Multipebble Simulations for Alternating Automata - (Extended Abstract)

Citation for published version:

Published In:
CONCUR 2010 - CONCURRENCY THEORY

Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.
Multipebble Simulations for Alternating Automata
(Extended Abstract)

Lorenzo Clemente and Richard Mayr
LFCS, School of Informatics, University of Edinburgh, UK

Abstract. We study generalized simulation relations for alternating Büchi automata (ABA), as well as alternating finite automata. Having multiple pebbles allows the Duplicator to “hedge her bets” and delay decisions in the simulation game, thus yielding a coarser simulation relation. We define \((k_1, k_2)\)-simulations, with \(k_1/k_2\) pebbles on the left/right, respectively. This generalizes previous work on ordinary simulation (i.e., \((1, 1)\)-simulation) for nondeterministic Büchi automata (NBA) in [3] and ABA in [4], and \((1, k)\)-simulation for NBA in [2]. We consider direct, delayed and fair simulations. In each case, the \((k_1, k_2)\)-simulations induce a complete lattice of simulations where \((1, 1)\) and \((n, n)\)-simulations are the bottom and top element (if the automaton has \(n\) states), respectively, and the order is strict. For any fixed \(k_1, k_2\), the \((k_1, k_2)\)-simulation implies \(\omega\)-language inclusion and can be computed in polynomial time. Furthermore, quotienting an ABA w.r.t. \((1, n)\)-delayed simulation preserves its language. Finally, multipebble simulations yield new insights into the Miyano-Hayashi construction [10] on ABA.

1 Introduction

We consider simulation relations on (alternating) finite- and infinite word automata: nondeterministic finite automata (NFA), alternating finite automata (AFA), nondeterministic Büchi automata (NBA) and alternating Büchi automata (ABA). Simulation preorder is a notion of semantic comparison of two states, called left state and right state, in automata, where the larger right state can match all moves of the smaller left one in a stepwise way. Simulation preorder implies language inclusion on NFA/AFA/NBA/ABA [3, 4], but not vice-versa. While checking language inclusion is PSPACE-complete for all these classes of automata [7, 11], the simulation relation can be computed in polynomial time [3, 4].

Checking simulation preorder between two states can be presented as a game with two players, Spoiler and Duplicator, where Spoiler tries to prove that the simulation relation does not hold while Duplicator has the opposite objective. In every round of the simulation game, Spoiler chooses a transition from the current left state and Duplicator must choose a transition from the current right state which has the same action label. Duplicator wins iff the game respects the accepting states in the automata, and different requirements for this yield finer or coarser simulation relations. In direct simulation, whenever the left state is accepting, the right state must be accepting. In delayed simulation, whenever the left state is accepting, the right state must be eventually accepting. In fair simulation, if the left state is accepting infinitely often, then the right state must be accepting infinitely often. For finite-word automata, only direct simulation is meaningful, but for Büchi automata delayed and fair simulation yield coarser relations; see [3] for an overview.
These notions have been extended in two directions. Etessami [2] defined a hierarchy of \((1, k)\) multipebble simulations on NBA. Intuitively, the \(k\) pebbles on the right side allow Duplicator to “hedge her bets” and thus to delay making decisions. This extra power of Duplicator increases with larger \(k\) and yields coarser simulation relations.

A different extension by Wilke and Fritz [4] considered simulations on ABA. In an ABA, a state is either existential or universal. The idea is that Spoiler moves from existential left states and universal right states, and dually for Duplicator.

Our contribution. We consider \((k_1, k_2)\)-simulations on ABA, i.e., with multiple pebbles on both sides: \(k_1\) on the left and \(k_2\) on the right. Intuitively, Duplicator controls pebbles on universal states on the left and existential states on the right (and dually for Spoiler). This generalizes all previous results: the \((1, k)\)-simulations on NBA of [2] and the \((1, 1)\)-simulations on ABA of [4].

For each acceptance condition (direct, delayed, fair) this yields a lattice-structured hierarchy of \((k_1, k_2)\)-simulations, where \((1, 1)\)- and \((n, n)\)-simulations are the bottom and top element if the automaton has \(n\) states. Furthermore, the order is strict, i.e., more pebbles make the simulation relation strictly coarser in general. For each fixed \(k_1, k_2 \geq 0\), \((k_1, k_2)\)-simulations are computable in polynomial time and they imply language inclusion (over finite or infinite words, depending on the type of simulation).

