emptyset), 1, 2)$ and that $w, \emptyset \models \varphi$, we deduce that $v_{n+2\cdot|\mathrm{Var}_\varphi|} = (c(\psi', w, \nu), 1, B)$ where $\psi'$ is quantifier-free and $w, \nu \models \psi'$.

For $k \geq n + 2 \cdot |\mathrm{Var}_\varphi|$, we have again 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 $f_\varphi$, $v_{k+1} = (c(\psi_i, w, \nu), 1, B)$ 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 $m' \geq n + 2 \cdot |\mathrm{Var}_\varphi|$ 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 $\varphi$ but also on both strategies $f_\varphi$ and $f'$. There are 3 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 $v_{m'+1} = (c(\mathrm{Win}, w, \nu), \nu(x) + 1, B)$.
2. $t = x < y$: we know that $\nu(x) < \nu(y)$ so $v_{m'+1} = (c(\text{Win-if-}y, w, \nu'), \nu(x), B)$ where $\nu' = \nu\backslash\{\{x' \to \nu(x)\} \mid x' \in \mathrm{Var}_\varphi\}$. Since $\nu(x) \neq \nu(y)$, $\nu'(y) = \nu(y) > \nu(x)$. So $v_{m'+2} = (c(\mathrm{Win}, w, \nu'), \nu(y), B)$.
3. $t = \neg(x < y)$: 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 $v_{m'+2} = (c(\mathrm{Win}, w, \nu), \nu(x), B)$.

Every case ends in an accepting node, therefore $\rho$ is winning. □

## 5 Undecidability of context-bounded games

In this section, we show that relaxing the notion of rounds quickly leads to undecidability. It should be noted that our undecidability proof also applies to the notion of context bounds introduced in [6].

### 5.1 Context-bounded runs

We now define *context-bounded runs*. A context is less restrictive than a round. As with round-bounded runs, each process gets to perform as many transitions as it wants but cannot act afterwards within the same context. However, within a context there is no fixed order on processes, as opposed to rounds where there is one fixed order which is the same for all rounds.

Accordingly, given $B \geq 1$ and a DPS $\mathcal{P} = (S, L, \Gamma, s_{\mathsf{in}}, \ell_{\mathsf{in}}, \Delta, F_{\mathsf{glob}}, F_{\mathsf{loc}})$, we define the *context-bounded semantics* of $\mathcal{P}$ as $[\![\mathcal{P}]\!]_{B\text{-cb}} = (V^B, E^B, v_{\mathsf{in}}^B)$ where

- Nodes are of the form $v = (c, P, p, r)$ where $c \in C_\mathcal{P}$ is a configuration, say, of size $k$, $P \subseteq \{1, \ldots, k\}$ is The set of processes that made a transition in the current context,

$p \in P \cup \{0\}$ is the last process that made a transition (or 0 if $P = \emptyset$), and $r \in \{1, \ldots, B\}$ is the number of the current context,

- The initial node is $v_{\mathsf{in}}^B = ((s_{\mathsf{in}}), \emptyset, 0, 1)$, and
- There is an edge between $(c, P, p, r)$ and $(c', P', p', r')$ if, in $[\![\mathcal{P}]\!] = (V, E, v_{\mathsf{in}})$, there is an edge $(c, c')$ in $E_{p'}$ and either
  - $p' \notin P \backslash \{p\}$, $P' = P \cup \{p'\}$, and $r' = r$, or
  - $p' \in P \backslash \{p\}$, $P' = \{p'\}$, $r < B$, and $r' = r + 1$.

The bounded semantics of a DFS is defined accordingly. We extend the definition of the acceptance condition $\mathcal{F}_{\mathcal{P}}$ of the DPS to the nodes of its context-bounded semantics, and we define (accepting) *B-context-bounded runs* as expected.

***Relation to round-bounded runs*** Note that if a run is $B$-round bounded for some $B$, then it is trivially $B$-context bounded too. Conversely for any $n \in \mathbb{N}$, there are 2-context bounded runs that are not $n$-round bounded: for instance, a run where processes 1 to $n + 1$ do one transition each in the first half, followed in the second half by one transition from processes $n$ down to 1. Such a run is 2-context bounded (one for each half), but it needs at least $n + 1$ rounds to be done (one for the first half, then $n$ rounds for transition from $n$ to 1).

***Context-bounded control*** Given a bound $B \geq 1$, analogously to the round-bounded case, we define *B-context-bounded parameterized pushdown game* induced by $\mathcal{P}$ as the game $\mathcal{G}_{B\text{-cb}}^{\mathcal{P}}$ given by $[\![\mathcal{P}]\!]_{B\text{-cb}} = (V^B, E^B, v_{\mathsf{in}}^B)$ with nodes $v = (c, p, r) \in V^B$ such that $c = (s, (\ell_1, \gamma_1), \ldots, (\ell_k, \gamma_k))$ belonging to Player $j$ if $s \in S_j$.

