# Büchi Automata can have Smaller Quotients

Lorenzo Clemente

LFCS. School of Informatics. University of Edinburgh. UK

**Abstract.** We study novel simulation-like preorders for quotienting nondeterministic Büchi automata. We define *fixed-word delayed simulation*, a new preorder coarser than delayed simulation. We argue that fixed-word simulation is the coarsest forward simulation-like preorder which can be used for quotienting Büchi automata, thus improving our understanding of the limits of quotienting. Also, we show that computing fixed-word simulation is PSPACE-complete.
On the practical side, we introduce *proxy simulations*, which are novel polynomial-time computable preorders sound for quotienting. In particular, *delayed proxy simulation* induce quotients that can be smaller by an arbitrarily large factor than direct backward simulation. We derive proxy simulations as the product of a theory of refinement transformers: A *refinement transformer* maps preorders nondecreasingly, preserving certain properties. We study under which general conditions refinement transformers are sound for quotienting.

## 1 Introduction

Büchi automata minimization is an important topic in automata theory, both for the theoretical understanding of automata over infinite words and for practical applications. Minimizing an automaton means reducing the number of its states as much as possible, while preserving the recognized language. Minimal automata need not be unique, and their structure does not necessarily bear any resemblance to the original model; in the realm of infinite words, this holds even for deterministic models. This hints at why exact minimization has high complexity: Indeed, minimality checking is PSPACE-hard for nondeterministic models (already over finite words [12]), and NP-hard for deterministic Büchi automata [19]. Moreover, even approximating the minimal model is hard [8].

By posing suitable restrictions on the minimization procedure, it is nonetheless possible to trade exact minimality for efficiency. In the approach of *quotienting*, smaller automata are obtained by merging together equivalent states, under appropriately defined equivalences. In particular, quotienting by simulation equivalence has proven to be an effective heuristics for reducing the size of automata in cases of practical relevance.

The notion of *simulation preorder* and *equivalence* [17] is a crucial tool for comparing the behaviour of systems. It is best described via a game between two players, Duplicator and Spoiler, where the former tries to stepwise match the moves of the latter. But not every simulation preorder can be used for quotienting: We call a preorder *good for quotienting* (GFQ) if the quotient automaton (w.r.t. the induced equivalence) recognizes the same language as the original automaton. In particular, a necessary condition for a simulation to be GFQ is to take into account the acceptance condition: For example, in *direct* simulation [4], Duplicator has the additional requirement to visit an accepting state whenever Spoiler does so, while in the coarser *fair* simulation [10], Duplicator has to visit infinitely many accepting states if Spoiler does so. But, while direct simulation is GFQ [2], fair simulation is not [11].[1] This prompted the development of *delayed* simulation [6], a GFQ preorder intermediate between direct and fair simulation.

We study the border of GFQ preorders. In our first attempt we generalize delayed simulation to *delayed containment*. While in simulation the two players take turns in selecting transitions, in containment the game ends in one round: First Spoiler picks an infinite path, and then Duplicator has to match it with another infinite path. The winning condition is delayed-like: Every accepting state of Spoiler has to be matched by an accepting state of Duplicator, possibly occurring later. Therefore, in delayed containment Duplicator is much stronger than in simulation; in other words, containment is coarser than simulation. In fact, it is *too coarse*: We give a counterexample where delayed containment is not GFQ. We henceforth turn our attention to finer preorders.

In our second attempt, we remedy to the deficiency above by introducing *fixed-word delayed simulation*, an intermediate notion between simulation and containment. In fixed-word simulation, Spoiler does not reveal the whole path in advance like in containment; instead, she only declares the input word beforehand. Then, the simulation game starts, but now transitions can be taken only if they match the word fixed earlier by Spoiler. Unlike containment, fixed-word delayed simulation is GFQ, as we show.

We proceed by looking at even coarser GFQ preorders. We enrich fixed-word simulation by allowing Duplicator to use *multiple pebbles*, in the style of [5]. The question arises as whether Duplicator gains more power by "hedging her bets" when she already knows the input word in advance. By using an ordinal ranking argument (reminiscent of [15]), we establish that this is not the case, and that the multipebble hierarchy collapses to the 1-pebble case, i.e., to fixed-word delayed simulation itself. Incidentally, this also shows that the whole delayed multipebble hierarchy from [5] is entirely contained in fixed-word delayed simulation—the containment being strict.

For what concerns the complexity of computing fixed-word simulation, we establish that it is PSPACE-complete, by a mutual reduction from Büchi automata universality.

With the aim of getting tractable preorders, we then look at a different way of obtaining GFQ relations, by introducing a theory of refinement transformers: A *refinement transformer* maps a preorder $\preceq$ to a coarser preorder $\preceq'$, s.t., once $\preceq$ is known, $\preceq'$ can be computed with only a polynomial time overhead. The idea is to play a simulation-like game, where we allow Duplicator to "jump" to $\preceq$-bigger states, called *proxies*, after Spoiler has selected her transition. Duplicator can then reply with a transition from the proxy instead of the original state. We say that proxy states are *dynamic* in the sense that they depend on the transition selected by Spoiler.[2] Under certain conditions, we show that refinement transformers induce GFQ preorders.

Finally, we introduce *proxy simulations*, which are novel polynomial time GFQ preorders obtained by applying refinement transformers to a concrete preorder $\preceq$, namely, to *backward* direct simulation (called reverse simulation in [20]). We define two versions of proxy simulation, direct and delayed, the latter being coarser than the former, and both coarser than direct backward simulation. Moreover, we show that the delayed variant can achieve quotients smaller than direct proxy simulation by an arbitrarily large factor. Full proofs can be found in the appendix.

[1] In fact, for Büchi automata it is well-known that also language equivalence is not GFQ.
[2] Proxies are strongly related to mediators [1]. We compare them in depth in Section 6.

*Related work.* Delayed simulation [6] has been extended to generalized automata [13], to multiple pebbles [5], to alternating automata [7] and to the combination of the last two [3]. Fair simulation has been used for state space reduction in [9]. The abstract idea of mixing forward and backward modes in quotienting can be traced back at least to [18]; in the context of alternating automata, it has been studied in [1].

## 2 Preliminaries

*Games.* For a finite sequence $\pi = e_0 e_1 \cdots e_{k-1}$, let $|\pi| = k$ be its length, and let $\mathrm{last}(\pi) = e_{k-1}$ be its last element. If $\pi$ is infinite, then take $|\pi| = \omega$.

A *game* is a tuple $G = (P, P_0, P_1, p_I, \Gamma, \Gamma_0, \Gamma_1, W)$, where $P$ is the set of positions, partitioned into disjoint sets $P_0$ and $P_1$, $p_I \in P_0$ is the initial position, $\Gamma = \Gamma_0 \cup \Gamma_1$ is the set of moves, where $\Gamma_0 \subseteq P_0 \times P_1$ and $\Gamma_1 \subseteq P_1 \times P_0$ are the set of moves of Player 0 and Player 1, respectively, and $W \subseteq P_0^\omega$ is the winning condition. A *path* is a finite or infinite sequence of states $\pi = p_0^0 p_0^1 p_1^0 p_1^1 \cdots$ starting in $p_I$, such that, for all $i < |\pi|$, $(p_i^0, p_i^1) \in \Gamma_0$ and $(p_i^1, p_{i+1}^0) \in \Gamma_1$. *Partial plays* and *plays* are finite and infinite paths, respectively. We assume that there are no dead ends in the game. A play is *winning* for Player 1 iff $p_0^0 p_1^0 p_2^0 \cdots \in W$; otherwise, is it winning for Player 0.

A *strategy* for Player 0 is a partial function $\sigma_0 : (P_0 P_1)^* P_0 \mapsto P_1$ s.t., for any partial play $\pi \in (P_0 P_1)^* P_0$, if $\sigma_0$ is defined on $\pi$, then $\pi \cdot \sigma_0(\pi)$ is again a partial play. A play $\pi$ is $\sigma_0$-*conform* iff, for every $i \geq 0$, $p_i^1 = \sigma_0(p_0^0 p_0^1 \cdots p_i^0)$. Similarly, a strategy for Player 1 is a partial function $\sigma_1 : (P_0 P_1)^+ \mapsto P_0$ s.t., for any partial play $\pi \in (P_0 P_1)^+$, if $\sigma_1$ is defined on $\pi$, then $\pi \cdot \sigma_1(\pi)$ is again a partial play. A play $\pi$ is $\sigma_1$-conform iff, for every $i \geq 0$, $p_{i+1}^0 = \sigma_0(p_0^0 p_0^1 \cdots p_i^0 p_i^1)$. While we do not require strategies to be total functions, we do require that a strategy $\sigma$ is defined on all $\sigma$-conform partial plays.

A strategy $\sigma_i$ is a *winning strategy* for Player $i$ iff all $\sigma_i$-conform plays are winning for Player $i$. We say that Player $i$ wins the game $G$ if she has a winning strategy.

*Automata.* A *nondeterministic Büchi automaton* (NBA) is a tuple $\mathcal{Q} = (Q, \Sigma, I, \Delta, F)$, where $Q$ is a finite set of states, $\Sigma$ is a finite alphabet, $I \subseteq Q$ is the set of initial states, $F \subseteq Q$ is the set of final states and $\Delta \subseteq Q \times \Sigma \times Q$ is the transition relation. We also write $q \xrightarrow{a} q'$ instead of $(q, a, q') \in \Delta$, and just $q \longrightarrow q'$ when $\exists a \in \Sigma \cdot q \xrightarrow{a} q'$. For two sets of states $\mathbf{q}, \mathbf{q}' \subseteq Q$, we write $\mathbf{q} \xRightarrow{a} \mathbf{q}'$ iff $\forall q' \in \mathbf{q}' \cdot \exists q \in \mathbf{q} \cdot q \xrightarrow{a} q'$.[3] For a state $q \in Q$, let $[q \in F] = 1$ if $q$ is accepting, and 0 otherwise. We assume that every state is reachable from some initial state, and that the transition relation is total.

For a finite or infinite sequence of states $\rho = q_0 q_1 \cdots$ and an index $i \leq |\rho|$, let $\mathsf{cnt\text{-}final}(\rho, i)$ be the number of final states occuring in $\rho$ up to (and including) the $i$-th element. Formally, $\mathsf{cnt\text{-}final}(\rho, i) = \sum_{0 \leq k < i} [q_k \in F]$, with $\mathsf{cnt\text{-}final}(\rho, 0) = 0$. Let $\mathsf{cnt\text{-}final}(\rho) = \mathsf{cnt\text{-}final}(\rho, |\rho|)$. If $\rho$ is infinite, then $\mathsf{cnt\text{-}final}(\rho) = \omega$ iff $\rho$ contains infinitely many accepting states.

Fix a finite or infinite word $w = a_0 a_1 \cdots$. A *path* $\pi$ *over* $w$ is a sequence $q_0 \xrightarrow{a_0} q_1 \xrightarrow{a_1} q_2 \cdots$ of length $|w|+1$. A path is *initial* if it starts in an initial state $q_0 \in I$, it is a *run* if it is initial and infinite, and it is *fair* if $\mathsf{cnt\text{-}final}(\pi) = \omega$. An *accepting run* is a run which is fair. The *language* $\mathcal{L}^\omega(\mathcal{Q})$ of a NBA $\mathcal{Q}$ is the set of infinite words which admit an accepting run, i.e., $\mathcal{L}^\omega(\mathcal{Q}) = \{w \in \Sigma^\omega \mid \text{ there exists an accepting run } \pi \text{ over } w\}$.

[3] This kind of backward-compatible transition had already appeared in [16].

*Quotients.* Let $\mathcal{Q} = (Q, \Sigma, I, \Delta, F)$ be a NBA and let $R$ be any binary relation on $Q$. We say that $\approx_R$ is the *equivalence induced by* $R$ if $\approx_R$ is the largest equivalence contained in the transitive and reflexive closure of $R$. I.e., $\approx_R = R^* \cap (R^*)^{-1}$. Let the function $[\cdot]_R : Q \mapsto 2^Q$ map each element $q \in Q$ to the equivalence class $[q]_R \subseteq Q$ it belongs to, i.e., $[q]_R := \{q' \in Q \mid q \approx_R q'\}$. We overload $[P]_R$ on sets $P \subseteq Q$ by taking the set of equivalence classes. When clear from the context, we avoid noting the dependence of $\approx$ and $[\cdot]$ on $R$.

An equivalence $\approx$ on $\mathcal{Q}$ induces the *quotient automaton* $\mathcal{Q}_\approx = ([Q], \Sigma, [I], \Delta_\approx, [F])$, where, for any $q, q' \in Q$ and $a \in \Sigma$, $([q], a, [q']) \in \Delta_\approx$ iff $(q, a, q') \in \Delta$. This is called a *naïve* quotient since both initial/final states and transitions are induced representative-wise. When we quotient w.r.t. a relation $R$ which is not itself an equivalence, we actually mean quotenting w.r.t. the induced equivalence $\approx$. We say that $R$ is *good for quotienting* (GFQ) if quotienting $\mathcal{Q}$ w.r.t. $R$ preserves the language, that is, $\mathcal{L}^\omega(\mathcal{Q}) = \mathcal{L}^\omega(\mathcal{Q}_\approx)$.

**Lemma 1.** *For two equivalences $\approx_0, \approx_1$, if $\approx_0 \subseteq \approx_1$, then $\mathcal{L}^\omega(\mathcal{Q}_{\approx_0}) \subseteq \mathcal{L}^\omega(\mathcal{Q}_{\approx_1})$. In particular, by letting $\approx_0$ be the identity, $\mathcal{L}^\omega(\mathcal{Q}) \subseteq \mathcal{L}^\omega(\mathcal{Q}_{\approx_1})$.*

# 3 Quotienting with forward simulations

In this section we study several generalizations of delayed simulation, in order to investigate the border of good for quotienting (GFQ) forward-like preorders. In our first attempt we introduce *delayed containment*, which is obtained as a modification of the usual simulation interaction between players: In the delayed containment game between $q$ and $s$ there are only two rounds. Spoiler moves first and selects both an infinite word $w = a_0 a_1 \cdots$ and an infinite path $q_0 \xrightarrow{a_0} q_1 \xrightarrow{a_1} \cdots$ over $w$ starting in $q = q_0$; then, Duplicator replies with an infinite path $s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} \cdots$ over $w$ starting in $s = s_0$. The winning condition is delayed-like: $\forall i \cdot q_i \in F \implies \exists j \geq i \cdot s_j \in F$. If Duplicator wins the delayed containment game between $q$ and $s$, we write $q \subseteq^{\mathrm{de}} s$. Clearly, $\subseteq^{\mathrm{de}}$ is a preorder implying language containment. One might wonder whether delayed-containment is GFQ. Unfortunately, this is not the case (see Figure 5 in the Appendix). Therefore, $\subseteq^{\mathrm{de}}$ is too coarse for quotienting, and we shall look at finer relations.

**Lemma 2.** *$\subseteq^{de}$ is not a GFQ preorder.*

## 3.1 Fixed-word delayed simulation

Our second attempt at generalizing delayed simulation still retains the flavour of containment. While in containment $\subseteq^{\mathrm{de}}$ Spoiler reveals both the input word $w$ and a path over $w$, in *fixed-word* simulation $\sqsubseteq^{\mathrm{de}}_{\mathrm{fx}}$ Spoiler reveals $w$ only. Then, after $w$ has been fixed, the game proceeds like in delayed simulation, with the proviso that transitions match symbols in $w$.[4] Formally, let $w = a_0 a_1 \cdots \in \Sigma^\omega$. In the $w$-simulation game $G^{\mathrm{de}}_w(q, s)$ the set of positions of Spoiler is $P_0 = Q \times Q \times \mathbb{N}$, the set of positions of Duplicator is $P_1 = Q \times Q \times Q \times \mathbb{N}$ and $\langle q, s, 0\rangle$ is the initial position. Transitions are determined as follows: Spoiler can select a move of the form $(\langle q, s, i\rangle, \langle q, s, q', i\rangle) \in \Gamma_0^{w\text{-de}}$

[4] The related notion of fixed-word *fair* simulation clearly coincides with $\omega$-language inclusion.

if $q \xrightarrow{a_i} q'$, and Duplicator can select a move of the form $(\langle q, s, q', i\rangle, \langle q', s', i+1\rangle) \in \Gamma_1^{w\text{-de}}$ if $s \xrightarrow{a_i} s'$. Notice that the input symbol $a_i$ is fixed, and it has to match the corresponding symbol in $w$. The winning condition is $W = \{\langle q_0, s_0, 0\rangle\langle q_1, s_1, 1\rangle \cdots \mid \forall i \cdot q_i \in F \implies \exists j \geq i \cdot s_j \in F\}$. Let $q \sqsubseteq_w^{\text{de}} s$ iff Duplicator wins the $w$-simulation game $G_w^{\text{de}}(q, s)$, and $q \sqsubseteq_{\text{fx}}^{\text{de}} s$ iff $q \sqsubseteq_w^{\text{de}} s$ for all $w \in \Sigma^\omega$. Clearly, fixed-word simulation is a preorder implying containment.

**Fact 1.** *$\sqsubseteq_{fx}^{de}$ is a reflexive and transitive relation, and $\forall q, s \in Q \cdot q \sqsubseteq_{fx}^{de} s \implies q \subseteq^{de} s$.*

Unlike delayed containment, fixed-word delayed simulation is GFQ. Moreover, fixed-word delayed simulation quotients can be more succint than (multipebble) delayed simulation quotients by an arbitrarily large factor. See Figure 6 in the Appendix.