Quotienting AFA w.r.t. \((k_1, k_2)\)-simulation preserves their language. We also provide a corresponding result for ABA by showing that quotienting ABA w.r.t. \((1, n)\)-delayed simulation preserves the \(\omega\)-language. This is a non-trivial result, since a naïve generalization of the definition of semielective-quotients [4] does not work. We provide the correct notion of semielective-quotients for \((1, n)\)-simulations on ABA, and show its correctness. Moreover, unlike for NBA [2], quotienting ABA w.r.t. \((1, k)\) delayed simulation for \(1 < k < n\) does not preserve their language in general.

Finally, multipebble simulations have close connections to various determinization-like constructions like the subset construction for NFA/AFA and the Miyano-Hayashi construction [10] on ABA. In particular, multipebble simulations yield new insights into the Miyano-Hayashi construction and an alternative correctness proof showing an even stronger property.

2 Preliminaries and Basic Definitions

Automata. An alternating Büchi automaton (ABA) \(Q\) is a tuple \((Q, \Sigma, q_I, \Delta, E, U, F)\), where \(Q\) is a finite set of states, \(\Sigma\) is a finite alphabet, \(q_I\) is the initial state, \(\{E, U\}\) is a partition of \(Q\) into existential and universal states, \(\Delta \subseteq Q \times \Sigma \times Q\) is the transition relation and \(F \subseteq Q\) is the set of accepting states. We say that a state \(q\) is accepting if \(q \in F\). We use \(n\) to denote the cardinality of \(Q\). A nondeterministic Büchi automaton (NBA) is an ABA with \(U = \emptyset\), i.e., where all choices are existential. We say that \(Q\) is complete iff \(\forall (q, a) \in Q \times \Sigma. \exists (q, a, q') \in \Delta\).

An ABA \(Q\) recognizes a language of infinite words \(L^\omega(Q)\). The acceptance condition is best described in a game-theoretic way [5]. Given an input word \(w \in \Sigma^\omega\), the acceptance game \(G^\omega(Q, w)\) is played by two players, Pathfinder and Automaton. Existential states are controlled by Automaton, while Pathfinder controls universal states. Automaton wins the game \(G^\omega(Q, w)\) iff she has a winning strategy s.t., for any
Pathfinder counter-strategy, the resulting computation visits some accepting state in \( F \) infinitely often. The language \( L^\omega(Q) \) recognized by \( Q \) is defined as the set of words \( w \in \Sigma^\omega \) s.t. Automaton wins \( G^\omega(Q, w) \). See [4] for a formal definition.

If we view an ABA \( Q \) as an acceptor of finite words, then we obtain an alternating finite automaton (AFA). For \( w = w_0 \ldots w_m \in \Sigma^* \), the finite acceptance game \( G^{fin}(Q, w) \) is defined as above for \( G^\omega(Q, w) \), except that the game stops when the last symbol \( w_m \) of \( w \) has been read, and Automaton wins if the last state is in \( F \). \( L^{fin}(Q) \) is defined in the obvious way. An alternating transition system (ATS) \( Q \) is an AFA where all states are accepting, and \( T_f(Q) := L^{fin}(Q) \) is its trace language. When we just say “automaton”, it can be an ABA, AFA or ATS, depending on the context.

If \( Q \) is a set, with \( 2^Q \) we denote the set of subsets of \( Q \), and, for any \( k \in \mathbb{N} \), with \( 2^Q \times k \) we denote the subset of \( 2^Q \) consisting of elements of cardinality at most \( k \). When drawing pictures, we represent existential states by \( \mathcal{E} \) and universal states by \( \mathcal{U} \).

**Multiplebble simulations.** We define multipebble simulations in a game-theoretic way. The game is played by two players, Spoiler and Duplicator, who play in rounds. The objective of Duplicator is to show that simulation holds, while Spoiler has the complementary objective. We use the metaphor of pebbles for describing the game: We call a pebble existential if it is on an existential state, and universal otherwise; Left if it is on the LHS of the simulation relation, and Right otherwise. Intuitively, Spoiler controls existential Left pebbles and universal Right pebbles, while Duplicator controls universal Left pebbles and existential Right pebbles. The presence of \( > 1 \) pebbles in each side is due to the further ability of Duplicator to split pebbles to several successors. Moreover, Duplicator always has the possibility of “taking pebbles away”. Since not all available pebbles have to be on the automaton, \( k + 1 \) pebbles are at least as good as \( k \).