DPS- CONTROL

**I**: DPS $\mathcal{P} = (S_0 \uplus S_1, L, \Gamma, s_{\mathsf{in}}, \ell_{\mathsf{in}}, \Delta, F_{\mathsf{glob}}, F_{\mathsf{loc}})$; $B \geq 1$
**Q**: Does Player 0 have an $\mathcal{F}_{\mathcal{P}}$-winning strategy in $\mathcal{G}_{B\text{-cb}}^{\mathcal{P}}$ ?

## 5.2 Undecidabilty for context-bounded runs

We show that even for DFS, and even with a fixed bound, the control problem is undecidable. This shows that relaxing the round-bounded constraint even a little easily leads to undecidability.

**Theorem 5.1** DFS- CONTROL$_{cb}$ *is undecidable, even if we fix* $B = 2$.

The rest of subsection Sect. 5.2 is devoted to the proof of Theorem 5.1.

We provide a reduction from the halting problem for 2-counter machines, whose definition is recalled in the following.

A *two-counter machine* (2CM) with counters $\mathsf{c}_1$ and $\mathsf{c}_2$ is given by a tuple $M = (Q, T, q_0, q_h)$, where $Q$ is the finite set of states and $T \subseteq Q \times \mathsf{Op} \times Q$ is the transition relation where the set of operations is defined as $\mathsf{Op} = \{\mathsf{c}_i{+}{+},\ \mathsf{c}_i{-}{-},\ \mathsf{c}_i{=}{=}0 \mid i \in \{1, 2\}\}$. As expected, $\mathsf{c}_i{+}{+}$ increments counter $\mathsf{c}_i$, while $\mathsf{c}_i{-}{-}$ decrements it, and $\mathsf{c}_i{=}{=}0$ checks whether its value is 0. Moreover, there are a distinguished initial state $q_0 \in Q$ and a halting state $q_h \in Q$.

The behavior of $M$ is described in terms of a global transition relation over configurations $\gamma = (q, \nu_1, \nu_2) \in Q \times \mathbb{N} \times \mathbb{N}$ where $q$ is the current state and $\nu_1, \nu_2$ are the current

counter values. Every transition $t \in T$ defines a binary relation $\vdash_t$ on configurations letting $(q, \nu_1, \nu_2) \vdash_t (q', \nu_1', \nu_2')$ if there is $i \in \{1, 2\}$ such that $\nu'_{3-i} = \nu_{3-i}$ and one of the following conditions hold:

- $t = (q, \mathsf{c}_i{+}{+}, q')$ and $\nu_i' = \nu_i + 1$,
- $t = (q, \mathsf{c}_i{-}{-}, q')$ and $\nu_i' = \nu_i - 1$, or
- $t = (q, \mathsf{c}_i{=}{=}0, q')$ and $\nu_i = \nu_i' = 0$.

An ($M$-)run is a sequence of the form $\gamma_0 \vdash_{t_1} \gamma_1 \vdash_{t_2} \ldots \vdash_{t_n} \gamma_n$ where $\gamma_0 = (q_0, 0, 0)$. It is successful if $\gamma_n \in F = \{q_h\} \times \mathbb{N} \times \mathbb{N}$. Now, the 2CM halting problem is to decide whether there is a successful run. It is well known that this problem is undecidable [33].

Let $M = (Q, T, q_0, q_h)$ be a 2CM. We define $\mathcal{P}$ and a game $\mathcal{G}^{\mathcal{P}}_{B\text{-cb}}$ with $B = 2$, with the following intuition. In the first context, Player 0 will simulate a run of the 2CM. The global state of the game will be the state of the 2CM. To encode the values of the counters, there are two local states $\ell_i$ and $\bar{\ell}_i$ for $i \in \{1, 2\}$, and the value of counter $i$ will be encoded as the number of processes with local state $\ell_i$ minus the number of processes with local state $\bar{\ell}_i$. To simulate a transition $(q, \mathsf{c}_i{+}{+}, q')$, Player 0 will change the global state from $q$ to $q'$ and create a new process with local state $\ell_i$. Similarly, for a transition $(q, \mathsf{c}_i{-}{-}, q')$, Player 0 will change the global state from $q$ to $q'$ and create a new process with local state $\bar{\ell}_i$. Finally, a transition $(q, \mathsf{c}_i{=}{=}0, q')$ is simulated by changing the global state from $q$ to $q'$ and creating a new process with a dummy local state $\ell_\perp$ that is not counted in the encoding of the values of $\mathsf{c}_1$ and $\mathsf{c}_2$. All local states are accepting and only $q_h$ is an accepting global state, so Player 0 wins if she can simulate a run of the 2CM leading to $q_h$.