**Theorem 1.** *$\sqsubseteq_{fx}^{de}$ is good for quotienting.*

*Complexity of delayed fixed word simulation.* Let $q, s$ be two states in $\mathcal{Q}$. We reduce the problem of checking $q \sqsubseteq_{\text{fx}}^{\text{de}} s$ to the universality problem of a suitable alternating Büchi product automaton (ABA) $\mathcal{A}$. We design $\mathcal{A}$ to accept exactly those words $w$ s.t. Duplicator wins $G_w^{\text{de}}(q, s)$. Then, by the definition of $\sqsubseteq_{\text{fx}}^{\text{de}}$, it is enough to check whether $\mathcal{A}$ has universal language. See [21] (or Appendix A.1) for background on ABAs.

The idea is to enrich configurations in the fixed-word simulation game by adding an obligation bit recording whether Duplicator has any pending constraint to visit an accepting state. Initially the bit is 0, and it is set to 1 whenever Spoiler is accepting; a reset to 0 can occur afterwards, if and when Duplicator visits an accepting state.

Let $\mathcal{Q} = (Q, \Sigma, I, \Delta, F)$ be a NBA. We define a product ABA $\mathcal{A} = (A, \Sigma, \delta, \alpha)$ as follows: The set of states is $A = Q \times Q \times \{0, 1\}$, final states are of the form $\alpha = Q \times Q \times \{0\}$ and, for any $\langle q, s, b\rangle \in A$ and $a \in \Sigma$,

$$\delta(\langle q, s, b\rangle, a) = \bigwedge_{q \xrightarrow{a} q'} \bigvee_{s \xrightarrow{a} s'} \langle q', s', b'\rangle, \qquad \text{where } b' = \begin{cases} 0 & \text{if } s \in F \\ 1 & \text{if } q \in F \wedge s \notin F \\ b & \text{otherwise} \end{cases}$$

It follows directly from the definitions that $q \sqsubseteq_{\text{fx}}^{\text{de}} s$ iff $\mathcal{L}^\omega(\langle q, s, 0\rangle) = \Sigma^\omega$. A reduction in the other direction is immediate already for NBAs: In fact, an NBA $\mathcal{Q}$ is universal iff $\mathcal{U} \sqsubseteq_{\text{fx}}^{\text{de}} \mathcal{Q}$, where $\mathcal{U}$ is the trivial, universal one-state automaton with an accepting $\Sigma$-loop. It is well-known that universality is PSPACE-complete for ABAs/NBAs [14].

**Theorem 2.** *Computing fixed-word delayed simulation is PSPACE-complete.*

### 3.2 Multipebble fixed-word delayed simulation

Having established that fixed-word simulation is GFQ, the next question is whether we can find other natural GFQ preorders between fixed-word and delayed containment. A natural attempt is to add a multipebble facility on top of $\sqsubseteq_{\text{fx}}^{\text{de}}$. Intuitively, when Duplicator uses multiple pebbles she can "hedge her bets" by moving pebbles to several successors. This allows Duplicator to delay committing to any particular choice by arbitrarily many steps: In particular, she can always gain knowledge on any *finite* number of moves by Spoiler. Perhaps surprisingly, we show that *Duplicator does not gain more* *power by using pebbles*. This is stated in Theorem 3, and it is the major technical result of this section. It follows that, once Duplicator knows the input word in advance, there is no difference between knowing only the next step by Spoiler, or the next $l$ steps, for any finite $l > 1$. Yet, if we allow $l = \omega$ lookahead, then we recover delayed containment $\subseteq^{\mathrm{de}}$, which is not GFQ by Lemma 2. Therefore, w.r.t. to the degree of lookahead, $\sqsubseteq^{\mathrm{de}}_{\mathrm{fx}}$ is the coarsest GFQ relation included in $\subseteq^{\mathrm{de}}$.

We now define the multipebble fixed-word delayed simulation. Let $k \geq 1$ and $w = a_0 a_1 \cdots \in \Sigma^\omega$. In the $k$-multipebble $w$-delayed simulation game $G^{k\text{-de}}_w(q, s)$ the set of positions of Spoiler is $Q \times 2^Q \times \mathbb{N}$, the set of positions of Duplicator is $Q \times 2^Q \times Q \times \mathbb{N}$, the initial position is $\langle q, \{s\}, 0\rangle$, and transitions are: $(\langle q, \mathbf{s}, i\rangle, \langle q, \mathbf{s}, q', i\rangle) \in \Gamma_0$ iff $q \xrightarrow{a_i} q'$, and $(\langle q, \mathbf{s}, q', i\rangle, \langle q', \mathbf{s}', i+1\rangle) \in \Gamma_1$ iff $\mathbf{s} \overset{a_i}{\Longrightarrow} \mathbf{s}'$ and $|\mathbf{s}'| \leq k$.

Before defining the winning set we need some preparation. Given an infinite sequence $\pi = \langle q_0, \mathbf{s}_0, 0\rangle\langle q_1, \mathbf{s}_1, 1\rangle \cdots$ over $w = a_0 a_1 \cdots$ and a round $j \geq 0$, we say that a state $s \in \mathbf{s}_j$ *has been accepting* since some previous round $i \leq j$, written $\mathsf{accepting}^i_j(s, \pi)$, iff either $s \in F$, or $i < j$ and there exists $\hat{s} \in \mathbf{s}_{j-1}$ s.t. $\hat{s} \xrightarrow{a_{j-1}} s$ and $\mathsf{accepting}^i_{j-1}(\hat{s}, \pi)$. We say that $\mathbf{s}_j$ is *good since round* $i \leq j$, written $\mathsf{good}^i_j(\mathbf{s}_j, \pi)$, iff at round $j$ every state $s \in \mathbf{s}_j$ has been accepting since round $i$, and $j$ is the least round for which this holds [5]. Duplicator wins a play if, whenever $q_i \in F$ there exists $j \geq i$ s.t. $\mathsf{good}^i_j(\mathbf{s}_j, \pi)$. We write $q \sqsubseteq^{k\text{-de}}_w s$ iff Duplicator wins $G^{k\text{-de}}_w(q, s)$, and we write $q \sqsubseteq^{k\text{-de}}_{\mathrm{fx}} s$ iff $\forall w \in \Sigma^\omega \cdot q \sqsubseteq^{k\text{-de}}_w s$.

Clearly, pebble simulations induce a non-decreasing hierarcy: $\sqsubseteq^{1\text{-de}}_{\mathrm{fx}} \subseteq \sqsubseteq^{2\text{-de}}_{\mathrm{fx}} \subseteq \cdots$. We establish that the hierarchy actually collapses to the $k = 1$ level. This result is non-trivial, since the delayed winning condition requires reasoning not only about the *possibility* of Duplicator to visit accepting states in the future, but also about exactly *when* such a visit occurs. Technically, our argument uses a ranking argument similar to [15] (see Appendix A.2), with the notable difference that our ranks are *ordinals* ($\leq \omega^2$), instead of natural numbers. We need ordinals to represent how long a player can delay visiting accepting states, and how this events nest with each other. Finally, notice that the result above implies that the multipebble delayed simulation hierarchy of [5] is entirely contained in $\sqsubseteq^{\mathrm{de}}_{\mathrm{fx}}$, and the containment is strict (Fig. 6 in the appendix).

**Theorem 3.** *For any NBA $\mathcal{Q}$, $k \geq 1$ and states $q, s \in Q$, $q \sqsubseteq^{k\text{-de}}_{\mathit{fx}} s$ iff $q \sqsubseteq^{\mathit{de}}_{\mathit{fx}} s$.*

## 4 Jumping-safe relations

In this section we present the general technique which is used throughout the paper to establish that preorders are GFQ. We introduce *jumping-safe relations*, which are shown to be GFQ (Theorem 4). In Section 5 we use jumping-safety as an invariant when applying refinement transformers. We start off with an analysis of acceping runs.

*Coherent sequences of paths.* Fix an infinite word $w \in \Sigma^\omega$. Let $\Pi := \pi_0, \pi_1, \ldots$ be an infinite sequence of longer and longer *finite* initial paths in $\mathcal{Q}$ over (prefixes of) $w$. We are interested in finding a sufficient condition for the existence of an accepting run over $w$. A necessary condition is that the number of final states in $\pi_i$ grows unboundedly as $i$ goes to $\omega$. In the case of deterministic automata this condition is also sufficient: Indeed, in a deterministic automaton there exists a unique run over $w$, which is accepting exactly when the number of accepting stated visited by its prefixes goes to infinity. In this case, we say that the $\pi_i$'s are *strongly coherent* since they next path extends the previous one. Unfortunately, in the general case of nondeterministic automata it is quite possible to have paths that visit arbitrarily many final states but no accepting run exists. This occurs because final states can appear arbitrarily late. Indeed, consider Figure 1. Take $w = aba^2ba^3b\cdots$: For every prefx $w_i = aba^2b\cdots a^i$ there exists a path $\pi_i = qq\cdots q \cdot s^i$ over $w_i$ visiting a final state $i$ times. Still, $w \notin \mathcal{L}^\omega(\mathcal{Q})$.

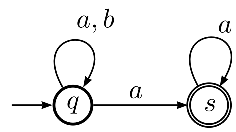



**Fig. 1.** Automaton $\mathcal{Q}$.

Therefore, we forbid accepting states to "clump away" in the tail of the path. We ensure this by imposing the existence of an infinite sequence of indices $j_0, j_1, \cdots$ s.t., for all $i$, and for all $k_i$ big enough, the number of final states in $\pi_{k_i}$ up to the $j_i$-th state is at least $i$. In this way, we are guaranteed that at least $i$ final states are present within $j_i$ steps in all but finitely many paths.

**Definition 1.** *Let $\Pi := \pi_0, \pi_1, \ldots$ be an infinite sequence of finite paths. We say that $\Pi$ is a* coherent sequence of paths *if the following property holds:*

$$\forall i \cdot \exists j \cdot \exists h \cdot \forall k \geq h \cdot j < |\pi_k| \wedge \mathsf{cnt\text{-}final}(\pi_k, j) \geq i \; . \tag{1}$$

**Lemma 3.** *If $\Pi$ is coherent, then any infinite subsequence $\Pi'$ thereof is coherent.*

We sketch below the proof that coherent sequences induce fair paths. Let $\Pi = \pi_0, \pi_1, \ldots$ be a coherent sequence of paths in $\mathcal{Q}$. Let $i = 1$, and let $j_1$ be the index witnessing $\Pi$ is coherent. Since the $\pi_k$'s are branches in a finitely branching tree, there are only a finite number of different prefixes of length $j_1$. Therefore, there exists a prefix $\rho_1$ which is common to infinitely many paths. Let $\Pi' = \pi'_0, \pi'_1, \ldots$ be the infinite subsequence of $\Pi$ containing only suffixes of $\rho_1$. Clearly $\rho_1$ contains at least 1 final state, and each $\pi'$ in $\Pi'$ extends $\rho_1$. By Lemma 3, $\Pi'$ is coherent. For $i = 2$, we can apply the reasoning again to $\Pi'$, and we obtain a longer prefix $\rho_2$ extending $\rho_1$, and containing at least 2 final states. Let $\Pi''$ be the coherent subsequence of $\Pi'$ containing only suffixes of $\rho_2$. In this fashion, we obtain an infinite sequence of *strongly* coherent (finite) paths $\rho_1, \rho_2, \cdots$ s.t. $\rho_i$ extends $\rho_{i-1}$ and contains at least $i$ final states. The infinite path to which the sequence converges is the fair path we are after.

**Lemma 4.** *Let $w \in \Sigma^\omega$ and $\pi_0, \pi_1, \ldots$ as above. If $\pi_0, \pi_1, \ldots$ is coherent, then there exists a fair path $\rho$ over $w$. Moreover, if all $\pi_i$'s are initial, then $\rho$ is initial.*

*Jumping-safe relations.* We established that coherent sequences induce accepting paths. Next, we introduce *jumping-safe* relations, which are designed to induce coherent sequences (and thus accepting paths) when used in quotienting. The idea is to view a path in the quotient automaton as a jumping path in the original automaton, where a "jumping path" is one that can take arbitrary jumps to equivalent states. Jumping-safe relations allows us to transform the sequence of prefixes of an accepting jumping path into a coherent sequence of non-jumping paths; by Lemma 4, this induces a (nonjumping) accepting path.

Fix a word $w = a_0 a_1 \cdots \in \Sigma^\omega$, and let $R$ be a binary relation over $Q$. An $R$-*jumping path* is an infinite sequence

$$\pi = q_0 \; R \; q_0^F \; R \; \hat{q}_0 \xrightarrow{a_0} q_1 \; R \; q_1^F \; R \; \hat{q}_1 \xrightarrow{a_1} q_2 \cdots , \tag{2}$$

and we say that $\pi$ is *initial* if $q_0 \in I$, and *fair* if $q_i^F \in F$ for infinitely many $i$'s.

**Definition 2.** *A binary relation $R$ is* jumping-safe *iff for any initial $R$-jumping path $\pi$ there exists an infinite sequence of initial finite paths $\pi_0, \pi_1, \ldots$ over suitable prefixes of $w$ s.t. $\mathrm{last}(\pi_i) \; R \; q_i$ and, if $\pi$ is fair, then $\pi_0, \pi_1, \ldots$ is coherent.*

**Theorem 4.** *Jumping-safe preorders are good for quotienting.*

In Section 5 we introduce refinement transformers, which are designed to preserve jumping-safety. Then, in Section 6 we specialize the approach to *backward direct simulation* $\sqsubseteq^{\mathrm{di}}_{\mathrm{bw}}$ [20], which provides an initial jumping-safe preorder, and which we introduce next: $\sqsubseteq^{\mathrm{di}}_{\mathrm{bw}}$ is the coarsest preorder s.t. $q \sqsubseteq^{\mathrm{di}}_{\mathrm{bw}} s$ implies 1) $\forall (q' \xrightarrow{a} q) \cdot \exists (s' \xrightarrow{a} s) \cdot q' \sqsubseteq^{\mathrm{di}}_{\mathrm{bw}} s'$, 2) $q \in F \implies s \in F$, and 3) $q \in I \implies s \in I$.

**Fact 2.** *$\sqsubseteq^{di}_{bw}$ is jumping-safe and computable in polynomial time.*

## 5 Refinement transformers

We study how to obtain GFQ preorders coarser than forward/backward simulation. As a preliminary example, notice that it is not possible to generalize simultaneously both forward and backward simulations. See the counterexample in Fig. 2, where any relation coarser than both forward and backward simulation is not GFQ. Let $\approx^{\mathrm{di}}_{\mathrm{bw}}$ and $\approx^{\mathrm{di}}_{\mathrm{fw}}$ be backward and forward direct simulation equivalence, respectively. We have $q_1 \approx^{\mathrm{di}}_{\mathrm{bw}} q_2 \approx^{\mathrm{di}}_{\mathrm{fw}} q_3$, but "glueing together" $q_1, q_2, q_3$ would introduce the extraneous word $ba^\omega$. Therefore, one needs to choose whether to extend either forward or backward simulation. The former approach has been pursued in the *mediated preorders* of [1] (in the more general context of alternating automata). Here, we extend backward refinements.

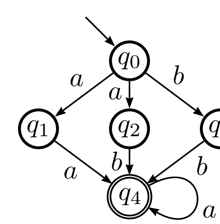



**Fig. 2.**

We define a *refinement transformer* $\tau_0$ mapping a relation $R$ to a new, coarser relation $\tau_0(R)$. We present $\tau_0$ via a forward direct simulation-like game where Duplicator is allowed to "jump" to $R$-bigger states—called *proxies*. Formally, in the $\tau_0(R)$ simulation game Spoiler's positions are in $Q \times Q$, Duplicator's position are in $Q \times Q \times \Sigma \times Q$ and transitions are as follows: Spoiler picks a transition $(\langle s, q \rangle, \langle s, q, a, q' \rangle) \in \Gamma_0$ simply when $q \xrightarrow{a} q'$, and Duplicator picks a transition $(\langle s, q, a, q' \rangle, \langle s', q' \rangle) \in \Gamma_1$ iff there exists a proxy $\hat{s}$ s.t. $s \; R \; \hat{s}$ and $\hat{s} \xrightarrow{a} s'$. The winning condition is: $\forall i \geq 0 \cdot q_i \in F \implies \hat{s}_i \in F$. If Duplicator wins starting from the initial position $\langle s, q \rangle$, we write $s \; \tau_0(R) \; q$. (Notice that we swapped the usual order between $q$ and $s$ here.)

**Lemma 5.** *For a preorder $R$, $R \subseteq R \circ \tau_0(R) \subseteq \tau_0(R)$.*

Unfortunately, $\tau_0(R)$ is not necessarily a transitive relation. Therefore, it is not immediately clear how to define a suitable equivalence for quotienting. Figure 2 shows that taking the transitive closure of $\tau_0(R)$ is incorrect—already when $R$ is direct backward simulation $\sqsubseteq^{\mathrm{di}}_{\mathrm{bw}}$: Let $\preceq = \tau_0(\sqsubseteq^{\mathrm{di}}_{\mathrm{bw}})$ and let $\approx = \preceq \cap \preceq^{-1}$. We have $q_3 \approx q_2 \approx q_1 \ \preceq \ q_3$, but $q_3 \not\preceq q_1$, and forcing $q_1 \approx q_3$ is incorrect, as noted earlier.

Thus, $\tau_0(R)$ is not GFQ and we need to look at its transitive fragments. Let $T \subseteq \tau_0(R)$. We say that $R$ is *$F$-respecting* if $q \ R \ s \wedge q \in F \implies s \in F$, that $T$ is *self-respecting* if Duplicator wins by never leaving $T$, that $T$ is *appealing* if transitive and self-respecting, and that $T$ *improves on* $R$ if $R \subseteq T$.