Formally, let \( Q \) be an alternating automaton, \( q_0 \in 2^Q \times k_1 \) a \( k_1 \)-set and \( s_0 \in 2^Q \times k_2 \) a \( k_2 \)-set. We define the basic \((k_1, k_2)\)-simulation game \( G_{(k_1, k_2)}(q_0, s_0) \) as follows. Let \( \Gamma^{Sp} \) and \( \Gamma^{Dup} \) be a set of actions (or transitions) for the two players (to be specified below). In the initial configuration \( (q_0, s_0) \), Left pebbles are on \( q_0 \) and Right pebbles on \( s_0 \). If the current configuration at round \( i \) is \( (q_i, s_i) \), then the next configuration \( (q_{i+1}, s_{i+1}) \) is determined as follows:

- Spoiler chooses a transition \( (q_i, s_i, a_i, q'_i, s'_i) \in \Gamma^{Sp} \).
- Duplicator chooses a transition \( (q_i, s_i, a_i, q'_i, s'_i, q_{i+1}, s_{i+1}) \in \Gamma^{Dup} \).

We now define the two transition relations. Let \( q^E := q \cap E \) be the set of existential states in \( q \), and define \( q^U, s^E, s^U \) similarly. Let \( P_1 := 2^Q \times k_1 \times 2^Q \times k_2 \) and \( P_0 := \Sigma \times 2^Q \times k_1 \times 2^Q \times k_2 \). \( \Gamma^{Sp} \subseteq P_1 \times P_0 \times P_1 \) models Spoiler’s moves: \( (q, s, a, q', s') \in \Gamma^{Sp} \) iff Spoiler chooses \( a \) as the next input symbol, and

- \( q' \) is obtained from \( q^E \) by choosing a successor for each pebble in \( q^E \). Formally, \( q' = \{ \text{select}(\Delta(q, a)) \mid q \in q^E \} \), where \( \text{select}(r) \) chooses an element in \( r \).
- Similarly, \( s' \) is obtained from \( s^U \) by choosing a successor for each pebble in \( s^U \).

Duplicator’s moves are of the form \((q, s, a, q', s', q'', s'') \in \Gamma^{Dup} \subseteq P_1 \times P_0 \times P_1 \):

- \( q'' \) is a non-empty \( k_1 \)-subset of \( q' \cup \Delta(q^U, a) \), and
- \( s'' \) is a non-empty \( k_2 \)-subset of \( s' \cup \Delta(s^E, a) \).
Notice that Duplicator is always allowed to “take pebbles away”, and to “hedge her bets” by splitting pebbles into different successors. We say that a pebble on state \( q \) is stuck if \( q \) has no \( a \)-successor (where \( a \) is clear from the context).

We now formally define strategies. A strategy for Spoiler is a function \( \delta : P_1^* P_1 \rightarrow P_0 \) compatible with \( I^{Sp} \), i.e., for any \( (\pi \cdot (q, s)) \in P_1^* P_1 \), \( \delta(\pi \cdot (q, s)) = (a, q', s') \) implies \( (q, s, a, q', s') \in I^{Sp} \). Similarly, a strategy for Duplicator is a function \( \sigma : P_1^* P_1 \rightarrow (P_0 \rightarrow P_1) \) compatible with \( I^{Duf} \), i.e., for any \( \pi \in P_1^* P_1 \) and \( (a, q', s') \in P_0 \), \( \sigma(\pi)(a, q', s') = (q'', s'') \) implies \( (q, s, a, q', q'', s'') \in I^{Duf} \). A play \( \pi = \langle q_0, s_0 \rangle \langle q_1, s_1 \rangle \cdots \in P_1^* \) is a finite or infinite sequence of configurations in \( P_1 \). For a word \( w = a_0 a_1 \cdots \in \Sigma^* \cup \Sigma^\omega \) s.t. \( |w| = |\pi| - 1 \) (with \( |\pi| = \omega = \omega - 1 \) if \( \pi \in \Sigma^\omega \)), we say that a play \( \pi \) is \( \sigma \)-conform to \( w \) iff, for any \( i < |\pi| \), there exists some \( (q_i, s_i, a_i, q'_i, s'_i) \in I^{Sp} \) s.t. \( \sigma((q_0, s_0) \cdots (q_i, s_i))(a_i, q'_i, s'_i) = (q_{i+1}, s_{i+1}) \).