However, we must ensure that Player 0 does not cheat during the simulation, that is that Player 0 does not decrement counter $i$ if its value is 0 or takes a zero-test transition when its value is not 0. To that end, whenever Player 0 simulates a decrement or a zero-test transition, we leave the possibility for Player 1 to claim that the transition was incorrectly taken. When that happens, the simulation is stopped and a verification is started. This verification phase uses another context, and the game always ends after this phase (thus 2 contexts are enough for the game). If the transition was a decrement of counter $i$, then Player 1 and Player 0 will alternately make a transition with a process in state $\bar{\ell}_i$ and $\ell_i$ respectively, with Player 1 aiming to prove that there are now more $\bar{\ell}_i$ than $\ell_i$ (i.e. that $\mathsf{c}_i$ became negative as a result of the transition) and Player 0 aiming to disprove that. Eventually, the player who cannot make a transition anymore loses the game. Similarly, if the transition was a zero-test, Player 1 will try to prove that there is an unequal number of $\ell_i$ and $\bar{\ell}_i$, and Player 0 will try to disprove it. Therefore, Player 0's only way to win the game is to correctly simulate an accepting run of the 2CM.

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

- $S = \quad Q \cup \{win\}$
  $\cup \{?\mathrm{dec}^i_{(q,q')}, ?\mathrm{zero}^i_{(q,q')} \mid i \in \{1, 2\}, q, q' \in Q\}$
  $\cup \{\mathrm{vdec}^i_j \mid i \in \{1, 2\}, j \in \{1, 2\}\}$
  $\cup \{\mathrm{vzero}^i_j \mid i \in \{1, 2\}, j \in \{1, 2, 3\}\}$,
- $L = \{\ell_{\mathsf{in}}, \ell_1, \bar{\ell}_1, \ell_2, \bar{\ell}_2, \ell_\perp\}$,
- $s_{\mathsf{in}} = q_0$, $F_{\mathsf{glob}} = \{q_h, win\}$, $F_{\mathsf{loc}} = L$,
- $\Delta$ defined as in Table 2.

States are partitioned into

$$S_1 = \{?\mathrm{dec}^i_{(q,q')}, ?\mathrm{zero}^i_{(q,q')} \mid i \in \{1, 2\}, q, q' \in Q\} \cup \{\mathrm{vdec}^i_1, \mathrm{vzero}^i_1 \mid i \in \{1, 2\}\}$$

**Table 2** Transitions of $\mathcal{G}^{\mathcal{P}}_{B\text{-cb}}$, illustrated in Fig. 6

| # | Transition in $\Delta$ | Condition |
|---|---|---|
| 1 | $(q, \ell_{\mathsf{in}}) \to (q', \ell_i)$ | $(q, \mathsf{c}_i{+}{+}, q') \in T$ |
| 2 | $(q, \ell_{\mathsf{in}}) \to (?\mathrm{dec}^i_{(q,q')}, \bar{\ell}_i)$ | |
| 3 | $(?\mathrm{dec}^i_{(q,q')}, \ell_{\mathsf{in}}) \to (q', \ell_\perp)$ | $(q, \mathsf{c}_i{-}{-}, q') \in T$ |
| 4 | $(?\mathrm{dec}^i_{(q,q')}, \ell_{\mathsf{in}}) \to (\mathrm{vdec}^i_1, \ell_\perp)$ | |
| 5 | $(q, \ell_{\mathsf{in}}) \to (?\mathrm{zero}^i_{(q,q')}, \ell_\perp)$ | |
| 6 | $(?\mathrm{zero}^i_{(q,q')}, \ell_{\mathsf{in}}) \to (q', \ell_\perp)$ | $(q, \mathsf{c}_i{=}{=}0, q') \in T$ |
| 7 | $(?\mathrm{zero}^i_{(q,q')}, \ell_{\mathsf{in}}) \to (\mathrm{vzero}^i_1, \ell_\perp)$ | |
| 8 | $(\mathrm{vdec}^i_1, \bar{\ell}_i) \to (\mathrm{vdec}^i_2, \ell_\perp)$ | |
| 9 | $(\mathrm{vdec}^i_2, \ell_i) \to (\mathrm{vdec}^i_1, \ell_\perp)$ | $i \in \{1, 2\}$ |
| 10 | $(\mathrm{vdec}^i_1, \ell_{\mathsf{in}}) \to (\mathit{win}, \ell_\perp)$ | |
| 11 | $(\mathrm{vzero}^i_1, \ell_i) \to (\mathrm{vzero}^i_2, \ell_\perp)$ | |
| 12 | $(\mathrm{vzero}^i_2, \bar{\ell}_i) \to (\mathrm{vzero}^i_1, \ell_\perp)$ | |
| 13 | $(\mathrm{vzero}^i_1, \bar{\ell}_i) \to (\mathrm{vzero}^i_3, \ell_\perp)$ | $i \in \{1, 2\}$ |
| 14 | $(\mathrm{vzero}^i_3, \ell_i) \to (\mathrm{vzero}^i_1, \ell_\perp)$ | |
| 15 | $(\mathrm{vzero}^i_1, \ell_{\mathsf{in}}) \to (\mathit{win}, \ell_\perp)$ | |