**Theorem 5.** *Let $R$ a $F$-respecting preorder, and let $T \subseteq \tau_0(R)$ be an appealing, improving fragment of $\tau_0(R)$. If $R$ is jumping-safe, then $T$ is jumping-safe.*

In particular, by Theorem 4, $T$ is GFQ. Notice that requiring that $R$ is GFQ is not sufficient here, and we need the stronger invariant given by jumping-safety.

Given an appealing fragment $T \subseteq \tau_0(R)$, a natural question is whether $\tau_0(T)$ improves on $\tau_0(R)$, so that $\tau_0$ can be applied repeatedly to get bigger and bigger preorders. We see in the next lemma that this is not the case.

**Lemma 6.** *For any reflexive $R$, let $T \subseteq \tau_0(R)$ be any appealing fragment of $\tau_0(R)$. Then, $\tau_0(T) \subseteq \tau_0(R)$.*

*Efficient appealing fragments.* By Theorems 4 and 5, appealing fragments of $\tau_0$ are GFQ. Yet, we have not specified any method for obtaining these. Ideally, one looks for fragments having maximal cardinality (which yelds maximal reduction under quotienting), but finding them is computationally expensive. Instead, we define a new transformer $\tau_1$ which is guaranteed to produce only appealing fragments,[5] which, while not maximal in general, are maximal amongst all *improving* fragments (Lemma 7).

The reason why $\tau_0(R)$ is not transitive is that only Duplicator is allowed to make "$R$-jumps". This asymmetry is an obstacle to compose simulation games. We recover transitivity by allowing Spoiler to jump as well, thus restoring the symmetry. Formally, the $\tau_1(R)$ simulation game is identical to the one for $\tau_0(R)$, the only difference being that also Spoiler is now allowed to "jump", i.e., she can pick a transition $(\langle s, q\rangle, \langle s, q, a, q'\rangle) \in \Gamma_0$ iff there exists $\hat{q}$ s.t. $q \ R \ \hat{q}$ and $\hat{q} \xrightarrow{a} q'$. The winning condition is: $\forall i \geq 0 \cdot \hat{q}_i \in F \implies \hat{s}_i \in F$. Let $s \ \tau_1(R) \ q$ if Duplicator wins from position $\langle s, q\rangle$. It is immediate to see that $\tau_1(R)$ is an appealing fragment of $\tau_0(R)$, and that $\tau_1$ is improving on transitive relations $R$'s. Thus, for a preorder $R$, $R \subseteq \tau_1(R) \subseteq \tau_0(R)$. By Theorems 4 and 5, $\tau_1(R)$ is GFQ (if $R$ is $F$-respecting).

It turns out that $\tau_1(R)$ is actually the *maximal* appealing, improving fragment of $\tau_0(R)$. This is non-obvious, since the class of appealing $T$'s is not closed under union—still, it admits a maximal element. Therefore, $\tau_1$ is an optimal solution to the problem of finding appealing, improving fragments of $\tau_0(R)$.

**Lemma 7.** *For any $R$, let $T \subseteq \tau_0(R)$ be any appealing fragment of $\tau_0(R)$. If $R \subseteq T$ (i.e., $R$ is improving), then $T \subseteq \tau_1(R)$.*

---

[5] $\tau_1$ needs not be the only solution to this problem: Other ways of obtaining appealing fragments of $\tau_0$ might exist. For this reason, we have given a separate treatment of $\tau_0$ in its generality, together with the general correctness statement (Theorem 5).

### 5.1 Delayed-like refinement transformers

We show that the refinement transformer approach can yield relations even coarser than $\tau_1$. Our first attempt is to generalize the direct-like winning condition of $\tau_0$ to a delayed one. Let $\tau_0^{\mathrm{de}}$ be the same as $\tau_0$ except for the different winning condition, which now is: $\forall i \geq 0 \cdot q_i \in F \Longrightarrow \exists j \geq i \cdot \hat{s}_j \in F$. Clearly, $\tau_0^{\mathrm{de}}$ inherits the same transitivity issues of $\tau_0$. Unfortunately, the approach of taking appealing fragments is not sound here, due to the weaker winning condition. See Figure 7 in the Appendix for a counterexample.

We overcome these issues by dropping $\tau_0^{\mathrm{de}}$ altogether, and directly generalize $\tau_1$ (instead of $\tau_0$) to a delayed-like notion. The *delayed refinement transformer* $\tau_1^{\mathrm{de}}$ is like $\tau_1$, except for the new winning condition: $\forall i \geq 0 \cdot \hat{q}_i \in F \Longrightarrow \exists j \geq i \cdot \hat{s}_j \in F$. Notice that $\tau_1^{\mathrm{de}}(R)$ is at least as coarse as $\tau_1(R)$, and incomparable with $\tau_0(R)$. Once $R$ is given, $\tau_1^{\mathrm{de}}(R)$ can be computed in polynomial time. See Appendix D.

**Lemma 8.** *For any $R$, $\tau_1^{de}(R)$ is transitive.*

**Theorem 6.** *If $R$ is a jumping-safe $F$-respecting preorder, then $\tau_1^{de}(R)$ is jumping-safe.*

## 6 Proxy simulations

We apply the theory of transformers from Section 5 to a specific $F$-respecting preorder, namely backward direct simulation, obtaining *proxy simulations*. Notice that proxy simulation-equivalent states need not have the same language; yet, proxy simulations are GFQ (and computable in polynomial time).

### 6.1 Direct proxy simulation

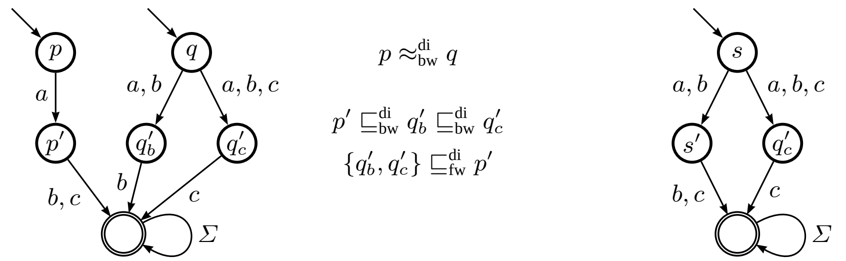
Let *direct proxy simulation*, written $\sqsubseteq_{\mathrm{xy}}^{\mathrm{di}}$, be defined as $\sqsubseteq_{\mathrm{xy}}^{\mathrm{di}} := [\tau_1(\sqsubseteq_{\mathrm{bw}}^{\mathrm{di}})]^{-1}$.

**Theorem 7.** *$\sqsubseteq_{xy}^{di}$ is a polynomial time GFQ preorder at least as coarse as $(\sqsubseteq_{bw}^{di})^{-1}$.*

*Proxies vs mediators.* Direct proxy simulation and mediated preorder [1] are in general incomparable. While proxy simulation is at least as coarse as backward direct simulation, mediated preorder is at least as coarse as *forward* direct simulation. (We have seen in Section 5 that this is somehow unavoidable, since one cannot hope to generalize simultaneously both forward and backward simulation.)

One notable difference between the two notions is that proxies are "dynamic", while mediators are "static": While Dupicator chooses the proxy only *after* Spoiler has selected her move, mediators are chosen uniformly w.r.t. Spoiler's move.

In Figure 3(a) we show a simple example where $\sqsubseteq_{\mathrm{xy}}^{\mathrm{di}}$ achieves greater reduction. Recall that mediated preorder $M$ is always a subset of $\sqsubseteq_{\mathrm{fw}}^{\mathrm{di}} \circ (\sqsubseteq_{\mathrm{bw}}^{\mathrm{di}})^{-1}$ [1]. In the example, static mediators are just the trivial ones already present in forward simulation. Thus, $\sqsubseteq_{\mathrm{fw}}^{\mathrm{di}} \circ (\sqsubseteq_{\mathrm{bw}}^{\mathrm{di}})^{-1} = \sqsubseteq_{\mathrm{fw}}^{\mathrm{di}}$ and mediated preorder $M$ collapses to forward simulation. On the other side, $p \approx_{\mathrm{xy}}^{\mathrm{di}} q$ and $p' \approx_{\mathrm{xy}}^{\mathrm{di}} q_b'$. Letting $s = [p, q]$ and $s' = [p', q_b']$, we obtain the quotient in Figure 3(b).




(a) Original automaton. (b) Quotient automaton.

**Fig. 3.** Direct proxy simulation quotients.

### 6.2 Delayed proxy simulation

Another difference between the mediated preorder approach [1] and the approach through proxies is that proxies directly enable a delayed simulation-like generalization (see Section 5.1). Again, we fix backward delayed simulation $\sqsubseteq^{\text{di}}_{\text{bw}}$ as a starting refinement, and we define *delayed proxy simulation* as $\sqsubseteq^{\text{de}}_{\text{xy}} := [\tau_1^{\text{de}}(\sqsubseteq^{\text{di}}_{\text{bw}})]^{-1}$.

**Theorem 8.** *$\sqsubseteq^{de}_{xy}$ is a polynomial time GFQ preorder.*

Notice that delayed proxy simulation is at least as coarse as direct proxy simulation. Moreover, quotients w.r.t. $\sqsubseteq^{\text{de}}_{\text{xy}}$ can be smaller than direct forward/backward/proxy and delayed simulation quotients by an arbitrary large factor. See Figure 4: Forward delayed simulation is just the identity, and no two states are direct backward or proxy simulation equivalent. But $q_i \sqsubseteq^{\text{di}}_{\text{bw}} s$ for any $0 < i \leq k-1$. This causes any two outer states $q_i, q_j$ to be $\sqsubseteq^{\text{de}}_{\text{xy}}$-equivalent. Therefore, the $\sqsubseteq^{\text{de}}_{\text{xy}}$-quotient automaton has only 2 states.

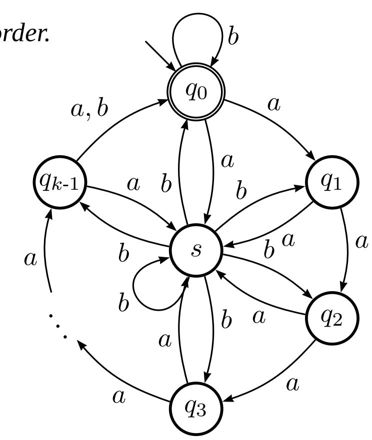


**Fig. 4.**

## 7 Conclusions and Future Work

We have proposed novel refinements for quotienting Büchi automata: fixed-word delayed simulation and direct/delayed proxy simulation. Each one has been shown to induce quotients smaller than previously known notions.

We outline a few directions for future work. First, we would like to study practical algorithms for computing fixed-word delayed simulation, and to devise efficient fragments thereof—one promising direction is to look at self-respecting fragments, which usually have lower complexity. Second, we would like to exploit the general correctness argument developed in Section 4 in order to get efficient purely backward refinements (coarser than backward direct simulation). Finally, experiments on cases of practical interest are needed for an empirical evaluation of the proposed techniques.

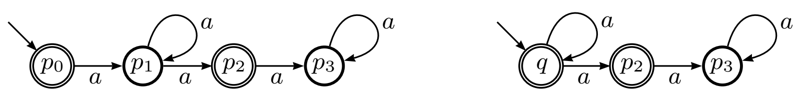
**Acknowledgment.** We thank Richard Mayr and Patrick Totzke for helpful discussions, and two anonymous reviewers for their valuable feedback.

## References


1. Abdulla, P., Chen, Y.F., Holik, L., Vojnar, T.: Mediating for Reduction. In: FSTTCS. pp. 1–12. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik (2009)
2. Aziz, A., Singhal, V., Swamy, G.M., Brayton, R.K.: Minimizing Interacting Finite State Machines. Tech. Rep. UCB/ERL M93/68, UoC, Berkeley (1993)
3. Clemente, L., Mayr, R.: Multipebble Simulations for Alternating Automata - (Extended Abstract). In: CONCUR. LNCS, vol. 6269, pp. 297–312. Springer-Verlag (2010), http://dx.doi.org/10.1007/978-3-642-15375-4_21
4. Dill, D.L., Hu, A.J., Wont-Toi, H.: Checking for Language Inclusion Using Simulation Preorders. In: CAV. LNCS, vol. 575. Springer-Verlag (1991), http://dx.doi.org/10.1007/3-540-55179-4_25
5. Etessami, K.: A Hierarchy of Polynomial-Time Computable Simulations for Automata. In: CONCUR. LNCS, vol. 2421, pp. 131–144. Springer-Verlag (2002), http://dx.doi.org/10.1007/3-540-45694-5_10
6. Etessami, K., Wilke, T., Schuller, R.A.: Fair Simulation Relations, Parity Games, and State Space Reduction for Büchi Automata. SIAM J. Comput. 34(5), 1159–1175 (2005), http://epubs.siam.org/sam-bin/dbq/article/42067
7. Fritz, C., Wilke, T.: Simulation Relations for Alternating Büchi Automata. Theor. Comput. Sci. 338(1-3), 275–314 (2005), http://dx.doi.org/10.1016/j.tcs.2005.01.016
8. Gramlich, G., Schnitger, G.: Minimizing NFA's and Regular Expressions. Journal of Computer and System Sciences 73(6), 908–923 (2007), http://www.sciencedirect.com/science/article/B6WJ0-4MMP64W-1/2/769c421c0456a3f502681ab
9. Gurumurthy, S., Bloem, R., Somenzi, F.: Fair Simulation Minimization. In: CAV. LNCS, vol. 2404, pp. 610–624. Springer-Verlag (2002), http://portal.acm.org/citation.cfm?id=647771.734423
10. Henzinger, T.A., Kupferman, O., Rajamani, S.K.: Fair Simulation. Information and Computation 173, 64–81 (2002), http://dx.doi.org/10.1006/inco.2001.3085
11. Henzinger, T.A., Rajamani, S.K.: Fair Bisimulation. In: TACAS. LNCS, vol. 1785, pp. 299–314. Springer-Verlag (2000), http://portal.acm.org/citation.cfm?id=646484.691752
12. Jiang, T., Ravikumar, B.: Minimal NFA Problems are Hard. In: Albert, J., Monien, B., Artalejo, M. (eds.) Automata, Languages and Programming, Lecture Notes in Computer Science, vol. 510, pp. 629–640. Springer Berlin / Heidelberg (1991), http://dx.doi.org/10.1007/3-540-54233-7_169
13. Juvekar, S., Piterman, N.: Minimizing Generalized Büchi Automata. In: CAV. LNCS, vol. 4414, pp. 45–58. Springer-Verlag (2006), http://dx.doi.org/10.1007/11817963_7
14. Kupferman, O., Vardi, M.: Verification of Fair Transition Systems. In: CAV, LNCS, vol. 1102, pp. 372–382. Springer-Verlag (1996), http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.29.9654
15. Kupferman, O., Vardi, M.: Weak Alternating Automata Are Not That Weak. ACM Trans. Comput. Logic 2, 408–429 (Jul 2001), http://dx.doi.org/10.1145/377978.377993
16. Lynch, N.A., Vaandrager, F.W.: Forward and Backward Simulations. Part I: Untimed Systems. Information and Computation 121(2), 214–233 (1995), http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.12.3241
17. Milner, R.: Communication and Concurrency. Prentice-Hall (1989)

18. Raimi, R.S.: Environment Modeling and Efficient State Reachability Checking. Ph.D. thesis, The University of Texas at Austin (1999)
19. Schewe, S.: Beyond Hyper-Minimisation—Minimising DBAs and DPAs is NP-Complete. In: Lodaya, K., Mahajan, M. (eds.) FSTTCS. LIPIcs, vol. 8, pp. 400–411. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany (2010), http://drops.dagstuhl.de/opus/volltexte/2010/2881
20. Somenzi, F., Bloem, R.: Efficient Büchi Automata from LTL Formulae. In: CAV, LNCS, vol. 1855, pp. 248–263. Springer-Verlag (2000), http://dx.doi.org/10.1007/10722167_21
21. Vardi, M.: Alternating Automata and Program Verification. In: Computer Science Today, LNCS, vol. 1000, pp. 471–485. Springer-Verlag (1995), http://dx.doi.org/10.1007/BFb0015261

# A Proofs and additional material for Section 3




(a) The original automaton $\mathcal{A}$.

(b) The quotient automaton $\mathcal{A}_{\approx}$, with $q = \{p_0, p_1\}$ the new quotient state.

**Fig. 5.** An example showing that delayed containment cannot be employed for quotienting. We have that $p_0$ is delayed containment equivalent to $p_1$. Notice that the automaton $\mathcal{A}$ in (a) does not accept $a^\omega$, but the quotient automaton $\mathcal{A}_{\approx}$ in (b), obtained by identifying $p_0$ and $p_1$, does.

We postpone the proof of Theorem 1 until Section E.

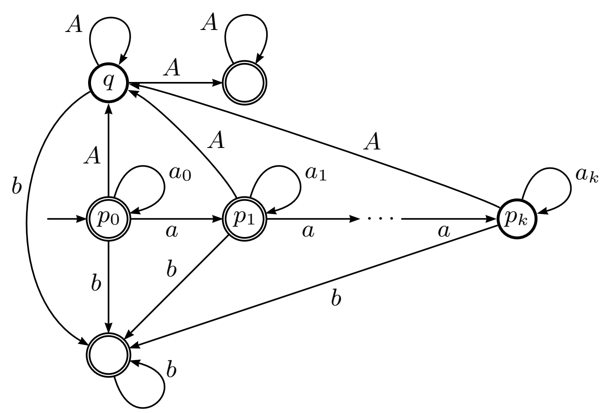



**Fig. 6.** Fixed-word delayed simulaton quotients can achieve arbitrarily high compression ratios.

### A.1 Alternating Büchi automata

Below, we give a self-contained definition of alternating Büchi automata. The syntax follows the presentation of [21], while tbe semantics adheres to [7].