Intuitively, \( \sigma \)-conform plays are those plays which originate when Duplicator’s strategy is fixed to \( \sigma \); \( \delta \)-conform plays, for \( \delta \) a Spoiler’s strategy, are defined similarly. Below, both strategies are fixed, and the resulting, unique play is conform to both.

The game can halt prematurely, for pebbles may get stuck. In this case, the winning condition is as follows: If there exists a Left pebble which cannot be moved, then Duplicator wins. Dually, if no Right pebble can be moved, then Spoiler wins.

**Remark 1.** Our winning condition differs from the one in [4] when pebbles get stuck. There, the losing player is always the one who got stuck. If we let Duplicator win when Spoiler is stuck on a universal Right pebble, we would obtain a simulation which does not imply language containment. (Notice that “simulation implies containment” is proved in [4] under the assumption that pebbles do not get stuck.) Furthermore, the condition in [4] is unnecessarily strong when Duplicator is stuck on a universal Left pebble, where letting Spoiler win is too conservative. Our definition generalizes the correct winning condition to multiple pebbles, for which we prove “simulation implies containment” without further assumptions.

In all other cases, we have that all Left pebbles can be moved and at least one Right pebble can be moved, and the two players build an infinite sequence of configurations \( \pi = \langle q_0, s_0 \rangle \langle q_1, s_1 \rangle \cdots \in P_1^* \). The winning condition is defined in terms of a predicate \( C(\pi) \) on \( \pi \). Different choices of \( C(\pi) \) lead to different notions of simulation.

1. **Ordinary \((k_1, k_2)\)-simulation.** The acceptance condition is ignored, and Duplicator wins as long as the game doesn’t halt: \( C(\pi) : \iff \text{true} \).
2. **Existential direct \((k_1, k_2)\)-simulation.** Duplicator wins if, whenever every \( q \in Q_1 \) is accepting, then some \( s \in S_i \) is accepting:

\[
C(\pi) : \iff (\forall i. q_i \subseteq F \implies s_i \cap F \neq \emptyset).\
\]

3. **Universal direct \((k_1, k_2)\)-simulation.** Duplicator wins if, whenever some \( q \in Q_1 \) is accepting, then every \( s \in S_i \) is accepting:

\[
C(\pi) : \iff (\forall i. q_i \cap F \neq \emptyset \implies s_i \subseteq F).\
\]

As we will see, ordinary simulation is used for ATSSs, while existential and universal direct simulation are used for automata over finite and infinite words, respectively.
The winning condition for delayed and fair simulation requires some technical preparation, which consists in the notion of being existentially/universally good since some previous round. Given the current round \( m \), we say that a state \( q \in q_m \) has seen a state \( \hat{q} \) since some previous round \( i \leq m \), written \( \text{has}_i q_m (q, \hat{q}) \), iff either 1) \( q = \hat{q} \), or \( i < m \) and there exists \( q' \in q_{m-1} \) s.t. 2.1) \( q \in \Delta(q', a_{m-1}) \), and 2.2) has\(_i\) \( q_{m-1} (q', \hat{q}) \). Dually, we write \( \text{can't avoid}^i q_m (q, \hat{q}) \) iff either 1) \( q = \hat{q} \), or \( i < m \) and, for all \( q' \in q_{m-1} \), \( q \in \Delta(q', a_{m-1}) \) implies \( \text{can't avoid}^i q_{m-1} (q', \hat{q}) \). We overload the notation on the set of accepting states, and we write \( \text{has}_i q_m (q, F) \) to mean that \( q \) has seen some \( \hat{q} \in F \); and similarly for \( \text{can't avoid}^i q_m (q, F) \). Finally, we say that \( s_j \) is existentially good since round \( i \leq j \), written \( \text{good}^3(s_j, i) \), if at round \( j \) every state in \( s_j \) has seen an accepting state since round \( i \), and \( j \) is the least round for which this holds [2]. Similarly, we say that \( q_j \) is universally good since round \( i \leq j \), written \( \text{good}^3(s_j, i) \), if at round \( j \) every state in \( q_j \) cannot avoid an accepting state since round \( i \), and \( j \) is the least round for which this holds. Formally,