and $S_0 = S \setminus S_1$, and we take $B = 2$. This ends the definition of $\mathcal{G}^{\mathcal{P}}_{B\text{-cb}}$. Refer to Fig. 6 for an illustration.

**Lemma 5.2** *There is an accepting run in $M$ iff Player 0 has a winning strategy in $\mathcal{G}^{\mathcal{P}}_{B\text{-cb}}$.*

***Proof*** To avoid possible confusions, a configuration of the 2CM $M$ may be referred as an $M$-configuration and will always be noted $\gamma$, whereas a configuration of the game $\mathcal{G}^{\mathcal{P}}_{B\text{-cb}}$ will be referred to as $c$. Moreover, runs of $M$ are denoted by $\rho$, and plays of $\mathcal{G}^{\mathcal{P}}_{B\text{-cb}}$ are denote by $\pi$.

For any $\mathcal{G}^{\mathcal{P}}_{B\text{-cb}}$-configuration $c = (s, \ell^1, \ldots, \ell^p)$ and $i \in \{1, 2\}$, let $st(c) = s$ and $n_i(c) = |\{1 \le j \le p \mid \ell^j = \ell_i\}| - |\{1 \le j \le p \mid \ell^j = \bar{\ell}_i\}|$. Let also $\min_i(c) = \min\{j \mid \ell^j = \ell_i\}$ if it exists.

One can build from any $M$-run $\rho$ a corresponding $\mathcal{G}^{\mathcal{P}}_{B\text{-cb}}$-play $\pi(\rho)$ inductively in the following way:

- $\pi(\gamma_0) = ((q_0), \emptyset, 0, 1)$,
- if $\pi(\rho)$ is defined and ends in $(c, [p], p, 1)$ with $c = (q, \ell^1, \ldots, \ell^p)$, then $\pi(\rho \vdash_t \gamma) =$

$$\begin{cases} \Big(\pi(\rho) \cdot ((q', \ell^1, \ldots, \ell^p, \ell_i), [p+1], p+1, 1)\Big) & \text{if } t = (q, \mathsf{c}_i{+}{+}, q') \\ \Big(\pi(\rho) \cdot ((?\mathrm{dec}^i_{(q,q')}, \ell^1, \ldots, \ell^p, \bar{\ell}_i), [p+1], p+1, 1) \\ \quad \cdot ((q', \ell^1, \ldots, \ell^p, \bar{\ell}_i, \ell_\perp), [p+2], p+2, 1)\Big) & \text{if } t = (q, \mathsf{c}_i{-}{-}, q') \\ \Big(\pi(\rho) \cdot ((?\mathrm{zero}^i_{(q,q')}, \ell^1, \ldots, \ell^p, \ell_\perp), [p+1], p+1, 1) \\ \quad \cdot ((q', \ell^1, \ldots, \ell^p, \ell_\perp, \ell_\perp), [p+2], p+2, 1)\Big) & \text{if } t = (q, \mathsf{c}_i{=}{=}0, q'). \end{cases}$$

This construction is such that for any $\rho$ ending in $\gamma = (q, \nu_1, \nu_2)$, we have that $\pi(\rho)$ ends in $(c, P, p, 1)$ with $st(c) = q$ such that $n_1(c) = \nu_1$ and $n_2(c) = \nu_2$. Remark also that $\pi(\rho)$ is winning for Player 0 iff $q_h$ is visited in $\rho$.