For a set $A$, let $\mathcal{B}^+(A)$ be the set of positive boolean formulas over $A$, that is, $\mathcal{B}^+(A)$ is the smallest set containing $A \cup \{\mathtt{true}, \mathtt{false}\}$ and closed under the operations $\wedge$ and $\vee$. For a formula $\varphi \in \mathcal{B}^+(A)$ and a set $X \subseteq A$, we write $X \models \varphi$ iff the truth assignment assigning `true` to elements in $X$ and `false` to the elements in $A \setminus X$ satisfies $\varphi$. An alternating Büchi automaton (ABA) is a tuple $\mathcal{A} = (A, \Sigma, \delta, \alpha)$, where $A$ is a finite set of states, $\Sigma$ is a finite set of input symbols, $\delta : A \times \Sigma \mapsto \mathcal{B}^+(A)$ is the transition relation and $\alpha \subseteq A$ is the set of accepting states. Acceptance of an ABA $\mathcal{A}$ is best defined via games [7]. In this context, the two players are usually named Automaton and Pathfinder. Given an infinite word $w = a_0 a_1 \cdots \in \Sigma^\omega$ and a distinguished starting state $p_I$, the *acceptance game for $w$ from $p_I$* is a game where $P_0 = Q \times \omega$ is the set of Automaton's positions, $P_1 = Q \times 2^Q \times \omega$ is the set of Pathfinder's positions, $(p_I, 0)$ is the initial position, and transitions are determined as follows. Automaton can select a transition $(\langle p, i\rangle, \langle p, \mathbf{p}', i\rangle)$ iff $\mathbf{p}' \models \delta(p, a_i)$, and Pathfinder can select a transition $(\langle p, \mathbf{p}', i\rangle, \langle p', i+1\rangle)$ iff $p' \in \mathbf{p}'$. Finally, the winning condition consists of those paths visiting $\alpha$ infinitely often. A state $p \in A$ accepts $w \in \Sigma^\omega$ iff Automaton wins the acceptance game for $w$ from $p$. A state $p$ is universal iff it accepts every word $w \in \Sigma^\omega$.

### A.2 Proof of Theorem 3

*Preliminaries on ordinals.* Let $\omega$ be the least infinite ordinal, and let $\omega_1$ be the set of all countable ordinals. We denote abitrary ordinals by $\alpha$ or $\beta$, and limit ordinals by $\lambda$ or $\mu$. In this paper, $0$ is considered to be a limit ordinal.

*Preliminaries on trees.* Let $[n] = \{0, 1, \ldots, n-1\}$. A tree domain is a non-empty, prefix-closed subset $V$ of $[n]^*$. With $<_{\text{prf}}$ we denote the prefix order on words; if $u <_{\text{prf}} u'$, then $u'$ is called a descendant of $u$ and $u$ is an ancestor of $u'$. In particular, if $u' = uc$ for some $c \in \mathbb{N}$, then $u'$ is a child of $u$. A (labelled) $L$-tree is a pair $(V, t)$, where $V$ is a tree domain and $t : V \mapsto L$ is a mapping which assigns a label from $L$ to any node in the tree.

*The ranking construction.* Len $\mathcal{Q} = (Q, \Sigma, I, \Delta, F)$ be an automaton, and let $n$ be the cardinality of $Q$. Given an infinite word $w = a_0 a_1 \cdots \in \Sigma^\omega$, we associate to any state $q \in Q$ a tree domain $T^w_q$ and a $Q$-tree $(T^w_q, t^w_q)$, the *unravelling* of $Q$ from $q$ while reading $w$, by applying the following two rules:

- $\varepsilon \in T^w_q$ and $t^w_q(\epsilon) = q$.
- If $u$ has length $i$, $u \in T^w_q$, $t^w_q(u) = p$ and $\Delta(p, a_i) = \{p'_0, p'_1, \ldots, p'_{k-1}\}$, then, for any $j$ s.t. $0 \le j < k$, $uj \in T^w_q$ and $t^w_q(uj) = p'_j$.

It is easy to see that if two nodes at the same level have the same label, then they generate isomorphic subtrees. Therefore, we can "compress" $(T^w_q, t^w_q)$ into an infinite DAG $G^w_q = (V, E)$, where $V \subseteq Q \times \mathbb{N}$ is such that $\langle q, l\rangle \in V$ iff there exists a node in $(T^w_q, t^w_q)$ at level $l$ with label $q$, and $(\langle q, l\rangle, \langle q', l+1\rangle) \in E$ iff there exist two nodes $u$ and $u'$, labelled with $q$ and $q'$, respectively, s.t. $u'$ is a child of $u$ in $(T^w_q, t^w_q)$. We say that a vertex $\langle q, l\rangle$ is *accepting* iff $q \in F$.

For any $G \subseteq G^w_q$, we say that a vertex $\langle q, l\rangle$ is a *dead end* in $G$ iff it has no successor in $G$, and we say that it is *inert* in $G$ iff no accepting vertex can be reached from $\langle q, l\rangle$ in $G$. In particular, an inert vertex is not accepting. The *girth* of $G$ at level $l$ is the maximal number of vertices of the form $\langle q, l\rangle$ in $G$, and the *width* of $G$ is the maximal girth over infinitely many levels.

We build a nonincreasing transfinite sequence of DAGs $\{G_\alpha \mid \alpha < \omega_1\}$ as follows:

$$\begin{aligned}
G_0 &= G^w_q \\
G_{\alpha+1} &= G_\alpha \ \setminus \ \{\langle q, l\rangle \mid \langle q, l\rangle \text{ is a dead end in } G_\alpha\} \\
G_\lambda &= H_\lambda \ \setminus \ \{\langle q, l\rangle \mid \langle q, l\rangle \text{ is inert in } H_\lambda\},
\end{aligned}$$

where, for any ordinal $\alpha$, $H_\alpha = \bigcap_{\beta<\alpha} G_\beta$. Notice that $H_{\alpha+1} = G_\alpha$; and $\alpha \leq \beta$ implies $G_\beta \subseteq G_\alpha$.

Assume that there is no path in $G_q^w$ with an infinite number of accepting vertices. As a direct consequence of König's Lemma, we have that when moving from $H_\lambda$ to $G_\lambda$ an infinite path is removed from the graph. Therefore, the width of $G_\lambda$ is strictly less than the width of $H_\lambda$. Since the width of $G_q^w$ is (uniformly) bounded by $\omega$, it follows that $H_{\omega^2}$ is empty, and thus $G_{\omega^2}$ is empty as well. Therefore, each vertex is either a dead end in $G_\alpha$ or inert in $H_\lambda$. In the former case $\langle q, l\rangle$ is in $G_\alpha$ but not in $G_{\alpha+1}$, whereas in the latter case $\langle q, l\rangle$ is in $H_\lambda$ but not in $G_\lambda$. Accordingly, we associate an ordinal *rank* to every vertex $\langle q, l\rangle$ in $G_q^w$:

$$\mathsf{rank}_q^w(q,l) = \sup_{\alpha<\omega^2} \{\alpha \mid \langle q, l\rangle \in H_\alpha\} \ . \qquad (\mathsf{Rank})$$

Therefore, under the assumption that $G_q^w$ does not contain any fair path, no vertex receives rank $\omega^2$. On the other side, if $G_q^w$ contained a fair path, then there exists an infinite path of non-inert vertices starting at $\langle q, 0\rangle$: In this case, the ranking construction "does not terminate" and stabilizes (at most) at a nonempty $G_{\omega^2} = G_\alpha \neq \emptyset$ for all $\alpha \geq \omega^2$. Thus, vertices in $G_{\omega^2}$ would receive rank $\omega^2$ according to $(\mathsf{Rank})$. Since no conflict can arise, we drop any assumption about fair paths thereafter, and we uniformly apply $(\mathsf{Rank})$ in either case.

*Remark 1.* It is clear from $(\mathsf{Rank})$ that no ordinal larger than $\omega^2$ is actually used in our construction. In fact, we could have given an equivalent presentation in terms of pairs of natural numbers ordered lexicographically. However, we have chosen to use ordinals $\leq \omega^2$ for technical convenience.

*Remark 2.* A vertex $\langle q, l\rangle$ is in $H_\alpha$ iff it has rank $\geq \alpha$, and it is in not in $G_\alpha$ iff it has rank $\leq \alpha$. Therefore, $\mathsf{rank}_q^w(q,l) = \alpha \iff \langle q, l\rangle \in H_\alpha \setminus G_\alpha$.

**Lemma 9.** *If a vertex $\langle q, l\rangle$ is accepting, then it has rank $\alpha + 1$. Furthermore, if it has rank $\lambda + 1$, then it is accepting.*

*Proof.* The first part follows from the fact that an accepting vertex $\langle q, l\rangle$ is not inert: Therefore, $\langle q, l\rangle$ is a dead end in $G_\alpha$, so $\langle q, l\rangle \notin G_{\alpha+1}$ and $\mathsf{rank}_q^w(q,l) = \alpha + 1$.

For the second part, assume $\mathsf{rank}_q^w(q,l) = \lambda+1$, i.e., $\langle q, l\rangle \in G_\lambda \setminus G_{\lambda+1}$. Therefore, $\langle q, l\rangle$ is a dead end in $G_\lambda$. Since $G_\lambda \subseteq H_\lambda$, $\langle q, l\rangle$ is in $H_\lambda$ as well. But $H_\lambda$ has no dead ends, therefore $\langle q, l\rangle$ has at least one successor $\langle q', l+1\rangle$ in $H_\lambda$. But $\langle q, l\rangle$ is a dead end in $G_\lambda$, therefore any such successor $\langle q', l+1\rangle$ is not in $G_\lambda$. Therefore, $\langle q', l+1\rangle$ is inert in $H_\lambda$.

By contradiction, assume $\langle q, l\rangle$ that is not accepting. Since it has only inert successors $\langle q', l+1\rangle$ in $H_\lambda$, it is itself inert in $H_\lambda$. But $\langle q, l\rangle \in G_\lambda$, so $\langle q, l\rangle$ is *not* inert in $H_\lambda$. This is a contradiction, therefore $\langle q, l\rangle$ is accepting. ☐

We say that a vertex $\langle q', l+1\rangle$ is a *maximal successor* of $\langle q, l\rangle$ if its rank is maximal amongst all successors of $\langle q, l\rangle$, and a sequence $\langle q_0, l\rangle\langle q_1, l+1\rangle \cdots \langle q_h, l+h\rangle$ is a *maximal path* if, for any $0 \geq k < h$, $\langle q_{k+1}, l+k\rangle$ is a maximal successor of $\langle q_k, l+k-1\rangle$.

We define a predecessor and a floor operation on ordinals. For an ordinal $\alpha$, its *predecessor* $\alpha - 1$ is either $\alpha$ itself if $\alpha$ is a limit ordinal, or $\beta$ if $\alpha = \beta + 1$ for some $\beta$; its *floor* $\lfloor \alpha \rfloor := \sup_{\lambda < \alpha} \lambda$ is the largest limit ordinal strictly smaller than $\alpha$. Notice that, for $0 < \alpha < \omega^\omega$, $\lfloor \alpha \rfloor < \alpha$.

**Lemma 10.** *Let vertex $\langle q, l \rangle$ have rank $\alpha$. Then, a) every successor $\langle q', l+1 \rangle$ has rank at most $\alpha - 1$, and b) there exists a maximal successor attaining rank $\alpha - 1$. As a direct consequence, c) every node $\langle q', l' \rangle$ reachable from $\langle q, l \rangle$ has a smaller rank $\alpha' \leq \alpha$.*

*Proof.* We split the proof in two cases, depending on whether $\alpha$ is a successor or limit ordinal. Let $\alpha$ be a successor ordinal $\beta + 1$. Then, $\langle q, l \rangle$ is a dead end in $G_\beta$, and thus it has no successor in $G_\beta$. Therefore, each successor $\langle q', l+1 \rangle$ has rank $\leq \beta$. Moreover, we show that at least one successor has rank exactly equal to $\beta$. To this end, let $\beta^* \leq \beta$ be the maximum rank amongst $\langle q, l \rangle$'s successors. Notice that no successor $\langle q', l+1 \rangle$ is in $G_{\beta^*}$. As $G_\beta \subseteq G_{\beta^*}$, it follows that $\langle q, l \rangle$ is a dead end in $G_{\beta^*}$. Therefore, $\langle q, l \rangle$ is not in $G_{\beta^*+1}$, which implies it has rank at most $\beta^* + 1 \leq \beta + 1$. But $\mathsf{rank}_q^w(q, l) = \beta + 1$ by assumption. Therefore, $\beta^* = \beta$, as required.

Otherwise, let $\alpha$ be a limit ordinal $\lambda$. Thus, $\langle q, l \rangle$ is inert in $H_\lambda$. Let $\langle q', l+1 \rangle$ be a successor of $\langle q, l \rangle$. If $\langle q', l+1 \rangle$ is not in $H_\lambda$, then, since $G_\lambda \subseteq H_\lambda$, $\langle q', l+1 \rangle$ is not in $G_\lambda$ either. Thus, $\langle q', l+1 \rangle$ has rank $\leq \lambda$ in this case. Otherwise, let $\langle q', l+1 \rangle$ be in $H_\lambda$. Since $\langle q, l \rangle$ is inert in $H_\lambda$, it follows that $\langle q', l+1 \rangle$ is inert in $H_\lambda$ as well. Therefore, $\langle q', l+1 \rangle$ gets rank exactly equal to $\lambda$ in this case. Finally, since $H_\lambda$ does not contain dead ends, there exists at least one such inert successor in $H_\lambda$. $\square$

**Lemma 11.** *If a vertex $\langle q_0, l \rangle$ has a successor ordinal rank $\alpha + 1$, then there exists a maximal path $\langle q_0, l \rangle \langle q_1, l+1 \rangle \cdots \langle q_h, l+h \rangle$ ending in $\langle q_h, l+h \rangle$ of rank $\lambda + 1$ with $\lfloor \alpha + 1 \rfloor \leq \lambda$.*

*Proof.* We proceed by ordinal induction. If $\alpha$ is a limit ordinal $\lambda$, the claim holds immediately: Take $h = 0$; clearly, $\lambda = \lfloor \lambda + 1 \rfloor$.

Otherwise, let $\alpha$ be a successor ordinal $\beta + 1$. That is, vertex $\langle q_0, l \rangle$ has rank $\alpha + 1 = (\beta + 1) + 1$. By Lemma 25 b), $\langle q_0, l \rangle$ has a maximal successor $\langle q_1, l+1 \rangle$ of rank $\beta + 1 = \alpha$. By induction, there exists a maximal path $\langle q_1, l+1 \rangle \cdots \langle q_h, l+h \rangle$ with $h > 0$, ending in $\langle q_h, l+h \rangle$ of rank $\lambda + 1$ with $\lfloor \beta + 1 \rfloor \leq \lambda$. But $\beta = \alpha + 1$, thus $\lfloor \beta + 1 \rfloor = \lfloor \alpha + 1 \rfloor \leq \lambda$. $\square$

**Lemma 12.** *If a vertex $\langle q_0, l \rangle$ has a nonzero limit ordinal rank $\lambda$, then there exists a path $\langle q_0, l \rangle \langle q_1, l+1 \rangle \cdots \langle q_h, l+h \rangle$ with $h \geq 1$ ending in $\langle q_h, l+h \rangle$ of rank $\alpha + 1$ with $\lfloor \lambda \rfloor \leq \alpha$.*

*Proof.* Let $\langle q_0, l \rangle$ have rank $\lambda > 0$. By contradiction, assume $\langle q_0, l \rangle$ has no descendant $\langle q', l' \rangle$ of rank $\alpha + 1$ with $\lfloor \lambda \rfloor \leq \alpha$. That is, all descendants $\langle q', l' \rangle$ of successor ordinal rank $\alpha + 1$ have $\alpha < \lfloor \lambda \rfloor$, which is the same as $\alpha + 1 < \lfloor \lambda \rfloor$. By definition, $\langle q_0, l \rangle$ is inert in $H_\lambda \subseteq H_{\lfloor \lambda \rfloor}$. We show that $\langle q_0, l \rangle$ is inert in $H_{\lfloor \lambda \rfloor}$ as well. This is a contradiction, since $\lambda$ is nonzero, therefore $\langle q_0, l \rangle$ would get rank $\lfloor \lambda \rfloor < \lambda$.

To this end, we show that any vertex reachable from $\langle q_0, l \rangle$ in $H_{\lfloor \lambda \rfloor}$ is non-accepting. For such a vertex $\langle q', l' \rangle$ to be accepting, by Lemma 9, it is necessary to have successor rank $\alpha + 1 < \lfloor \lambda \rfloor$. Clearly, $\langle q', l' \rangle \notin H_{\lfloor \lambda \rfloor}$. Therefore, $\langle q_0, l \rangle$ is inert in $H_{\lfloor \lambda \rfloor}$. $\square$

**Lemma 13.** *Let $w \in \Sigma^\omega$. If $\mathsf{rank}^w_{q_0}(q_0, 0) \leq \mathsf{rank}^w_{s_0}(s_0, 0)$, then $q_0 \sqsubseteq^{de}_w s_0$.*

*Proof.* Assume $\mathsf{rank}^w_{q_0}(q_0, 0) \leq \mathsf{rank}^w_{s_0}(s_0, 0)$. We show that Duplicator has a winning strategy in $G^{\text{de}}_w(q_0, s_0)$. For any round $i$, let $\langle q_i, s_i\rangle$ be the current configuration of the simulation game, and let the rank of Spoiler and Duplicator at round $i$ be $\mathsf{rank}^w_{q_0}(q_i, i)$ and $\mathsf{rank}^w_{s_0}(s_i, i)$, respectively. Intuitively, Duplicator wins by ensuring both a *safety* and a *liveness* condition. The safety condition requires Duplicator to always preserve the ordering between ranks. I.e., at round $i$, $\mathsf{rank}^w_{q_0}(q_i, i) \leq \mathsf{rank}^w_{s_0}(s_i, i)$. The liveness condition enforces Duplicator to (eventually) visit an accepting state if Spoiler does so.