\[
\text{good}^3(s_j, i) \iff \left( \forall s \in s_j, \text{has}_i q_j (s, F) \right) \land \\
\forall j'. \left( \forall s' \in s_j, \text{has}_i q_j (s', F) \right) \implies j' \geq j
\]

\[
\text{good}^3(s_j, i) \iff \left( \forall s \in s_j, \text{can't avoid}_i q_j (s, F) \right) \land \\
\forall j'. \left( \forall s' \in s_j, \text{can't avoid}_i q_j (s', F) \right) \implies j' \geq j
\]

We write \( \text{good}^3(s_j) \), omitting the second argument, when we just say that \( s_j \) is good since some previous round. For a path \( \pi = s_0s_1 \ldots \), we write \( \text{good}^3(\pi, \infty) \), with the second argument instantiated to \( i = \infty \), to mean that \( \text{good}^3(s_j) \) holds for infinitely many \( j \)’s; and similarly for \( \text{good}^3(\pi, \infty) \).

We are now ready to define delayed and fair simulations.

4. **Delayed \((k_1, k_2)\)-simulation.** Duplicator wins if, whenever \( q_i \) is universally good, then there exists \( j \geq i \) s.t. \( s_j \) is existentially good since round \( i \):

\[
C(\pi) : \iff \forall i. \text{good}^3(q_i) \implies \exists j \geq i. \text{good}^3(s_j, i)
\]

5. **Fair \((k_1, k_2)\)-simulation.** Duplicator wins if, whenever there are infinitely many \( i \)’s s.t. \( q_i \) is universally good, then, for any such \( i \), there exists \( j \geq i \) s.t. \( s_j \) is existentially good since round \( i \):

\[
C(\pi) : \iff \text{good}^3(\pi_0, \infty) \implies \left( \forall i. \text{good}^3(q_i) \implies \exists j \geq i. \text{good}^3(s_j, i) \right)
\]

where \( \pi_0 = q_0q_1 \ldots \) is the projection of \( \pi \) onto its first component.

We will denote the previous acceptance conditions with \( x \in \{ o, \exists i, \forall i, \exists i, \forall i \} \), and the corresponding game is denoted as \( G_{(k_1, k_2)}(q_0, s_0) \).

**Remark 2.** Notice that the condition for fair simulation is equivalent to the following simpler one: If \( q_i \) is universally good since some previous round infinitely often, then \( s_j \) is existentially good since some previous round infinitely often: \( C'(\pi) : \iff \text{good}^3(\pi_0, \infty) \implies \text{good}^3(\pi_1, \infty) \), where \( \pi_1 = s_0s_1 \ldots \) is the projection of \( \pi \) onto its second component.
We are now ready to define the simulation relation \( \sqsubseteq_{(k_1, k_2)}^x \) with \( x \) as above. We say that a \( k_2 \)-set \( s \) \( (k_1, k_2) \)-simulates a \( k_1 \)-set \( q \), written \( q \sqsubseteq_{(k_1, k_2)}^x s \), if Duplicator has a winning strategy in \( G^x_{(k_1, k_2)}(q, s) \). We overload the simulation relation \( \sqsubseteq_{(k_1, k_2)}^x \) on singletons: \( q \sqsubseteq_{(k_1, k_2)}^x s \iff \{ q \} \sqsubseteq_{(k_1, k_2)}^x \{ s \} \). For two automata \( A \) and \( B \), we write \( A \sqsubseteq_{(k_1, k_2)} B \) for \( q_A^k \sqsubseteq_{(k_1, k_2)}^x q_B^k \), where the simulation is actually computed on the disjoint union of \( A \) and \( B \). If \( \sqsubseteq_{(k_1, k_2)}^x \) is a simulation, then its transitive closure is defined as \( \sqsubseteq_{(k_1, k_2)}^x \). Note that, in general, \( \sqsubseteq_{(k_1, k_2)}^x \) is not itself a transitive relation.

**Multipebble simulations hierarchy.** In general, having more pebbles (possibly) gives more power to the Duplicator. This is similar to the \((1, k)\)-simulations for NBA studied in [2], but in our context there are two independent directions of “growing power”.