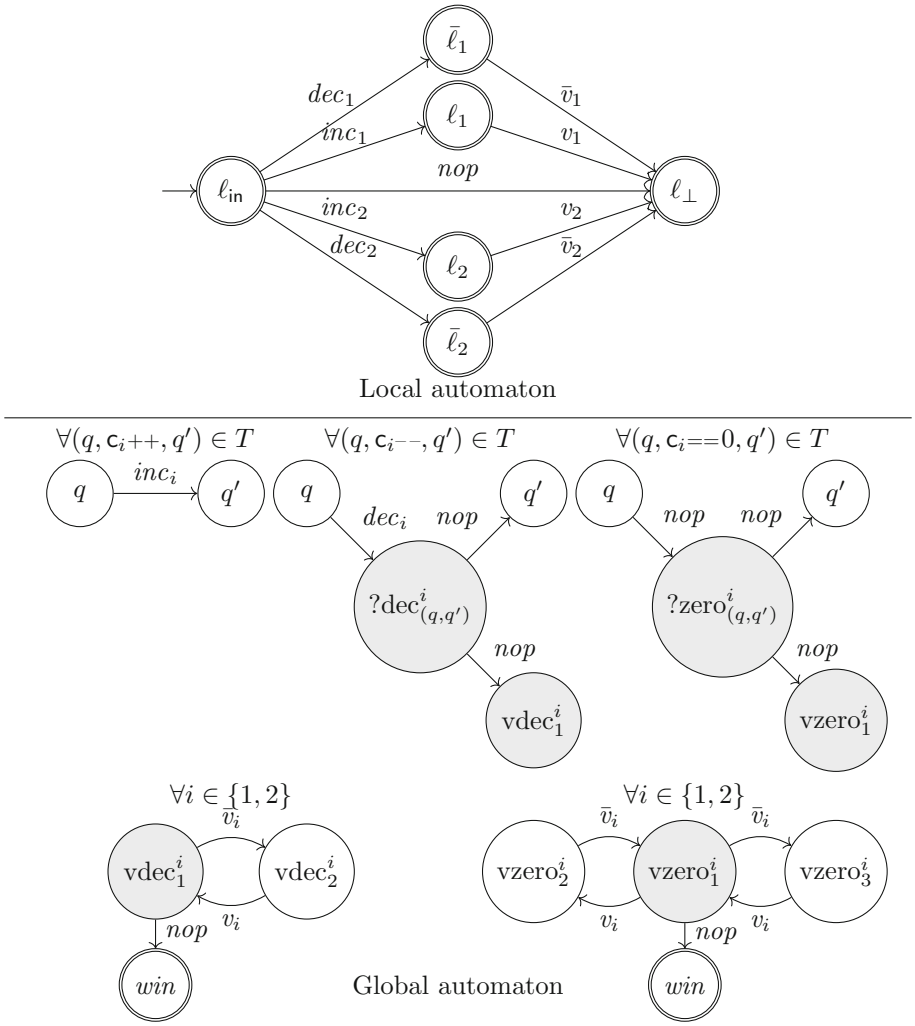


**Fig. 6** Construction of $\mathcal{G}^{\mathcal{P}}_{B\text{-cb}}$. Global states belonging to Player 1 are drawn with a light gray background. Not pictured: $q_0$ is the initial global state and $q_h$ is accepting

We define a strategy $f_{\mathrm{vdec}}$ as follows. If $c = (\mathrm{vdec}^i_2, \ell^1, \ldots, \ell^p)$ is a $\mathcal{G}^{\mathcal{P}}_{B\text{-cb}}$-configuration such that $m = \min_i(c)$ exists (that is, there is at least one process in state $\ell_i$), then $f_{\mathrm{vdec}}(c, P, p', 2) = ((\mathrm{vdec}^i_1, \hat{\ell}^1, \ldots, \hat{\ell}^p), P \cup \{m\}, m, 2)$ with $\hat{\ell}^m = \ell_\perp$, and $\hat{\ell}^j = \ell^j$ for all $j \neq m$. In all other cases, $f_{\mathrm{vdec}}$ gives an arbitrary successor node. Let $c$ be a configuration such that $st(c) = \mathrm{vdec}^i_1$ and $n_i(c) \geq 0$, that is there are at least as many processes in local state $\ell_i$ than in local state $\bar{\ell}_i$. Then it is easy to see that $f_{\mathrm{vdec}}$ is a winning strategy for Player 0 from node $(c, \{1, \ldots, p\}, p, 1)$, as there will always be at least one process in state $\ell_i$ for Player 0 to make a transition until Player 1 is forced to go from global state $\mathrm{vdec}^i_1$ to *win* because there are no more processes in state $\bar{\ell}_i$. Conversely, if $n_i(c) < 0$, then Player 0 cannot win from $(c, \{1, \ldots, p\}, p, 1)$ because Player 1 can force Player 0 to exhaust all processes in state $\ell_i$ until there are no more left and then be stuck in $\mathrm{vdec}^i_2$.

Similarly, one can build a strategy $f_{\text{vzero}}$ such that for all configuration $c$ with $st(c) = \text{vzero}_1^i$, Player 0 is winning from $(c, \{1, \ldots, p\}, p, 1)$ iff $n_i(c) = 0$.