Duplicator plays in two modes, *normal mode* and *obligation mode*. In normal mode Duplicator only enforces the safety condition, while in obligation mode Duplicator needs to satisfy the liveness condition, while still preserving the safety condition.

In normal mode, we asssume that Duplicator's rank is a limit ordinal, and, by Lemma 25, Duplicator can preserve the rank by always selecting maximal successors. We say that Duplicator *plays maximally* during normal mode. The game stays in normal mode as long as Spoiler is not accepting. Whenever $q_i \in F$ at round $i$, then Duplicator switches to obligation mode. Suppose that the current rank of Duplicator at round $i$ is a limit ordinal $\lambda$. Since $q_i \in F$, by Lemma 9 Spoiler's rank is a successor ordinal $\alpha + 1 < \lambda$. W.l.o.g. we assume that Spoiler plays maximally during obligation mode. By Lemma 11, there exists a maximal path $\langle q_i, i\rangle\langle q_{i+1}, i + 1\rangle \cdots \langle q_j, j\rangle$ s.t. Spoiler's rank at round $j \geq i$ is $\lambda' + 1$. A further move by Spoiler extends the previous path to $\langle q_{j+1}, j + 1\rangle$. By Lemma 25 b), Spoiler's rank at round $j + 1$ is now $\lambda'$, and by part c) of the same lemma, $\lambda' \leq \alpha + 1$. By part b), Duplicator can play a maximal path $\langle s_i, i\rangle\langle s_{i+1}, i + 1\rangle \cdots \langle s_{j+1}, j + 1\rangle$ s.t. Duplicator's rank at round $j + 1$ is $\lambda$. Thus, $\lambda' < \lambda$, which implies $\lambda' \leq \lfloor\lambda\rfloor$. So, let $\langle q_{j+1}, s_{j+1}\rangle$ be the configuration at round $j + 1$. By Lemma 12, Duplicator can play a path $\langle s_{j+1}, j + 1\rangle\langle s_{j+2}, j + 2\rangle \cdots \langle s_k, k\rangle$ with $k > j + 1$ and s.t. Duplicator's rank at round $k$ is $\alpha' + 1$ with $\lfloor\lambda\rfloor \leq \alpha'$. Therefore, $\lambda' \leq \alpha'$. By Lemma 11, Duplicator can extend the previous path with a maximal path $\langle s_k, k\rangle\langle s_{k+1}, k + 1\rangle \cdots \langle s_h, h\rangle$ s.t. Duplicator's rank at round $h > k$ is $\lambda'' + 1$ with $\lfloor\alpha' + 1\rfloor \leq \lambda''$. By Lemma 9, $s_h \in F$, thus Duplicator has satisfied the pending obligation. At round $h + 1$, Duplicator's rank is $\lambda''$ by Lemma 25 b), and the game can switch to normal mode. Notice that $\lambda' \leq \alpha' < \alpha' + 1$ implies $\lambda' \leq \lfloor\alpha + 1\rfloor$. Therefore, $\lambda' \leq \lambda''$ and the safety condition is satisfied. □

**Lemma 14.** *Let $w \in \Sigma^\omega$ and $k \geq 1$. If $q_0 \sqsubseteq^{k\text{-}de}_w s_0$, then $\mathsf{rank}^w_{q_0}(q_0, 0) \leq \mathsf{rank}^w_{s_0}(s_0, 0)$*

*Proof.* We prove the contrapositive. Assume $\mathsf{rank}^w_{q_0}(q_0, 0) \not\leq \mathsf{rank}^w_{s_0}(s_0, 0)$. Since ordinals are linearly ordered, this means $\mathsf{rank}^w_{q_0}(q_0, 0) > \mathsf{rank}^w_{s_0}(s_0, 0)$. We have to show $q_0 \not\sqsubseteq^{k\text{-de}}_w s_0$, for arbitrary $k \geq 1$. Take $n$ to be the size of the automaton. We actually prove that Duplicator does not win even with $n$ pebbles, i.e., $q_0 \not\sqsubseteq^{n\text{-de}}_w s_0$.

For any round $i$, let $\langle q_i, \mathbf{s}_i\rangle$ be the current configuration of the simulation game $G^{n\text{-de}}_w(q_0, s_0)$. (For simplicity, we omit the third component.) Notice that $\mathbf{s}_i$ identifies a *subset* of vertices at level $i$ in $G^w_{s_0}$: $\mathbf{s}_i \subseteq \{s \mid \langle s, i\rangle \in G^w_{s_0}\}$. We extend the notion of rank to sets of vertices by taking the maximal rank. That is, the rank of Duplicator at round $i$ is $\sup_{s\in\mathbf{s}_i} \mathsf{rank}^w_{s_0}(s, i)$. As before, Spoiler's rank is just $\mathsf{rank}^w_{q_0}(q_i, i)$.

We assume that, at round 0, every pebble has limit rank. If not, Spoiler can enforce such a situation by waiting a suitable number of rounds. (I.e., by playing maximally according to Lemma 25.) So, let's Spoiler have limit rank $\lambda$ and Duplicator have limit rank $\mu$, with $\lambda > \mu$. We assume that Duplicator always plays maximally, unless she is forced to act differently. By Lemma 12, Spoiler can play a path $\langle q_0, 0\rangle\langle q_1, 1\rangle \cdots \langle q_i, i\rangle$ with $i > 0$, s.t. her rank at round $i$ is $\alpha+1$ and $\alpha \geq \lfloor\lambda\rfloor$. From $\lambda > \mu$ we have $\lfloor\lambda\rfloor \geq \mu$, which implies $\alpha \geq \mu$. By Lemma 11, Spoiler can extend the previous path with a maximal path $\langle q_i, i\rangle\langle q_{i+1}, i+1\rangle \cdots \langle q_j, j\rangle$ with $j > i$, s.t. her rank at round $j$ is $\lambda' + 1$ and $\lambda' \geq \lfloor\alpha + 1\rfloor$. By Lemma 9, $q_j \in F$. From $\alpha + 1 > \alpha \geq \mu$ we have $\lfloor\alpha + 1\rfloor \geq \mu$, which implies $\lambda' \geq \mu$. By performing a further maximal step, Spoiler reaches state $\langle q_{j+1}, j+1\rangle$, thus attaining rank $\lambda'$. From now on, Spoiler plays maximally.

Since Duplicator was playing maximally, in the meanwhile she replied to Spoiler with a sequence $\langle \mathbf{s}_0, 0\rangle\langle \mathbf{s}_1, 1\rangle \cdots \langle \mathbf{s}_{j+1}, j+1\rangle$ s.t. she has rank $\mu$ at round $j+1$.

Now, let $\langle q_{j+1}, \mathbf{s}_{j+1}\rangle$ be the current configuration, and remember that Duplicator has a pending obligation. That is, Duplicator has to ensure that at some future round $k$ all pebbles are good since round $j+1$. Let $\mathbf{s}_k$ be the position of pebbles at round $k$. This implies that every state in $\mathbf{s}_k$ has an accepting predecessor since round $j+1$. By Lemma 9, accepting pebbles receive successor ranks, and, since ranks are nonincreasing along paths in $G^w_{s_0}$ (by Lemma 25), it follows that every pebble in $\mathbf{s}_k$ has rank $< \mu$. That is, Duplicator's rank at round $k$ is $< \mu$. Since Duplicator has now satisfied the pending obligation, she will again play maximally, from round $k$ on. By Lemma 25, all pebbles eventually stabilize to a limit rank. Since there is a finite number of pebbles, it follows that at some round $h \geq k$ Duplicator's rank is $\mu' < \mu$. Let $\mathbf{s}_h$ be the position of Duplicator's pebbles at round $h$.

In the meanwhile Spoiler replied with a maximal path $\langle q_{j+1}, j+1\rangle \cdots \langle q_h, h\rangle$, preserving rank $\lambda' \geq \mu > \mu'$ until round $h$. Therefore, $\lambda' > \mu'$ and the situation at round $h$ is identical to the initial situation at round 0.

Since ordinals are well-founded, Spoiler can iterate the whole procedure and after a finite number of repetitions Duplicator hits the trap rank $\omega$. At that point, Spoiler would have a limit rank $\lambda'' > \omega$, so she will just force one more obligation, which would remain unmet (vertices of rank $\omega$ have no accepting successor). Thus, Spoiler wins. ☐

**Theorem 3.** *For any NBA $\mathcal{Q}$, $k \geq 1$ and states $q, s \in Q$, $q \sqsubseteq^{k\text{-de}}_{fx} s$ iff $q \sqsubseteq^{de}_{fx} s$.*

*Proof.* By combining the previous two lemmas, we get

$$q \sqsubseteq^{\text{de}}_{\text{fx}} s \implies q \sqsubseteq^{k\text{-de}}_{\text{fx}} s \implies \left(\mathsf{rank}^w_q(q, 0) \leq \mathsf{rank}^w_s(s, 0)\right) \implies q \sqsubseteq^{\text{de}}_{\text{fx}} s\ ,$$

where the first implication holds by the definition of $\sqsubseteq^{k\text{-de}}_{\text{fx}}$, and the last two by Lemmas 14 and 13, respectively.

## B Proofs for Section 4

**Lemma 3.** *Let $w \in \Sigma^\omega$ and $\pi_0, \pi_1, \ldots$ as in Definition 1. If $\Pi = \pi_0, \pi_1, \ldots$ is coherent, then any infinite subsequence $\Pi' = \pi_{f(0)}, \pi_{f(1)}, \ldots$ thereof is coherent.*

*Proof.* Let $\Pi := \pi_0, \pi_1, \ldots$ be an infinite coherent sequence, and let $\Pi' := \pi_{f(0)}, \pi_{f(1)}, \ldots$ be any infinite subsequence thereof, for some $f : \mathbb{N} \mapsto \mathbb{N}$ with $f(0) < f(1) < \cdots$. We have to show

$$\forall i' \cdot \exists j' \cdot \exists h' \cdot \forall k' \geq h' \cdot j' < |\pi_{f(k')}| \wedge \mathsf{cnt\text{-}final}(\pi_{f(k')}, j') \geq i' \ .$$

Let $i' \in \mathbb{N}$. By taking $i := i'$, by the coherence of $\Pi$, there exists $j, h$ s.t

$$(*)\ \forall k \geq h \cdot j < |\pi_k| \wedge \mathsf{cnt\text{-}final}(\pi_k, j) \geq i'.$$

Let $h'$ be the minimal $m$ s.t. $f(m) \geq h$. For any $k' \geq h'$, we have $f(k') \geq f(h') \geq h$. Thus, by letting $k := f(k')$ in $(*)$, we obtain $j < |\pi_{f(k')}| \wedge \mathsf{cnt\text{-}final}(\pi_{f(k')}, j) \geq i'$. Take $j' := j$. Since $k' \geq h'$ was arbitrary, we have proved that $\Pi'$ is coherent. □

**Lemma 4.** *For $w \in \Sigma^\omega$, let $\Pi = \pi_0, \pi_1, \ldots$ be a coherent sequence of paths over (prefixes of) $w$. Then, there exists a fair path $\rho$ over $w$. Moreover, if all $\pi_i$'s are initial, then $\rho$ is initial.*

*Proof.* Let $\Pi := \pi_0, \pi_1, \ldots$ be a coherent sequence. We prove by induction the following claim: For $l \in \mathbb{N}$, $R(l)$ holds iff there exists a finite sequence of finite paths $\rho_0 <_{\text{prf}} \rho_1 <_{\text{prf}} \ldots <_{\text{prf}} \rho_l$, with $\rho_l$ of length $m_l$, and an infinite subsequence $\Pi_l := \pi_{f_l(0)}, \pi_{f_l(1)}, \ldots$ of $\Pi$ with $f_l(0) < f_l(1) < \cdots$, such that

$$(a)\ \mathsf{cnt\text{-}final}(\rho_l, m_l) \geq l \qquad (b)\ \Pi_l \text{ is coherent} \qquad (c)\ \forall k \cdot \rho_l \leq_{\text{prf}} \pi_{f_l(k)} \ . \tag{3}$$

For the base case $l = 0$, take $\rho_0 := \varepsilon$ of length $m_0 := 0$, and $f_0(i) = i$ for any $i$. Then, $\Pi_0 = \Pi$ and $R(0)$ holds.

For the inductive step, assume $R(l-1)$ holds. That is, there exist $\rho_0 <_{\text{prf}} \rho_1 <_{\text{prf}} \ldots <_{\text{prf}} \rho_{l-1}$, with $\rho_{l-1}$ of length $m_{l-1}$, and $\Pi_{l-1} = \pi_{f_{l-1}(0)}, \pi_{f_{l-1}(1)}, \ldots$ with $\rho_{l-1} \leq_{\text{prf}} \pi_{f_{l-1}(k)}$ for any $k$. Since $\Pi_{l-1}$ is coherent, by taking $i := l$, there exist $j$ and $h$ s.t., for any $\pi$ in the sequence $\pi_{f_{l-1}(h)}, \pi_{f_{l-1}(h+1)}, \ldots, \pi$ has length at least $j$ and $\mathsf{cnt\text{-}final}(\pi, j) \geq l$. Since the various $\pi$'s are branches in a finitely-branching tree, it follows that at any fixed depth $d$ there are only finitely many different branches of length $d$. Therefore, there exists a least one such finite branch which is shared by infinitely many $\pi$'s. For $d = j$, we get that there exists a finite path $\rho'$ of length $j$ s.t. $\mathsf{cnt\text{-}final}(\rho', j) \geq l$ and $\rho' \leq_{\text{prf}} \pi$ for infinitely many such $\pi$'s. Let $\Pi_l := \pi_{g(f_{l-1}(h))}, \pi_{g(f_{l-1}(h+1))}, \ldots$ be this infinite subsequence. We assume w.l.o.g. that $m_{l-1} < j$, and, consequently, $\rho_l <_{\text{prf}} \rho'$. Take $f_l := g \circ f_{l-1}$, $\rho_l := \rho'$ and $m_l := j$. Then, (a) and (c) are satisfied by construction, while (b) follows by Lemma 3. This proves $R(l)$, concluding the inductive step.

Therefore, one can build the infinite sequence of finite paths $\varepsilon = \rho_0 <_{\text{prf}} \rho_1 <_{\text{prf}} \cdots$ such that, for any $l$, $\rho_l$ visits at least $l$ final states. Take $\rho$ to be the limit of the $\rho_l$'s. Finally, since $\rho_1 <_{\text{prf}} \pi_{f_1(0)}$ by property $(c)$, it follows that if all $\pi_i$'s are initial, then so is $\pi_{f_1(0)}$, and thus $\rho$. □

**Theorem 4.** *Let $R$ be a jumping-safe preorder. Then, $R$ is good for quotienting.*

*Proof.* Assume $R$ is jumping-safe and let $\approx_R$ be the equivalence induced by $R$. We have to show $\mathcal{L}^\omega(\mathcal{Q}) = \mathcal{L}^\omega(\mathcal{Q}_{\approx_R})$. The direction $\mathcal{L}^\omega(\mathcal{Q}) \subseteq \mathcal{L}^\omega(\mathcal{Q}_{\approx_R})$ holds by Lemma 1.

For the other direction, assume $w \in \mathcal{L}^\omega(\mathcal{Q}_{\approx_R})$, with $w = a_0 a_1 \cdots \in \Sigma^\omega$. Let $\pi_{\approx_R} = [q_0] \xrightarrow{a_0} [q_1] \xrightarrow{a_1} [q_2] \cdots$ be an accepting run over $w$ in $\mathcal{Q}_{\approx_R}$. By the definition of quotient, for any $i$, there exist states $q_i, q_i^F, \hat{q}_i \in Q$ s.t. $q_i \mathrel{R} q_i^F \mathrel{R} \hat{q}_i$ and $\hat{q}_i \xrightarrow{a_i} q_{i+1}$. That is, $\pi_{\approx_R}$ induces a jumping path $\pi$ as in Equation 2. Moreover, $q_i^F$ can be taken in $F$ if $[q_i]$ is accepting. Since $[q_0]$ is initial, we assume w.l.o.g. that $q_0 \in I$. Since $R$ is jumping-safe and $\pi$ is both initial and fair, there exists a coherent sequence of initial paths $\pi_0, \pi_1, \ldots$ over prefixes of $w$. By Lemma 4, there exist an (non-jumping) accepting run over $w$ in $Q$. Therefore, $w \in \mathcal{L}^\omega(\mathcal{Q})$. □

## C Proofs for Section 5

**Lemma 5.** *For a preorder $R$, $R \subseteq R \circ \tau_0(R) \subseteq \tau_0(R)$.*

*Proof.* Directly from Lemmas 15 and 16 below.