**Theorem 1.** Let \( x \in \{ \text{o}, \exists \text{di}, \forall \text{di}, \text{de}, \text{f} \} \) and \( k'_1 \geq k_1, k'_2 \geq k_2 \).

1. **Inclusion:** \( \sqsubseteq_{(k_1, k_2)}^x \subseteq \sqsubseteq_{(k'_1, k'_2)}^x \) (In particular, \( \sqsubseteq_{(k_1, k_2)}^x \subseteq \sqsubseteq_{(k'_1, k'_2)}^x \)).

2. **Strictness:** If \( k'_1 > k_1 \) or \( k'_2 > k_2 \), there exists an automaton \( Q' \) s.t. \( \not\sqsubseteq_{(k_1, k_2)}^x \).  

**Proof (Sketch).** Point 1) follows directly from the definitions, since Duplicator can always take pebbles away. Point 2) is illustrated in Figure 1, which holds for any kind of simulation \( x \in \{ \text{o}, \exists \text{di}, \forall \text{di}, \text{de}, \text{f} \} \). \( \square \)

![Fig. 1. Example in which \( q \sqsubseteq_{(2,3)}^x \) \( s \), but \( q \not\sqsubseteq_{(1,2)}^x \) \( s \) for any \( k_1 \leq 2, k_2 \leq 3 \), with \( k_1 \leq 2 \) or \( k_2 < 3 \). The alphabet is \( \Sigma' = \{ a \} \cup \Sigma_1 \), with \( \Sigma_1 = \{ b_1, b_2, c_1, c_2, c_3 \} \). Note that both automata recognize the same language, both over finite and infinite words: \( L_{\text{fin}}^{\text{fin}}(q) = L_{\text{fin}}^{\text{fin}}(s) = a(c_1 + c_2 + c_3) \Sigma^* \) and \( L_{\text{inf}}^{\text{inf}}(q) = L_{\text{inf}}^{\text{inf}}(s) = a(c_1 + c_2 + c_3) \Sigma^* \).

**Theorem 2.** For any \( k_1, k_2 \in \mathbb{N}_{>0} \) and any automaton \( Q \).

1. \( \sqsubseteq_{(k_1, k_2)}^\exists \subseteq \sqsubseteq_{(k_1, k_2)}^\forall \)

2. \( \sqsubseteq_{(k_1, k_2)}^\text{de} \subseteq \sqsubseteq_{(k_1, k_2)}^\text{f} \subseteq \sqsubseteq_{(k_1, k_2)}^\text{f} \).

Moreover, for each containment, there exists \( Q \) s.t. the containment is strict.

**Proof.** The containments follow directly from the definitions. For the strictness, consider again the example in Figure 1, with the modifications below. If no state on the right is accepting, then no simulation holds except ordinary simulation. If \( q \) is accepting, then universal direct simulation does not hold, but delayed simulation does. Finally, if the only accepting state is \( q \), then delayed simulation does not hold, but fair simulation does. Is is easy to generalize this example for any \( k_1, k_2 \in \mathbb{N}_{>0} \). \( \square \)
3 Finite words

Lemma 1. For any automaton $Q$ with $n$ states and states $q, s \in Q$:

1. $q \preceq^{\exists di}(k_1, k_2) s$ implies $L^{\text{fin}}(q) \subseteq L^{\text{fin}}(s)$, for any $k_1, k_2 \in \mathbb{N}_{>0}$.
2. $q \preceq^{o}(k_1, k_2) s$ implies $T(r)(q) \subseteq T(r)(s)$, for any $k_1, k_2 \in \mathbb{N}_{>0}$.
3. $L^{\text{fin}}(q) \subseteq L^{\text{fin}}(s)$ implies $q \preceq^{\exists di}(n, n) s$, provided that $Q$ is complete.
4. $T(r)(q) \subseteq T(r)(s)$ implies $q \preceq^{o}(n, n) s$, provided that $Q$ is complete.

In particular, the last two points above show that existential-direct (resp., ordinary) simulation "reaches" language inclusion (resp., trace inclusion) at $(n, n)$.