We are now ready to prove Lemma 5.2.

$\boxed{\Rightarrow}$ Let $\rho = \gamma_0 \vdash_{t_1} \cdots \vdash_{t_k} \gamma_k$ be an accepting $M$-run. We define a (memoryless) strategy $f$ for Player 0 in $\mathcal{G}^{\mathcal{P}}_{B\text{-cb}}$ that simulates $\rho$ as follows: let $(c, P, p, 1)$ be a $\mathcal{G}^{\mathcal{P}}_{B\text{-cb}}$-configuration.

- If $(c, P, p, 1)$ is the last node of $\pi(\gamma_0 \vdash_{t_1} \cdots \vdash_{t_j} \gamma_j)$ for some $j \in \{0, \ldots, k-1\}$, then $f(c, P, p, 1)$ is its successor in $\pi(\rho)$.
- If $st(c) = \text{vdec}_2^i$ then $f$ follows $f_{\text{vdec}}$.
- If $st(c) = \text{vzero}_2^i$ or $st(c) = \text{vzero}_3^i$ then $f$ follows $f_{\text{vzero}}$.
- Otherwise $f$ gives an arbitrary successor.

Let $\pi$ be a maximal $f$-compatible play. There are two cases to study:

If global states $\text{vdec}_1^i$ and $\text{vzero}_1^i$ are not visited in $\pi$ for both $i = 1$ and $i = 2$, then necessarily $\pi = \pi(\rho)$. Since $\rho$ is an accepting $M$-run, then $\pi(\rho)$ is winning.

In the other case, suppose that $\text{vdec}_1^i$ is visited in $\pi$. Until visiting $\text{vdec}_1^i$, the play simulates a prefix of $\rho$. Then necessarily $\pi$ is of the form

$$\begin{aligned}\pi = &\pi(\gamma_0 \vdash_{t_1} \cdots \vdash_{t_j} \gamma_j) \cdot ((?\text{dec}^i_{(q,q')}, \ell^1, \ldots, \ell^p, \bar{\ell}_i), \{1, \ldots, p+1\}, p+1, 1)\\ &\cdot ((\text{vdec}_1^i, \ell^1, \ldots, \ell^p, \bar{\ell}_i, \ell_\perp), \{1, \ldots, p+2\}, p+2, 1) \cdot \pi'\end{aligned}$$

with $j < k$, $t_{j+1} = (q, \mathsf{c}_i{--}, q')$, and $\pi'$ is $f_{\text{vdec}}$-compatible by definition of $f$. Moreover, if $\gamma_j = (q, \nu_1, \nu_2)$, then $\pi(\gamma_0 \vdash_{t_1} \cdots \vdash_{t_j} \gamma_j)$ ends in $(c, \{1, \ldots, p\}, p, 1)$ with $c = (q, \ell^1, \ldots, \ell^p)$ and $n_i(c) = \nu_i > 0$, otherwise $t_{j+1}$ could not have been taken in $\rho$. Therefore, $n_i(\text{vdec}_1^i, \ell^1, \ldots, \ell^p, \bar{\ell}_i, \ell_\perp) = n_i(c) - 1 \geq 0$. Then because the rest of the play $\pi'$ follows $f_{\text{vdec}}$, we showed that $\pi$ is winning. Similarly, if $\text{vzero}_1^i$ is visited in $\pi$, we show that $\pi$ is winning.

Thus in any case $\pi$ is winning, and $f$ is a winning strategy for Player 0.

$\boxed{\Leftarrow}$ Let $f$ be a winning strategy of Player 0 in $\mathcal{G}^{\mathcal{P}}_{B\text{-cb}}$. We first show that if $\pi$ is an $f$-compatible play ending in $c = (?\text{dec}^i_{(q,q')}, \ell^1, \ldots, \ell^p)$, then $n_i(c) \geq 0$. Assume this is not the case and $n_i(c) < 0$. Then let $\pi'$ be a maximal $f$-compatible play such that $\pi' = \pi.((\text{vdec}_1^i, \ell^1, \ldots, \ell^p, \ell_\perp), \{1, \ldots, p, p+1\}, 1).\pi''$. Then, by construction, $\pi''$ ends in a configuration $c'$ such that $st(c') = \text{vdec}_2^i$ and no process in local state $\ell_i$, which is impossible because $f$ is a winning strategy. Similarly, if $\pi$ is an $f$-compatible play ending in $c = (?\text{zero}^i_{(q,q')}, \ell^1, \ldots, \ell^p)$, then $n_i(c) = 0$.