**Lemma 15.** *For any reflexive $R$, $R \subseteq \tau_0(R)$.*

*Proof.* Let $T := \tau_0(R)$, and assume $s \mathrel{R} q$. We have to show $s \mathrel{T} q$. Let's Spoiler select $a$ and $q'$ s.t. $q \xrightarrow{a} q'$. Since $s \mathrel{R} q$ by assumption, Duplicator can directly take $\hat{s} := q$. Trivially $q \in F \implies \hat{s} \in F$, as required by the winning condition. □

**Lemma 16.** *For any transitive $R$, $R \circ \tau_0(R) \subseteq \tau_0(R)$.*

*Proof.* Let $T := \tau_0(R)$, and assume $\bar{s} \mathrel{R} s \mathrel{T} q$. We have to show $\bar{s} \mathrel{T} q$. Let's Spoiler select $a$ and $q'$ s.t. $q \xrightarrow{a} q'$. Since $s \mathrel{T} q$ by assumption, Duplicator can select $\hat{s}$ s.t. $s \mathrel{R} \hat{s}$ and $\hat{s} \xrightarrow{a} s'$, for some $s'$. Then, by transitivity, $\bar{s} \mathrel{R} \hat{s}$. As $q \in F \implies \hat{s} \in F$ (by $s \mathrel{T} q$), we conclude that Duplicator wins from $\bar{s}$ as well, thus $\bar{s} \mathrel{T} q$. □

**Theorem 5.** *Let $R$ a $F$-respecting preorder, and let $T \subseteq \tau_0(R)$ be an appealing, improving fragment of $\tau_0(R)$. If $R$ is jumping-safe, then $T$ is jumping-safe.*

*Proof.* Assume that $R$ is jumping-safe and $F$-respecting, and let $T$ be an appealing, improving fragment of $\tau_0(R)$. That is, $T$ is a self-respecting and transitive fragment of $\tau_0(R)$, with $R \subseteq T$. We have to show that $T$ is jumping-safe. To this end, let $w = a_0 a_1 \cdots \in \Sigma^\omega$, and let the following be an initial $T$-jumping path

$$\pi = q_0 \mathrel{T} q_0^F \mathrel{T} \hat{q}_0 \xrightarrow{a_0} q_1 \mathrel{T} q_1^F \mathrel{T} \hat{q}_1 \xrightarrow{a_1} q_2 \cdots, \quad q_0 \in I .$$

First, we show by induction the following claim: For any $i \geq 0$, there exists a finite initial path

$$\rho_i = r_0 \mathrel{R} \hat{r}_0 \xrightarrow{a_0} r_1 \mathrel{R} \hat{r}_1 \xrightarrow{a_1} \cdots r_i, \quad r_0 \in I ,$$

s.t. $r_i \mathrel{T} q_i$, and, for any $0 \leq k < i$, $q_k^F \in F \implies \hat{r}_k \in F$.

For $i = 0$, just take $r_0 := q_0$. For $i \geq 0$, assume $\rho_i = r_0 \mathrel{R} \hat{r}_0 \xrightarrow{a_0} r_1 \cdots r_i$ has already been built. Since $q_i^F \mathrel{T} \hat{q}_i \xrightarrow{a_i} q_{i+1}$, by the definition of $T$ there exists $\hat{q}_i^F \xrightarrow{a_i} q'$ for some $\hat{q}_i^F$ and $q'$ with $q_i^F \mathrel{R} \hat{q}_i^F$ and $q' \mathrel{T} q_{i+1}$. But $q_i \mathrel{T} q_i^F$ and, by induction hypothesis, $r_i \mathrel{T} q_i$. Since $T$ is transitive, we get $r_i \mathrel{T} q_i^F$, so there exists

$\hat{r}_i \xrightarrow{a_i} r_{i+1}$ with $r_i \; R \; \hat{r}_i$ and $r_{i+1} \; T \; q'$. Again by transitivity, we get $r_{i+1} \; T \; q_{i+1}$. Moreover, if $q_i^F \in F$, then since $R$ respects final states, we have $\hat{q}_i^F \in F$, and, by the definition of $T$, we finally derive $\hat{r}_i \in F$. Thus, we have just built $\rho_{i+1} = r_0 \; R \; \hat{r}_0 \xrightarrow{a_0} r_1 \cdots r_i \; R \; \hat{r}_i \xrightarrow{a_i} r_{i+1}$. This concludes the inductive step, and the claim is proved.

From the claim above, let $\rho$ be the infinite initial $R$-jumping sequence resulting by taking limit of the $\rho_i$'s. Since $R$ is jumping-safe, there exists an infinite sequence of initial finite paths $\pi_0, \pi_1, \ldots$ s.t. $\mathrm{last}(\pi_i) \; R \; r_i$. By assumption $R \subseteq T$, so $\mathrm{last}(\pi_i) \; T \; r_i$ holds as well. By $r_i \; T \; q_i$ and transitivity, we obtain $\mathrm{last}(\pi_i) \; T \; q_i$. Therefore, the same sequence $\pi_0, \pi_1, \ldots$ can be taken as a witness for $T$ being jumping-safe.

Finally, assume that $\pi$ is fair, i.e., $q_i^F \in F$ for infinitely many $i$'s. By the claim above, $\hat{r}_i \in F$ for infinitely many $i$'s, therefore $\rho$ is fair as well. Since $R$ is jumping-safe (by taking $r_i^F := \hat{r}_i$, $R$ being reflexive), we finally infer that $\pi_0, \pi_1, \ldots$ is coherent, which concludes the proof. $\square$

**Lemma 6.** *For any reflexive $R$, let $T \subseteq \tau_0(R)$ be any appealing fragment of $\tau_0(R)$. Then, $\tau_0(T) \subseteq \tau_0(R)$. That is, at the second iteration $\tau_0$ does not introduce any new fragment which could not be found before.*

*Proof.* Let $R$ be reflexive. Let $T$ be an appealing (= transitive and self-respecting) fragment of $V_0 := \tau_0(R)$, and let $V_1 := \tau_0(T)$. We have to show $V_1 \subseteq V_0$. To this end, let $s \; V_1 \; q$ and let Spoiler choose a transition $q \xrightarrow{a} q'$. By the definition of $V_1$, there exist $s \; T \; \bar{s}$ and $\bar{s}'$ with $\bar{s} \xrightarrow{a} \bar{s}'$ and $\bar{s}' \; V_1 \; q'$. By the definition of $T$, there exist $s \; R \; \hat{s}$ and $s'$ with $\hat{s} \xrightarrow{a} s'$ and $s' \; T \; \bar{s}'$ (since $T$ is self-respecting). $T$ being transitive, from $s' \; V_1 \; \bar{s}' \; T \; q'$ and from Lemma 16, we get $s' \; V_1 \; q'$. Thus, we let Duplicator choose $\hat{s}$ and $s'$ above, as required by the definition of $V_0$. Duplicator is winning as $q \in F$ implies $\bar{s} \in F$, and the latter implies $\hat{s} \in F$, the first implication holding by the definition of $V_1$, and the second by $T$. Therefore, $s \; V_0 \; q$. $\square$

**Lemma 17.** *For any relation $R$, $\tau_1(R)$ is transitive.*

*Proof.* Let $T := \tau_1(R)$, and let $s \; T \; r \; T \; p$. We have to show $s \; T \; p$. Let Spoiler choose $a$ and $\hat{p}$ and $p'$ s.t. $p \; R \; \hat{p}$ and $\hat{p} \xrightarrow{a} p'$. We have to show 1) that Duplicator can choose $\hat{s}$ and $s'$ s.t. $s \; R \; \hat{s}$ and $\hat{s} \xrightarrow{a} s'$, and 2) $\hat{p} \in F \implies \hat{s} \in F$. For 1), from $r \; T \; p$ it follows that there exist $\hat{r}$ and $r'$ s.t. $r \; R \; \hat{r}$ and $\hat{r} \xrightarrow{a} r'$. Then, from $s \; T \; r$ one can directly find the required $\hat{s}$ and $s'$. For 2), assume $\hat{p} \in F$. From $r \; T \; p$ it follows that the $\hat{r}$ found above is in $F$ as well. Finally, $\hat{s} \in F$ follows from $s \; T \; r$ in a similar way. $\square$

**Lemma 18.** *For any transitive $R$, $R \subseteq \tau_1(R)$.*

*Proof.* Let $T := \tau_1(R)$, and assume $s \; R \; q$. We have to show $s \; T \; q$. Let's Spoiler select $a$ and $\hat{q}$ and $q'$ s.t. $q \; R \; \hat{q}$ and $\hat{q} \xrightarrow{a} q'$. Since $s \; R \; q$ by assumption, and from $R$ being transitive, we have $s \; R \; \hat{q}$. Thus Duplicator can directly take $\hat{s} := \hat{q}$. Finally, trivially $\hat{q} \in F \implies \hat{s} \in F$, as required by the winning condition. $\square$

**Lemma 7.** *For any $R$, let $T \subseteq \tau_0(R)$ be any appealing fragment of $\tau_0(R)$. If $R \subseteq T$ (i.e., $R$ is improving), then $T \subseteq \tau_1(R)$.*

*Proof.* Let $R, T$ as in the statement of the lemma, and let $V := \tau_1(R)$. We have to show $T \subseteq V$. Let $q\ T\ p$, and let Spoiler choose $\hat{p}$ and $p'$ with $p\ R\ \hat{p}$ and $\hat{p} \xrightarrow{a} p'$, as required by the definition of $V$. Then, as $R \subseteq T$ by assumption, and $T$ being transitive, we have $q\ T\ \hat{p}$. Therefore, by the definition of $T$, Duplicator can choose $\hat{q}$ and $q'$ with $q\ R\ \hat{q}$ and $\hat{q} \xrightarrow{a} q'$. Since $T$ is self-respecting, we have $p'\ T\ q'$. Finally, $\hat{p} \in F \implies \hat{s} \in F$ by the definition of $T$. Therefore, Duplicator is winning, and $q\ V\ p$. $\square$

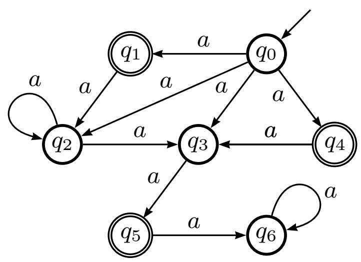



**Fig. 7.** Quotienting w.r.t. appealing fragments of $\tau_0^{\text{de}}$ is incorrect, already for unary automata. We have $q_3 \sqsubseteq_{\text{bw}}^{\text{di}} q_2$ and $q_4 \sqsubseteq_{\text{bw}}^{\text{di}} q_2$, and the relation $T := \{(q_i, q_i) \mid 0 \leq i \leq 6\} \cup \{(q_i, q_6) \mid 0 \leq i \leq 6\} \cup \{(q_i, q_j) \mid 2 \leq i, j \leq 4\} \cup \{(q_i, q_5) \mid 2 \leq i \leq 4\}$ is an appealing fragment of $\tau_0^{\text{de}}(\sqsubseteq_{\text{bw}}^{\text{di}})$. (In particular, $q_3\ \tau_0^{\text{de}}(\sqsubseteq_{\text{bw}}^{\text{di}})\ q_4$ since $q_3$ can "jump" to $q_2$.) The equivalence induced by $T$ identifies the states $q_2, q_3, q_4$, but this is incorrect as the resulting automaton would accept the spurious word $a^\omega$.

**Lemma 8.** *For any $R$, $\tau_1^{de}(R)$ is transitive.*

*Proof.* A complete and formal proof of transitivity requires the machinery of logbooks and composition of (winning) strategies, which is a standard tool for delayed simulation (for more details see, e.g., [7]). Here, we highlight the ingredients pertinent to $\tau_1^{\text{de}}$.

Let $T := \tau_1^{\text{de}}(R)$, and let $r\ T\ q\ T\ p$. We have to show $r\ T\ p$. Let $G_0$ be the game between $r$ and $q$, let $G_1$ be the game between $q$ and $p$, and let $G$ be the outer game between $r$ and $p$.

The idea is that Duplicator plays $G$ and at the same time updates $G_0, G_1$ accordingly. At round $i$, if the $G$-configuration is $\langle r_i, p_i \rangle$, then there exists $q_i$ s.t. the $G_0$ configuration is $\langle r_i, q_i \rangle$ and the $G_1$ configuration is $\langle q_i, p_i \rangle$.

Let Spoiler choose $\hat{p}$ and transition $\hat{p} \xrightarrow{a_i} p_{i+1}$, with $p_i\ R\ \hat{p}$. Since $G_1$-Duplicator is winning, there exist $\hat{q}$ and transition $\hat{q} \xrightarrow{a_i} q_{i+1}$, with $q_i\ R\ \hat{q}$. Similarly, $G_0$-since Duplicator is winning, there exist $\hat{r}$ and transition $\hat{r} \xrightarrow{a_i} r_{i+1}$, with $r_i\ R\ \hat{r}$. Thus, Duplicator can proceed in $G$ by taking the last transition above. The configuratons are updated as follows: The game $G_0$ goes to $\langle r_{i+1}, q_{i+1} \rangle$, $G_1$ goes to $\langle q_{i+1}, p_{i+1} \rangle$ and $G$ goes to $\langle r_{i+1}, p_{i+1} \rangle$.

We now argue that the strategy above is winning. W.l.o.g. we assume that the games $G_0, G_1$ are updated according to a fixed winning strategy. We show that Duplicator is winning in $G$. Assume $\hat{p}_i \in F$. Since $G_1$-Duplicator is playing according to a winning strategy, there exists $k \geq i$ s.t. $\hat{q}_k \in F$. Similarly, as $G_0$-Duplicator is playing according to a winning strategy, there exists $j \geq k \geq i$ s.t. $\hat{r}_j \in F$. Thus, take $j \geq i$ s.t. $\hat{r}_j \in F$, as required. $\square$

**Lemma 19.** *For any transitive $R$, $R \subseteq \tau_1^{de}(R)$.*

*Proof.* Immediate from $R \subseteq \tau_1(R)$ by Lemma 18, and $\tau_1(R) \subseteq \tau_1^{\mathrm{de}}(R)$ by definition. $\square$

**Theorem 6.** *If $R$ is a jumping-safe $F$-respecting preorder, then $\tau_1^{de}(R)$ is jumping-safe.*

*Proof.* Assume that $R$ is a jumping-safe, $F$-respecting preorder, and let $T := \tau_1^{\mathrm{de}}(R)$. We have to show that $T$ is jumping-safe. During the proof we refer to Figure 8, hereafter called "the diagram". Let $w = a_0 a_1 \cdots \in \Sigma^\omega$, and let $\pi$ be an initial $T$-jumping path

$$\pi = q_0 \; T \; q_0^F \; T \; \hat{q}_0 \xrightarrow{a_0} q_1 \; T \; q_1^F \; T \; \hat{q}_1 \xrightarrow{a_1} q_2 \cdots , \quad q_0 \in I \; .$$

See the blue path in the diagram. We inductively show how to build the rest of the diagram, and then we use this construction for showing that $T$ is jumping-safe.

Formally, we inductively build a sequence $\rho_0, \rho_1, \ldots, \rho_i$ such that, for any $k \leq i$, $\rho_k$ is a $T$-ordered $k+4$-tuple of states representing the $k$-th layer of the diagram,

$$\rho_k = s_k^0 \; T \; s_k^1 \; T \; \cdots \; T \; s_k^{k-1} \; T \; s_k^k \; T \; q_k \; T \; q_k^F \; T \; \hat{q}_k \; .$$

Two successive layers are in relations with transitions as follows (cf. the diagram):

$$\forall \, (1 \leq h \leq k) \cdot s_k^h \overset{a_k}{\dashrightarrow} s_{k+1}^h, \quad q_k^F \overset{a_k}{\dashrightarrow} s_{k+1}^{k+1}, \quad \hat{q}_k \xrightarrow{a_k} q_{k+1} \; ,$$

where the dashed arrow $x \overset{a}{\dashrightarrow} y$ represents an $R$-jumping transition via some suitable proxy. That is, $x \overset{a}{\dashrightarrow} y$ iff there exists a proxy $\hat{x}$ s.t. $x \; R \; \hat{x}$ and $\hat{x} \xrightarrow{a} y$.

For $i = 0$, just take $s_0^0 := q_0$. Then, the invariant is clearly satisfied, as $q_0 \; T \; q_0^F$ by assumption and $q_0 T \; q_0$ by $T$ being reflexive.

For $i \geq 0$, assume $\rho_0, \rho_1, \ldots, \rho_i$ has already been built. By induction hypothesis, $\rho_i$ is the following $T$-ordered tuple:

$$\rho_i = s_i^0 \; T \; s_i^1 \; T \; \cdots \; T \; s_i^{i-1} \; T \; s_i^i \; T \; q_i \; T \; q_i^F \; T \; \hat{q}_i \; ,$$

The next layer $\rho_{i+1}$,

$$\rho_{i+1} = s_{i+1}^0 \; T \; s_{i+1}^1 \; T \; \cdots \; T \; s_{i+1}^{i-1} \; T \; s_{i+1}^i \; T \; s_{i+1}^{i+1} \; T \; q_{i+1} \; T \; q_{i+1}^F \; T \; \hat{q}_{i+1} \; ,$$

is obtained as follows. The last three components $q_{i+1}, q_{i+1}^F, \hat{q}_{i+1}$ are fixed by the $T$-jumping path $\pi$. The rest is determined next. Since $\hat{q}_i \xrightarrow{a_i} q_{i+1}$, we propagate the transition down the chain, by using the definition of $T$—as indicated by the zigzag arrows in the diagram. As $q_i^F \; T \; \hat{q}_i$, there exists an $R$-jumping transition $q_i^F \overset{a_i}{\dashrightarrow} q' \; T \; q_{i+1}$. Take $s_{i+1}^{i+1} := q'$. Similarly, from $s_i^i \; T \; q_i \; T \; q_i^F$ there exists $s_i^i \overset{a_i}{\dashrightarrow} q'' \; T \; s_{i+1}^{i+1}$. Take $s_{i+1}^i := q''$. Clearly, one can build all the remaining states down to $s_{i+1}^0$ in the same way, thus completing layer $i+1$ in the diagram. This concludes the inductive step in the definition of $\rho_{i+1}$.

*Remark.* We assume that each time a new $T$-game starts from configuration $\langle q_i^F, q_i \rangle$, Duplicator fixes a winning strategy, and alway plays accordingly.