Subset constructions. The subset construction is a well-known procedure for determinizing NFAs [7]. It is not difficult to generalize it over alternating automata, where it can be used for eliminating existential states, i.e., to perform the de-existentialization of the automaton. The idea is the same as in the subset construction, except that, when considering $a$-successors of a macrostate (for a symbol $a \in \Sigma$), existential and universal states are treated differently. For existential states, we apply the same procedure as in the classic subset construction, by taking always all $a$-successors. For universal states, each $a$-successor induces a different transition in the subset automaton. This ensures that macrostates can be interpreted purely disjunctively, and the language of a macrostate equals the union over the language of the states belonging to it. Accordingly, a macrostate is accepting if it contains some state which is accepting.

The previous construction can be dualized for de-universalizing finite automata. For an AFA $Q$, let $S^\exists(Q)$ and $S^\forall(Q)$ be its de-existentialization and de-universalization, respectively. (See Definitions 1 and 2 in Appendix B.1.)

The following lemma formalizes the intuition that multiple simulations for AFA in fact correspond to $(1, 1)$-simulations over the appropriate subset-constructions.

Lemma 2. Let $Q_1, Q_2$ be two AFAs over the same alphabet $\Sigma$, with $|Q_1| = n_1$ and $|Q_2| = n_2$. Then, for any $k_1 \leq n_1$ and $k_2 \leq n_2$.

\[
\begin{align*}
Q_1 \preceq^{\exists di}(k_1, n_2) Q_2 & \iff Q_1 \preceq^{\exists di}(k_1, 1) S^\exists(Q_2) \quad (1) \\
Q_1 \preceq^{\exists di}(n_1, k_2) Q_2 & \iff S^\forall(Q_1) \preceq^{\exists di}(1, k_2) Q_2 \quad (2) \\
Q_1 \preceq^{\exists di}(n_1, n_2) Q_2 & \iff S^\forall(Q_1) \preceq^{\exists di}(1, 1) S^\exists(Q_2) \quad (3)
\end{align*}
\]

4 Infinite words

Multiple existence-direct simulation is not suitable for being used for $\omega$-automata, since it does not even imply $\omega$-language inclusion.

Theorem 3. For any $k_1, k_2 \in \mathbb{N}_{>0}$, not both equal to 1, there exist an automaton $Q$ and states $q, s \in Q$ s.t. $q \preceq^{\exists di}(k_1, k_2) s$ holds, but $L^{\omega}(q) \not\subseteq L^{\omega}(s)$.
(a) An example in which \( q \sqsubseteq^{3,4}_{(1,2)} s \) holds, but \( L^\omega(q) \not\subseteq L^\omega(s) \).

(b) An example in which \( L^\omega(q_0) \subseteq L^\omega(s_0) \) holds, but \( q_0 \not\sqsubseteq^{f}_{(n,n)} s_0 \).

Fig. 2. Two examples.

Proof. Consider the example in Figure 2(a). Clearly, \( q \sqsubseteq^{3,4}_{(1,2)} s \) holds, since Duplicator can split pebbles on the successors of \( s \), and one such pebble is accepting, as required by existential-direct simulation. But \( L^\omega(q) \not\subseteq L^\omega(s) \): In fact, \( (ab)c^\omega \in L^\omega(q) = (a(b+c))^\omega \), but \( (ab)c^\omega \not\in L^\omega(s) = ((ab)^*ac)^\omega \).

This motivates the definition of universal-direct simulation, which does imply \( \omega \)-language inclusion, like the coarser delayed and fair simulations.

Theorem 4. For \( x \in \{\forall \text{di}, \text{de}, f\} \), automaton \( Q, k_1, k_2 \in \mathbb{N}_{>0} \) and states \( q, s \in Q \),

\[ q \sqsubseteq^{r}_{(k_1, k_2)} s \quad \text{implies} \quad L^\omega(q) \subseteq L^\omega(s). \]

Unlike in the finite word case, \( \omega \)-language inclusion is not “reached” by the simulations \( \{\forall \text{di}, \text{de}, f\} \). See Figure 2(b) and Appendix C.

Theorem 5. For any \( x \in \{\forall \text{di}, \text{de}, f\} \), there exist an automaton \( Q \) and states \( q_0, s_0 \in Q \) s.t. \( L^\omega(q_0) \subseteq L^\omega(s_0) \) but \( q_0 \not\sqsubseteq^{f}_{(n,n)} s_0 \).