Now let $\pi$ be the $f$-compatible maximal play when put against the strategy of Player 1 that never goes to $\text{vdec}_1^i$ or $\text{vzero}_1^i$. Let $c_0, c_1, \ldots$ be the configurations visited during $\pi$. For all $j$ such that $st(c_j) = q$, one can build a valid $M$-run $\rho(c_0, \ldots, c_j)$ that ends in the $M$-configuration $\gamma = (q, n_1(c_j), n_2(c_j))$ in the following way: first we let $\rho(c_0) = \gamma_0 = (q_0, 0, 0)$, which satisfies the conditions above. Then if $\rho(c_0, \ldots, c_j) = \gamma_0 \vdash_{t_1} \cdots \vdash_{t_k} \gamma_k$ has been defined with $c_j = (q, \ell^1, \ldots, \ell^p)$ and $\gamma_k = (q, \nu_1, \nu_2)$ which satisfies the conditions, then there are three possible successors to consider:

- If $c_{j+1} = (q', \ell^1, \ldots, \ell^p, \ell_i)$ then there exists $t = (q, \mathsf{c}_i{++}, q') \in T$. We then define $\rho(c_0, \ldots, c_{j+1}) = \rho(c_0, \ldots, c_j) \vdash_t (q', \nu'_1, \nu'_2)$ with $\nu'_i = \nu_i + 1$ and $\nu'_{3-i} = \nu_{3-i}$ which satisfies the conditions as $n_i(c_{j+1}) = n_i(c_j) + 1 = \nu_i + 1 = \nu'_i$ and $n_{3-i}(c_{j+1}) = n_{3-i}(c_j) = \nu_{3-i} = \nu'_{3-i}$.
- If $c_{j+1} = (?\text{dec}^i_{(q,q')}, \ell^1, \ldots, \ell^p, \bar{\ell}_i)$ and $c_{j+2} = (q', \ell^1, \ldots, \ell^p, \bar{\ell}_i, \ell_\perp)$, then there exists $t = (q, \mathsf{c}_i{--}, q') \in T$. We define $\rho(c_0, \ldots, c_{j+2}) = \rho(c_0, \ldots, c_j) \vdash_t (q', \nu'_1, \nu'_2)$ with $\nu'_i = \nu_i - 1$ and $\nu'_{3-i} = \nu_{3-i}$. This is a valid $M$-run as $\nu_i > 0$, otherwise we would

have $n_i(c_{j+1}) = n_i(c_j) - 1 = \nu_i - 1 < 0$, which is impossible as showed hereabove. Moreover, $\nu'_i = n_i(c_{j+2})$ for $i \in \{1, 2\}$, so this run satisfies the required conditions.

- If $c_{j+1} = (?\text{zero}^i_{(q,q')}, \ell^1, \ldots, \ell^p, \ell_\perp)$ and $c_{j+2} = (q', \ell^1, \ldots, \ell^p, \ell_\perp, \ell_\perp)$, then there exists $t = (q, \mathsf{c}_i{=}{=}0, q') \in T$ and $\rho(c_0, \ldots, c_{j+2}) = \rho(c_0, \ldots, c_j) \vdash_t (q', \nu_1, \nu_2)$. Again this is a valid $M$-run because $\nu_i = n_i(c_{j+1}) = 0$, as showed hereabove. Since $\nu_i = n_i(c_j) = n_i(c_{j+2})$ for $i \in \{1, 2\}$, the conditions are also satisfied.

Therefore, $\pi$ is of the form $\pi(\rho)$ for some valid $M$-run $\rho$. As $\pi$ is winning, we deduce that $\rho$ is an accepting run of $M$. □

## 6 Conclusion

We extended the verification of round-bounded parameterized systems to a game-based setting, which allows us to model an uncontrollable environment. As games constitute an important approach to verifying branching-time properties (e.g., [31]), our results may be used for branching-time model checking of parameterized systems (using a variant of data logics [25] and a reduction of the model-checking problem to a parameterized pushdown game).

**Acknowledgements** This work is partly supported by ANR FREDDA (ANR-17-CE40-0013). One of the authors is, at the time of submission, working with the editor-in-chief, who will not be involved at any part of the evaluation process.