We now prove that final states are "propagated" in the diagram right-to-left, top-to-bottom: Formally, we show that, for any $i \geq 0$, if $q_i^F \in F$, then there exists $j \geq i$ s.t. $\hat{s}_j^0 \in F$, where $\hat{s}_j^0$ is the proxy witnessing $s_j^0 \overset{a_j}{\dashrightarrow} s_{j+1}^0$. Assume $q_i^F \in F$. Then, since $R$ is $F$-respecting, $\hat{q}_i^F \in F$, where $\hat{q}_i^F$ is the proxy witnessing $q_i^F \overset{a_i}{\dashrightarrow} s_{i+1}^{i+1}$. Since $s_i^i \; T \; q_i^F$, by the definition of $\tau_1^{\text{de}}$ and by the above remark, there exists $j_0 \geq i$ s.t. $\hat{s}_{j_0}^i \in F$, where $\hat{s}_{j_0}^i$ is the proxy witnessing $s_{j_0}^i \overset{a_{j_0}}{\dashrightarrow} s_{j_0+1}^i$. But $s_{j_0}^{i-1} \; T \; s_{j_0}^i$, therefore there exists $j_1 \geq j_0$ s.t. $\hat{s}_{j_1}^{i-1} \in F$, and so on ... until we reach index $j_i \geq j_{i-1}$, for which $\hat{s}_{j_i}^0 \in F$. Thus, take $j := j_i$.

We are finally ready to prove that $T$ is jumping-safe. Notice that the leftmost path in the diagram represents an initial $R$-jumping path $\pi'$,

$$\pi' = s_0^0 \; R \; \hat{s}_0^0 \xrightarrow{a_0} s_1^0 \; R \; \hat{s}_1^0 \xrightarrow{a_1} \cdots, \; s_0^0 = q_0 \in I \; .$$

Since $R$ is jumping-safe, there exists an infinite sequence of initial finite paths $\pi_0, \pi_1, \ldots$ s.t. $\text{last}(\pi_i) \; R \; s_i^0$. Since $R$ is transitive, $R \subseteq T$ by Lemma 19. Therefore, $\text{last}(\pi_i) \; T \; s_i^0$. By $s_i^0 \; T \; q_i$ and transitivity, we obtain $\text{last}(\pi_i) \; T \; q_i$. Therefore, the same sequence $\pi_0, \pi_1, \ldots$ can be taken as a witness for $T$ being jumping-safe. Finally, since $\pi$ is fair, i.e., $q_i^F \in F$ for infinitely many $i$'s, then $\pi'$ is fair, as final states are "propagated" (shown above). Since $R$ is jumping-safe, we conclude that $\pi_0, \pi_1, \ldots$ is coherent. $\square$

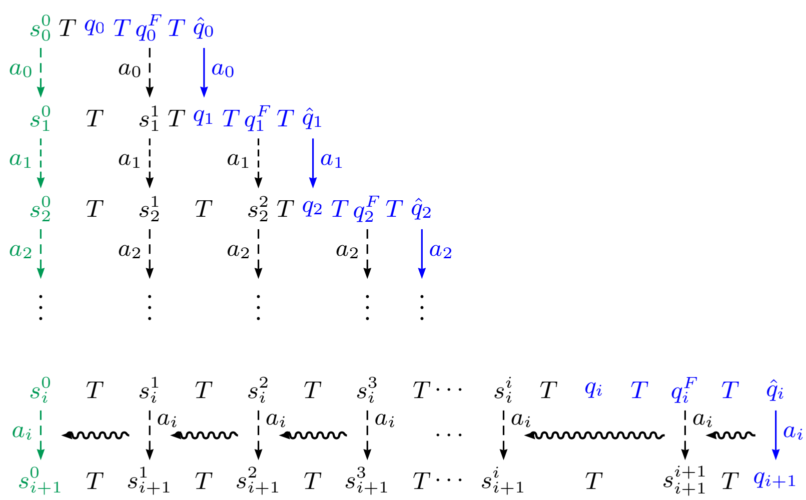


**Fig. 8.** Construction for the proof of Theorem 6.

By using similar techniques, it is possible to show that repeated application of $\tau_1^{\text{de}}$ does not give coarser relations. This is analogous of what proved in Lemma 6 for $\tau_0$. The proof of this fact is omitted.

**Lemma 20.** *For any preorder $R$, $\tau_1^{de}(\tau_1^{de}(R)) \subseteq \tau_1^{de}(R)$.*

## D Computing $\tau_1^{\mathrm{de}}(R)$

In this section we give an algorithm for computing $\tau_1^{\mathrm{de}}(R)$ from Section 5.1, obtained as an extension of the classical algorithm for computing delayed simulation [6]. We assume that the relation $R$ has already been computed. We build a game graph where Duplicator has a Büchi winning objective.

We enrich configurations from the basic semantic game for $\tau_1^{\mathrm{de}}(R)$ with an obligation bit recording whether Duplicator has to visit an accepting state. Formally, Spoiler's positions are of the form $\langle s, q, b\rangle$, with $q, s \in Q$ and $b \in \{0, 1\}$, and Duplicator's positions are of the form $\langle s, q, \hat{b}, a, q'\rangle$, with $q, s, q' \in Q$, $a \in \Sigma$ and $\hat{b} \in \{0, 1\}$. Spoiler can pick a move $(\langle s, q, b\rangle, \langle s, q, \hat{b}, a, q'\rangle) \in \Gamma_0'$ if there exists $\hat{q} \in Q$ s.t. $qR\hat{q} \xrightarrow{a} q'$, and $\hat{b} = 1$ if $\hat{q} \in F$ and $b$ otherwise. Similarly, Duplicator can pick a move $(\langle s, q, \hat{b}, a, q'\rangle, \langle s', q', b'\rangle) \in \Gamma_1'$ if there exists $\hat{s} \in Q$ s.t. $sR\hat{s} \xrightarrow{a} s'$, and $b' = 0$ if $\hat{s} \in F$ and $\hat{b}$ otherwise. The objective for Duplicator is to ensure that the winning bit is $0$ infinitely often, that is, every obligation to visit an accepting state is eventually met. Formally, the winning condition is

$$W' = \{\langle s_0, q_0, b_0\rangle\langle s_1, q_1, b_1\rangle \cdots \mid \forall i \geq 0 \cdot \exists j \geq i \cdot b_j = 0\} .$$

Let $\mathsf{CPre}$ be a controlled predecessor operator for Duplicator, defined as

$$\mathsf{CPre}(X) = \{x \mid \forall (x, y) \in \Gamma_0' \cdot \exists (y, z) \in \Gamma_1' \cdot z \in X\} .$$

That is, $x = \langle s, q, b\rangle \in \mathsf{CPre}(X)$ if Duplicator can force the game in $X$ in one step from configuration $x$. Then, the winning region for Duplicator can be computed by evaluating the following fixpoint:

$$V = \nu X \cdot \mu Y \cdot [b = 0] \cap \mathsf{CPre}(X) \cup \mathsf{CPre}(Y) ,$$

where with $[b = 0]$ we have indicated the set of configurations with no obligation pending, i.e., $[b = 0] = \{\langle q, s, b\rangle \mid q, s \in Q, b = 0\}$. Finally, $s\ \tau_1^{\mathrm{de}}(R)\ q$ holds iff $\langle s, q, 0\rangle \in V$.

## E Proof of Theorem 1

First, we define yet another refinement transformer, called *fixed-word delayed transformer* $\tau_1^{\text{fx-de}}$, which is the same as $\tau_1^{\mathrm{de}}$, with the only difference that Spoiler has to reveal the whole input word $w = a_0 a_1 \cdots$ in advance. Notice that $\tau_1^{\text{fx-de}}$, though not efficiently computable in general, has properties very similar to $\tau_1^{\mathrm{de}}$. In particular, the proof of Theorem 6 works as it is for the lemma below.

**Lemma 21.** *If $R$ is a jumping-safe $F$-respecting preorder, then, $\tau_1^{\text{fx-de}}(R)$ is jumping-safe.*

**Theorem 1.** $\sqsubseteq_{fx}^{de}$ *is good for quotienting.*

*Proof.* Directly from Lemma 21, since $\sqsubseteq_{\text{fx}}^{\text{de}}$ is (the transpose of) $\tau_1^{\text{fx-de}}$ applied to the identity relation.

### E.1 A multipebble hierarchy converging to $\sqsubseteq_{\mathbf{fx}}^{\mathbf{de}}$

WARNING: this section is based on a multipebble encoding of fx-de simulation which is only suitable for fair containment. The problem is that here a configuration is a tuple pair $(q, b, f)$, where $b$ is a state and $b, f$ are suitable colorings for a set of states. The problem is that for fx-de simulation, a configuration should be a *set* of tuples $(q, s, c, g)$, with $q$ a state (Spoiler's position), $s$ a state (Duplicator's position) and $c, g$ natural numbers (the actual colorings of $s$).

Thus the encoding below is *wrong*. While the encoding can be fixed as described above, we find unnatural to describe the sets of tuples in a multipebble language. Thus, we drop the multipebble analogy at this point and spare the content below for future investigation.

*A multipebble hierarchy converging to $\sqsubseteq_{fx}^{de}$.* The we introduce the following linear order between colors: $0 \leq_{\mathsf{c}} 2 \leq_{\mathsf{c}} \cdots \leq_{\mathsf{c}} 1 \leq_{\mathsf{c}} 3 \leq_{\mathsf{c}} \cdots$, i.e., even colors are smaller than odd colors, and colors with the same parity are ordered as usual. For a set of colors $X \subseteq \mathbb{N}$, we define $\min^{\leq_{\mathsf{c}}} X$ to be the $\leq_{\mathsf{c}}$-minimum in $X$ if $X$ is not empty, and $\perp$ otherwise.

A *pebble coloring* is a function $f : Q \mapsto [2v] \cup \{\perp\}$. Let $\mathcal{C}_Q^v = ([2v] \cup \{\perp\})^Q$ be the set of all pebble colorings. The *domain* $\mathsf{dom}\ f$ of a pebble coloring $f \in \mathcal{C}_Q^v$ is the set of states where $f$ has non-$\perp$ value, i.e., $\mathsf{dom}\ f = \{s \in Q \mid f(s) \neq \perp\}$. For an integer $k \geq 1$, a $k$-domain is a domain of size at most $k$, and a $k$-pebble coloring is a pebble coloring whose domain is a $k$-domain. Given a coloring $f$ and a subset of its domain $\mathbf{s} \subseteq \mathsf{dom}\ f$, the *restriction* of $f$ to $\mathbf{s}$, noted $f|_{\mathbf{s}}$, has domain $\mathbf{s}$, and on $\mathbf{s}$ agrees with $f$. For any $s \in Q$, let $\mathbb{1}_s^v \in \mathcal{C}_Q^v$ be the *initial pebble coloring*, defined as $\mathbb{1}_s^v(p) = 2v$ if $p = s$ and $\mathbb{1}_s^v(p) = \perp$ otherwise. For two pebble colorings $f, g : Q \mapsto [2v] \cup \{\perp\}$ let $f \leq g$ iff $\mathsf{dom}\ f = \mathsf{dom}\ g$ and $\forall s \in \mathsf{dom}\ f \cdot f(s) \leq g(s)$.

Let $k, v \geq 1$. The multipebble progress-delayed simulation game is defined as $G_{(k,v)}^{\text{pde}}(q, s) = (P^{\text{pde}}, P_0^{\text{pde}}, P_1^{\text{pde}}, p_I^{\text{pde}}, \Gamma^{\text{pde}}, \Gamma_0^{\text{pde}}, \Gamma_1^{\text{pde}}, W^{\text{pde}})$, where $P_0^{\text{pde}} = Q \times \{0, 1, \perp\}^Q \times \mathcal{C}_Q^v$ is the set of positions of Spoiler, $P_1^{\text{pde}} = Q \times \Sigma \times Q \times \{0, 1, \perp\}^Q \times \mathcal{C}_Q^v$ is the set of positions of Duplicator, $p_I^{\text{pde}} = (q, b, \mathbb{1}_s^v)$ is the initial position, where $b$ is defined only on $s$ and equals 1 if $q \in F \wedge s \notin F$ and 0 otherwise, and $\Gamma_0^{\text{pde}}, \Gamma_1^{\text{pde}}, W^{\text{pde}}$ are determined as follows.

Spoiler can select a move $(\langle q, b, f\rangle, \langle q, a, q', b', f'\rangle) \in \Gamma_0^{\text{pde}}$ iff $q \xrightarrow{a} q'$, and $b', f'$ satisfy the constraints below. Let $b'$ be the unique function which has as domain the $a$-successors of $\mathsf{dom}\ b$ and, for all $s'$ in its domain,

$$b'(s') = \begin{cases} 0 & \text{if } s' \in F \\ 1 & \text{if } s' \notin F, q' \in F \\ \min\{b(s) \mid s \xrightarrow{a} s'\} & \text{otherwise.} \end{cases}$$

Intuitively, $b'$ assigns value $1$ to $s'$ if either Spoiler is accepting and Duplicator is not, or all predecessors have value $1$. When a state gets value $1$ we say that it is *active*. Notice that $b'$ is uniquely determined by $b$. Then, $f'$ has the same domain as $b'$ and,

$$\forall \left(s \xrightarrow{a} s'\right) \quad f'(s') \leq f(s) \quad \wedge \quad (f'(s') \text{ odd } \implies b'(s') = 1)\ . \qquad (\dagger)$$

That is, the value of $f$ is nonincreasing along every path and Spoiler can "threat" Duplicator only on active positions. Notice that $f'$ is not uniquely determined and, therefore, it is under the control of Spoiler.

Finally, Duplicator restricts the domain of both $b'$ and $f'$ to one of cardinality at most $k$. This induces a move $(\langle q, a, q', b', f' \rangle, \langle q', b'', f'' \rangle) \in \Gamma_1^{\text{pde}}$, where $\mathsf{dom}\ b'' = \mathsf{dom}\ f'' \subseteq \mathsf{dom}\ b'$, $|\mathsf{dom}\ b''| \leq k$ and $b''$ ($f''$) agrees with $b'$ ($f'$, resp.) on its domain.

Before defining $W^{\text{pde}}$, we need some preparation. Given an infinite sequence $\pi = \langle q_0, b_0, f_0 \rangle \langle q_1, b_1, f_1 \rangle \cdots \in \left(P_0^{\text{pde}}\right)^\omega$ and a round $j \geq 0$, we say that a state $s \in \mathsf{dom}\ f_j$ *has been even* in $\pi$ since some previous round $i \leq j$, written $\mathsf{even}_j^i(s, \pi)$ iff $f_j(s)$ is even and, if $i < j$, then there exists $\hat{s} \in \mathsf{dom}\ f_{j-1}$ s.t. $\hat{s} \xrightarrow{a_{j-1}} s$ and $\mathsf{even}_{j-1}^i(\hat{s}, \pi)$. We say that $f_j$ is *good since round* $i \leq j$ in $\pi$, written $\mathsf{good}_j^i(f_j, \pi)$, iff at round $j$ there exists a state $s \in \mathbf{s}_j$ s.t. $s$ has been even since round $i$. Duplicator wins a play $\pi$ if $f_i$ is good since some previous round in $\pi$, for all but finitely many $i$'s, i.e.,

$$W^{\text{pde}} = \{\pi = \langle q_0, b_0, f_0 \rangle \langle q_1, b_1, f_1 \rangle \cdots \in \left(P_0^{\text{pde}}\right)^\omega \mid \exists i \cdot \forall j \geq i \cdot \mathsf{good}_j^i(f_j, \pi)\}\ .$$

NOTE: by König's Lemma there exists an infinite path which is eventually always even.

We write $q \sqsubseteq_{(k,v)}^{\text{pde}} s$ iff Duplicator wins the simulation game $G_{(k,v)}^{\text{pde}}(q, s)$, we write $q \sqsubseteq_k^{\text{pde}} s$ iff $q \sqsubseteq_{(k,v)}^{\text{pde}} s$ holds for all $v \geq 0$, and we write $q \sqsubseteq_k^{\text{pde'}} s$ iff $q \sqsubseteq_{(k,n)}^{\text{pde}} s$.

Immediately from the definition of the progress-delayed simulation game we have that more pebbles help Duplicator, while higher bounds for the maximal color help Spoiler.

**Lemma 22.** *For any $k_0 \leq k_1$ and $v_0 \leq v_1$, $\sqsubseteq_{(k_0,v_1)}^{pde} \subseteq \sqsubseteq_{(k_1,v_0)}^{pde}$.*

In particular, we have the following hierarchy of simulations

$$\sqsubseteq_1^{\text{pde}} \subseteq \sqsubseteq_2^{\text{pde}} \subseteq \cdots \subseteq \sqsubseteq_n^{\text{pde}},$$

which clearly converges at level $n$, for an automaton with $n$ states. There are three interesting facts about the above hierarchy: 1) the first element actually coincides with the usual delayed simulation, 2) it is pointwise bigger than the multipebble hierarchy from [5] and 3) the limit is fixed-word delayed simulation.