**Funding** Open access funding provided by University of Gothenburg.



## References

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

9. Bojańczyk M, David C, Muscholl A, Schwentick T, Segoufin L (2011) Two-variable logic on data words. ACM Trans Comput Log 12(4):27
10. Bollig Benedikt, Lehaut Mathieu, Sznajder Nathalie (2018) Round-bounded control of parameterized systems. In: 16th international symposium on automated technology for verification and analysis, Proceedings of ATVA'18, volume 11138 of Lecture notes in computer science, pages 370–386. Springer
11. Bollig Benedikt, Lehaut Mathieu, Sznajder Nathalie (2019) Round-bounded control of parameterized systems. Technical Report hal-01849206, HAL, March
12. Bouajjani Ahmed, Esparza Javier, Schwoon Stefan, Strejcek Jan (2005) Reachability analysis of multithreaded software with asynchronous communication. In: FSTTCS 2005: Foundations of software technology and theoretical computer science, 25th international conference, Hyderabad, India, December 15-18, 2005, Proceedings, volume 3821 of Lecture notes in computer science, pages 348–359. Springer
13. Brázdil T, Jancar P, Kucera A (2010) Reachability games on extended vector addition systems with states. In: ICALP'10, Part II, volume 6199 of LNCS, pp. 478–489. Springer
14. Brütsch B, Thomas W (2016) 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
15. Courtois J-B, Schmitz S (2014) Alternating vector addition systems with states. In: MFCS'14, volume 8634 of LNCS, pages 220–231. Springer
16. Wojciech Czerwinski, Slawomir Lasota, Ranko Lazic, Jérôme Leroux, Filip Mazowiecki (2021) The reachability problem for petri nets is not elementary. J ACM 68(1):7:1-7:28
17. Ehlers Rüdiger, Seshia Sanjit A, Kress-Gazit Hadas (2014) Synthesis with identifiers. In: international conference on verification, model checking, and abstract interpretation, pages 415–433. Springer
18. Emerson EA, Jutla CS (1991) Tree automata, mu-calculus and determinacy. In: proceedings of FOCS'91, pages 368–377. IEEE computer society
19. Emerson EA, Namjoshi KS (2003) On reasoning about rings. Int. J. Found. Comput. S. 14(4):527–550
20. Esparza J (2014) 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
21. Exibard Léo, Filiot Emmanuel, Reynier Pierre-Alain (2019) Synthesis of data word transducers. In: 30th international conference on concurrency theory
22. Figueira D., Praveen M (2018) Playing with repetitions in data words using energy games. In: proceedings of LICS'18, pages 404–413. ACM
23. Jacobs S, Bloem R (2014) Parameterized synthesis. Log Methods Comput Sci 10(1):151
24. Jancar P (2015) On reachability-related games on vector addition systems with states. In: RP'15, volume 9328 of LNCS, pages 50–62. Springer
25. Kara A (2016) Logics on data words: Expressivity, satisfiability, model checking. PhD thesis, Technical University of Dortmund
26. Khalimov Ayrat, Maderbacher Benedikt, Bloem Roderick (2018) Bounded synthesis of register transducers. In: international symposium on automated technology for verification and analysis, pages 494–510. Springer
27. La Torre S, Madhusudan P, Parlato G (2007) A robust class of context-sensitive languages. In: LICS'07, pages 161–170. IEEE Computer Society Press
28. Torre SLa, Madhusudan P., Parlato G (2008) Context-bounded analysis of concurrent queue systems. In: proceedings of TACAS'08, volume 4963 of LNCS, pp. 299–314. Springer
29. Torre SLa, Madhusudan P, Parlato G (2010) Model-checking parameterized concurrent programs using linear interfaces. In: CAV'10, volume 6174 of LNCS, pp. 629–644. Springer
30. Torre SLa, Madhusudan P, Parlato G (2010) Model-checking parameterized concurrent programs using linear interfaces. Technical report 2142/15410, University of Illinois, Available at http://hdl.handle.net/2142/15410
31. Lange M, Stirling C (2002) Model checking games for branching time logics. J Log Comput 12(4):623–639
32. Mayr Ernst W (1984) An algorithm for the general petri net reachability problem. SIAM J Comput 13(3):441–460
33. Minsky Marvin L (1967) Computation: finite and infinite machines. Prentice Hall, Upper Saddle River, NJ, USA
34. Qadeer S, Rehof J (2005) Context-bounded model checking of concurrent software. In: TACAS'05, volume 3440 of LNCS, pages 93–107. Springer
35. Ramalingam G (2000) Context-sensitive synchronization-sensitive analysis is undecidable. ACM Trans Program Lang Syst 22(2):416–430

36. Seth A (2009) Games on multi-stack pushdown systems. In: LFCS'09, volume 5407 of LNCS, pages 395–408. Springer
37. Stockmeyer LJ (1974) The complexity of decision problems in automata theory and logic. PhD thesis, MIT
38. Thomas W (1997) Languages, automata and logic. In: Salomaa A, Rozenberg G. editors, Handbook of formal languages, volume 3, pages 389–455. Springer
39. Walukiewicz I (2001) Pushdown processes: games and model-checking. Inf Comput 164(2):234–263
40. Zielonka W (1998) Infinite games on finitely coloured graphs with applications to automata on infinite trees. TCS 200(1–2):135–183