**Lemma 23.** *We have that 1) $\sqsubseteq_1^{pde} = \sqsubseteq_1^{de}$, 2) for any $k$, $\sqsubseteq_k^{de} \subseteq \sqsubseteq_k^{pde}$, and 3) $\sqsubseteq_n^{pde} = \sqsubseteq_{fx}^{de}$.*

*Proof.*

Moreover, if we allow Spoiler to "threat" Duplicator only once, i.e., by bounding the allowed colors by 2, then we get delayed containment

**Lemma 24.** *We have that $\sqsubseteq_{(n,1)}^{pde} = \subseteq^{de}$.*

*Proof.*

$$
\begin{array}{ccccccccccc}
\sqsubseteq_1^{\text{de}} & \subseteq & \sqsubseteq_2^{\text{de}} & \subseteq & \sqsubseteq_3^{\text{de}} & \subseteq & \cdots & \subseteq & \sqsubseteq_n^{\text{de}} & & \\
\| & & \cap\!| & & \cap\!| & & & & \cap\!| & & \\
\sqsubseteq_1^{\text{pde}} & \subseteq & \sqsubseteq_2^{\text{pde}} & \subseteq & \sqsubseteq_3^{\text{pde}} & \subseteq & \cdots & \subseteq & \sqsubseteq_n^{\text{pde}} & = & \sqsubseteq_{\text{fx}}^{\text{de}} \\
\cap\!| & & \cap\!| & & \cap\!| & & & & \| & & \\
\sqsubseteq_1^{\text{pde'}} & \subseteq & \sqsubseteq_2^{\text{pde'}} & \subseteq & \sqsubseteq_3^{\text{pde'}} & \subseteq & \cdots & \subseteq & \sqsubseteq_n^{\text{pde'}} & &
\end{array}
$$

**Fig. 9.**

### E.2 Uniform bounds on maximal colors

Assume that Spoiler is winning in the game $G^{\text{pde}}_{(n,v)}(q_0, s_0)$ with $v \geq n$. The aim of this section is to show that Spoiler is winning in $G^{\text{pde}}_{(n,n)}(q_0, s_0)$ as well. Notice that, when Duplicator has as much as $n$ pebbles under her control, we can assume w.l.o.g. that she always selects maximal domains, i.e., only transitions of the form $(\langle q, a, q', b', f'\rangle, \langle q', b', f'\rangle) \in \Gamma_1^{\text{pde}}$ are picked by her. Let $\delta$ be a winning strategy for Spoiler. We associate to $\delta$ an infinite $L$-labelled tree $(T_\delta, t_\delta)$, with labelling domain $L = P_0^{\text{pde}} \times Q$, by applying the following two rules:

- $\varepsilon \in T_\delta$ and $t_\delta(\epsilon) = (q_0, b_0, f_0, s_0)$, where $\langle q_0, b_0, f_0\rangle = p_I^{\text{pde}}$ and $s_0$ is the unique element in the domain of $f_0$.
- Assume $u$ has length $i$, $u \in T_\delta$, $t_\delta(u) = (q, b, f, s)$, and let $\pi$ be the labelling in $\left(P_0^{\text{pde}}\right)^*$ of the unique path going from the root to $u$, with the last component stripped away. Let $\delta(\pi) = \langle q, a, q', b', f'\rangle$[6], and let $S = \{v_0, \ldots, v_{h-1}\} \subseteq L$ be the set of all possible next configurations, where a configuration $(q', b', f', s')$ is in the set $S$ iff $s \xrightarrow{a} s' \in \mathsf{dom}\ f'$. (Notice that we assume that Duplicator plays "maximally".) Then, for any $0 \leq m < h$, $um \in T_\delta$ and $t_\delta(um) = v_m$.

*Remark 3.* Nodes at the same depth agree on the first three components. I.e., if $u_0$ and $u_1$ have the same length, then $t_\delta(u_0) = (q_0, b_0, f_0, s_0)$ and $t_\delta(u_1) = (q_1, b_1, f_1, s_1)$ imply $q_0 = q_1, b_0 = b_1$ and $f_0 = f_1$.

Notice that two identically-labelled nodes actually generate isomorphic subtrees. This is due to the fact Duplicator is assumed to play maximally (and maximal strategies are memoryless), and to the fact that $\delta$ only depends on $(q, b, f)$, which are unique for each level in the tree. Therefore, we can identify nodes with the same label and "compress" the tree $(T_\delta, t_\delta)$ into a DAG $G_\delta = (V_\delta, E_\delta)$, where $V_\delta \subseteq L \times \mathbb{N}$ is such that $(q, b, f, s, i) \in V_\delta$ iff there exists $u \in T_\delta$ of length $i$ s.t. $t_\delta(u) = (q, b, f, s)$, and $((q, b, f, s, i), (q', b', f', s', i')) \in E_\delta$ iff $i' = i + 1$ and there exists $u \in T_\delta$ of length $i$ such that $t_\delta(u) = (q, b, f, s)$ and $t_\delta(um) = (q', b', f', s')$ for some $m \in \mathbb{N}$.

For a DAG $G \subseteq G_\delta$, we say that a vertex $v = (q, b, f, s, i) \in G$ is *endangered* in $G$ if only finitely many vertices are reachable from $v$ in $G$, and *active* if $b(s) = 1$; then, a vertex $v$ is said to be *safe* in $G$ if the only vertices reachable from $v$ in $G$ are active.

[6] Notice that $\delta$, as defined here, is an *adversary-blind strategy* in the sense of Section 2.

Consider the following decreasing sequence of graphs $G_0 \supseteq G_1 \supseteq \cdots$, defined as $G_0 = G_\delta$ and, for $i \geq 0$,

$$G_{2i+1} = G_{2i} \setminus \{v \mid v \text{ is endangered in } G_{2i}\}$$
$$G_{2i+2} = G_{2i+1} \setminus \{v \mid v \text{ is safe in } G_{2i+1}\} \ .$$

Define the *girth of* $G$ *at level* $i$ to be the number of vertices $(q, b, f, l)$ in $G$ with $l = i$, and define the *width of* $G$ to be the maximal girth occurring for infinitely many levels. By Remark 3 the width of $G_0$ is bounded by $n$ (as it is its girth at any level). Using a standard argument based on König's Lemma [15], it is possible to prove that when passing from $G_{2i+1}$ to $G_{2i+2}$ at least one infinite path is removed from $G_{2i+1}$. Therefore, for any $i$, the width of $G_{2i}$ is at most $n - i$, which implies that $G_{2n}$ is finite and $G_{2n+1}$ is empty. Thus, a vertex $v \in G_\delta$ happens to be either endangered in $G_{2i}$ or safe in $G_{2i+1}$, and we assign a rank to $v$ accordingly:

$$\mathsf{rank}_\delta(v) = \begin{cases} 2i & \text{if } v \text{ is endangered in } G_{2i} \\ 2i+1 & \text{if } v \text{ is safe in } G_{2i+1} \end{cases}$$

By the previous discussion, ranks are bounded by $2n$.

*Useful properties of ranks.* Ranks are nonincreasing along branches of $G_\delta$.

**Lemma 25.** *For any two vertices $v, v'$ in $G_\delta$, if $v'$ is reachable from $v$, then* $\mathsf{rank}_\delta(v') \leq \mathsf{rank}_\delta(v)$.

In every infinite branch of $G_\delta$ the rank eventually settles into an odd number.

**Lemma 26.** *In every infinite branch of $G_\delta$, there exists a vertex $\hat{v}$ with an odd rank s.t. all vertices $v'$ in the branch reachable from $\hat{v}$ have* $\mathsf{rank}_\delta(v') = \mathsf{rank}_\delta(v)$.

The proofs of the two lemmas above are substantially the same as the ones in Lemmas 3.5 and 3.6 of [15].

**Theorem 9.** *For any $v \geq n$, $\sqsubseteq^{pde}_{(n,n)} = \sqsubseteq^{pde}_{(n,v)}$.*

*Proof.* The inclusion $\supseteq$ holds by Lemma 22. For the other direction, we show that if Spoiler wins, then she can win with bound $2n$ on the maximal color.

Assume that Spoiler is winning in $G^{\text{pde}}_{(n,v)}(q_0, s_0)$, and let $\delta$ be any winning strategy for her. We define a new strategy $\delta'$ for Spoiler in the same game which uses colors bounded by $2n$; that is, $\delta'$ will be winning even in $G^{\text{pde}}_{(n,n)}(q_0, s_0)$.

Intuitively, $\delta'$ chooses transitions $q \xrightarrow{a} q'$ according to $\delta$, and assigns colors according to the corresponding ranks in $G_\delta$. Formally, let $\pi = p^0_0 p^1_0 p^0_1 p^1_1 \cdots p^0_k$ be a $\delta'$-conform partial play ending in $\langle q, b, f\rangle$, and let $\delta(\pi) = \langle q, a, q', b', g'\rangle$. Then, $\delta'(\pi)$ is defined as $\langle q, a, q', b', f'\rangle$, where $f'$ has the same domain as $g'$, and, for any $s' \in \mathsf{dom}\ f'$, $f'(s')$ equals the rank of $(q', b', f', s', k + 1)$ in $G_\delta$.

First, we show that $\delta'$ defines valid moves. Indeed, $f'(s') \leq f(s)$ holds since ranks are nonincreasing (by Lemma 25), and if $f'(s')$ is an odd rank, then, by the definition of rank, the corresponding vertex in $G_\delta$ is safe, which implies that $b'(s') = 1$. Therefore, property (†) holds, and $(\langle q, b, f\rangle, \langle q, a, q', b', f'\rangle) \in \Gamma^{\text{pde}}_0$ is a valid move.

Second, we show that $\delta'$ is winning. By Lemma 26, all paths eventually get an odd rank, which immediately implies the winning condition for Spoiler.

**Lemma 27.** *For any $v \geq n$, $\sqsubseteq^{\textit{pde}}_{(1,n)} = \sqsubseteq^{\textit{pde}}_{(1,v)}$.*

*Proof.*

We leave it open to establish a uniform bound for $v$ in the case $1 < k < n$. In any case, we can still prove that $\sqsubseteq^{\text{pde}}_{,}\ k$ is GFQ. Since $\sqsubseteq^{\text{pde}}_{(n,n)}$ is GFQ (by Lemma 23 and Theorem 1), and since $\sqsubseteq^{\text{pde}}_{(k,n)} \subseteq \sqsubseteq^{\text{pde}}_{(n,n)}$ (by Lemma 22), it follows by Corollary **??** that $\sqsubseteq^{\text{pde}}_{,}\ k$ is GFQ.

### E.3 Computing fixed-word delayed simulation

We transform a progress-delayed simulation game into a simpler game where Duplicator has a co-Büchi winning condition. The idea is to annotate the current position with the set of states which have been even since some previous round [**?**]. If the set becomes empty, a new phase starts and the set is filled in with even states. Duplicator wins if, from some point on, the set of even states always stays nonempty.

Let $G^{\text{pde}}_{(k,v)}(q_0, s_0) = (P^{\text{pde}}, P^{\text{pde}}_0, P^{\text{pde}}_1, p^{\text{pde}}_I, \Gamma^{\text{pde}}, \Gamma^{\text{pde}}_0, \Gamma^{\text{pde}}_1, W^{\text{pde}})$ be a progress-delayed simulation game. We define a new game

$$G^{\text{pde'}}_{(n,v)}(q_0, s_0) = (P^{\text{pde'}}, P^{\text{pde'}}_0, P^{\text{pde'}}_1, p^{\text{pde'}}_I, \Gamma^{\text{pde'}}, \Gamma^{\text{pde'}}_0, \Gamma^{\text{pde'}}_1, W^{\text{pde'}}) \ ,$$

where $P^{\text{pde'}}_0 = P^{\text{pde}}_0 \times 2^Q$, $P^{\text{pde'}}_1 = P^{\text{pde}}_1 \times 2^Q$, $p^{\text{pde'}}_I = \langle q_0, b_0, f_0, \{s_0\}\rangle$, where $\langle q_0, b_0, f_0\rangle$ is the initial configuration $p^{\text{pde}}_I$ in $G^{\text{pde}}_{(k,v)}(q_0, s_0)$, and transitions are defined as follows. $(\langle q, b, f, \mathbf{o}\rangle, \langle q, a, q', b', f', \mathbf{o}'\rangle) \in \Gamma^{\text{pde'}}_0$ iff $(\langle q, b, f\rangle, \langle q, a, q', b', f'\rangle) \in \Gamma^{\text{pde}}_0$ and $\mathbf{o}'$ is the unique subset of $\mathsf{dom}\ f'$ s.t.

$$\mathbf{o}' = \begin{cases} \{o' \in \mathsf{dom}\ f' \mid \exists o \in \mathbf{o} \cdot o \xrightarrow{a} o' \wedge f'(o') \text{ even}\} & \text{if } \mathbf{o} \neq \emptyset \\ \{o' \in \mathsf{dom}\ f' \mid \exists o \in \mathsf{dom}\ f \cdot o \xrightarrow{a} o' \wedge f'(o') \text{ even}\} & \text{otherwise.} \end{cases}$$

$(\langle q, a, q', b', f', \mathbf{o}'\rangle, \langle q', b'', f'', \mathbf{o}''\rangle) \in \Gamma^{\text{pde'}}_1$ iff $(\langle q, a, q', b', f'\rangle, \langle q', b'', f''\rangle) \in \Gamma^{\text{pde}}_1$ and $\mathbf{o}'' = \mathbf{o}' \cap \mathsf{dom}\ f''$. Finally, $\pi = \langle q_0, b_0, f_0, \mathbf{o}_0\rangle\langle q_1, b_1, f_1, \mathbf{o}_1\rangle \cdots \in \left(P^{\text{pde'}}_0\right)^\omega$ is in $W^{\text{pde'}}$ iff $\mathbf{o}_i \neq \emptyset$ for all but finitely many $i$'s.

The number of states in $G^{\text{pde'}}_{(k,v)}(q_0, s_0)$ is bounded by $2 \cdot n^2 \cdot |\Sigma| \cdot (n+1)^k \cdot (3v)^k$. This can be seen as follows. Clearly $|P| = \left|P^{\text{pde}}_0\right| + \left|P^{\text{pde}}_1\right| \leq 2 \cdot \left|P^{\text{pde}}_1\right|$. Each $\langle q, a, q', b', f'\rangle \in P^{\text{pde}}_1$ can be obtained by first selecting $q$, $a$ and $q'$, for which there are clearly at most $n^2 \cdot |\Sigma|$ choices. Then, a $k$-domain for $b'$ has to be selected (or for $f'$, since they have the same domain). Let this domain have size $h \leq k$, and notice that there are at most $\binom{n}{h}$ options. Then, for any $s'$ in the domain, one can select a pair $(b'(s'), f'(s'))$. By construction, if $b'(s')$ is 0, then $f'(s')$ is even. Therefore, there are at most $(v + 2v)^h$ further options. Thus, one gets the following formula

$$\sum_{h=0}^{k} \binom{n}{h} \cdot (3v)^h \ ,$$

which can be bounded by $(n+1)^k(3v)^k$ using the estimate $\binom{n}{h} \leq (n+1)^k$.

Therefore, for each fixed $k, v \geq 1$, $\sqsubseteq^{\text{pde}}_{(k,v)}$ can be computed in polynomial time.

# F Alternative algorithm for computing $\tau_1^{\mathrm{de}}(R)$

We now present an alternative algorithm for computing $\tau_1^{\mathrm{de}}(R)$, inspired from [13], which avoids to enrich the game graph with a winning bit.

The game graph is the same as in Section 5, which we repeat here for clarity. Spoiler's positions are of the form $\langle s, q\rangle$, with $q, s \in Q$, and Duplicator's positions are of the form $\langle s, q, a, q'\rangle$, with $q, s, q' \in Q$ and $a \in \Sigma$. Spoiler can pick a move $(\langle s, q\rangle, \langle s, q, a, q'\rangle) \in \Gamma_0$ if there exists $\hat{q} \in Q$ s.t. $qR\hat{q} \xrightarrow{a} q'$. Similarly, Duplicator can pick a move $(\langle s, q, a, q'\rangle, \langle s', q'\rangle) \in \Gamma_1$ if there exists $\hat{s} \in Q$ s.t. $sR\hat{s} \xrightarrow{a} s'$. The winning condition is $W = \{\langle s_0, q_0\rangle\langle s_1, q_1\rangle \cdots \mid \forall i \geq 0 \cdot q_i \in F \implies \exists j \geq i \cdot s_j \in F\}$.

We take the point of view of Spoiler, and introduce three controllable predecessor operators: Let $X, Y$ be two sets of Spoiler's configurations. Then,

– $\mathsf{CPre}(X)$ is the usual 1-step forcing operator (from the point of view of Spoiler):

$$\mathsf{CPre}(X) = \bigcup_{a \in \Sigma} \{\langle s, q\rangle \mid \exists (qR\hat{q} \xrightarrow{a} q') \cdot \forall (sR\hat{s} \xrightarrow{a} s') \cdot \langle s', q'\rangle \in X\}\ ,$$

– $\mathsf{CPre}^0(X, Y)$ is a 1-step forcing operator where Spoiler can ensure that either Duplicator does not visit an accepting state and the game goes in $X$, or the game goes in $Y$:

$$\mathsf{CPre}^0(X, Y) = \bigcup_{a \in \Sigma} \left\{ \langle s, q\rangle \mid \exists (qR\hat{q} \xrightarrow{a} q') \cdot \forall (sR\hat{s} \xrightarrow{a} s') \cdot \begin{array}{c} \hat{s} \notin F \wedge \langle s', q'\rangle \in X \\ \vee \\ \langle s', q'\rangle \in Y \end{array} \right\}\ ,$$

– and $\mathsf{CPre}^1(X)$ is a 1-step forcing operator where Spoiler can ensure to visit an accepting state and to go in $X$, without Duplicator visiting an accepting state:

$$\mathsf{CPre}^1(X) = \bigcup_{a \in \Sigma} \left\{ \langle s, q\rangle \mid \exists (qR\hat{q} \xrightarrow{a} q') \cdot \forall (sR\hat{s} \xrightarrow{a} s') \cdot \begin{array}{c} \hat{q} \in F \wedge \hat{s} \notin F \\ \wedge \\ \langle s', q'\rangle \in X \end{array} \right\}\ .$$

For a set $Y$, let $\mathsf{Force}(Y)$ be the set of states from which Spoiler can force $X$ in a finite number of steps, that is,

$$\mathsf{Force}(Y) = \mu X \cdot Y \cup \mathsf{CPre}(X)\ .$$

Then, the winning region of Spoiler is given by the following formula:

$$U = \mu W \cdot \mathsf{Force}(W \cup \mathsf{CPre}^1(\nu X \cdot \mathsf{CPre}^0(X, W)))\ .$$