The Miyano-Hayashi construction The Miyano-Hayashi (MH) construction [10] is a subset-like construction for ABAs which removes universal non-determinism, i.e., it performs the de-universalization of \( \omega \)-automata. The idea is similar to the analogous construction over finite words, with extra bookkeeping needed for recording visits to accepting states, which may occur not simultaneously for different runs. A set of obligations is maintained, encoding the requirement that, independently of how universal non-determinism is resolved, an accepting state has to be eventually reached. There is a tight relationship between these obligations and fair multipebble simulation. For an ABA \( Q \), let \( Q_{nd} \) be the de-universalized automaton obtained by applying the MH-construction. (See also Definition 3 in Appendix C.1.)

The following lemma says that the MH-construction produces an automaton which is \((n,1)\)-fair-simulation equivalent to the original one, and this result is “tight” in the sense that it does not hold for either direct, or delayed simulation.

Lemma 3. For any ABA \( Q \), let \( Q_{nd} \) be the NBA obtained according to the Miyano-Hayashi de-universalization procedure applied to \( Q \). Then,

a) \( Q \sqsubseteq^{r}_{(n,1)} Q_{nd} \), for \( x \in \{f, \forall \text{di}\} \), and \( a') \exists \) automaton \( Q^1 \) s.t. \( Q^1 \not\sqsubseteq^{de}_{(n,1)} Q_{nd}^1 \).

b) \( Q_{nd} \sqsubseteq^{r}_{(1,1)} Q \), and \( b') \exists \) automaton \( Q^2 \) s.t. \( Q_{nd}^2 \not\sqsubseteq^{f}_{(1,1)} Q^2 \), for \( x \in \{\text{de}, \forall \text{di}\} \).
Fig. 3. An example showing automata $Q_1$ and $Q_2$ s.t. $Q_1 \preceq^{[\forall \text{di}, \text{de}]}_{(n, 1)} Q_2$ for $x \in \{\forall \text{di, de}\}$. The only difference between $Q_1$ and $Q_2$ is the state $q_{11}$ being accepting in the former and $q'_{11}$ being non-accepting in the latter. Notice that $Q_1^{nd} = Q_2^{nd} = Q_{nd}$.

Since fair simulation implies language inclusion, $Q$ and $Q_{nd}$ have the same language. This constitutes an alternative proof of correctness for the MH-construction.

The MH-construction "preserves" fair simulation in the following sense.

**Lemma 4.** Let $Q$, $S$ be two ABAs. Then, $Q \preceq^f_{(n, 1)} S \iff Q_{nd} \preceq^f_{(1, 1)} S_{nd}$.

**Remark 3.** A weaker version of the "only if" direction of Lemma 4 above, namely $Q \preceq^f_{(1, 1)} S \implies Q_{nd} \preceq^f_{(1, 1)} S_{nd}$ (notice the $(1, 1)$ in the premise), had already appeared in [4]. The same statement for both direct and delayed simulation is false, unlike as incorrectly claimed in [4]. In fact, it can be shown (with an example similar to Figure 3) that there exist automata $Q$ and $S$ s.t. $Q \preceq^f_{(1, 1)} S$, but $Q_{nd} \not\preceq^f_{(1, 1)} S_{nd}$, with $x \in \{\text{di, de}\}$. Finally, the "if" direction of Lemma 4 can only be established in the context of multiple pebbles, and it is new.

**Transitivity.** While most $(k_1, k_2)$-simulations are not transitive, some limit cases are. By defining a notion of join for $(1, n)$- and $(n, 1)$-strategies (see Appendix C.2), we establish that $(1, n)$ and $(n, 1)$ simulations are transitive.

**Theorem 6.** Let $Q$ be an ABA with $n$ states, and let $x \in \{\forall \text{di, de, f}\}$. Then, $\preceq^x_{(1, n)}$ and $\preceq^x_{(n, 1)}$ are transitive.

**Remark 4 (Difficulties for $(n, n)$ transitivity).** We did consider transitivity for $(n, n)$-simulations on ABA, but found two major issues there. The first issue concerns directly the definition of the join of two $(n, n)$-strategies, and this holds for any $x \in \{\forall \text{di, de, f}\}$: The so-called "puppeteering technique", currently used for defining the join for $(1, n)$- and $(n, 1)$-strategies, requires to maintain several games, and to pipe the output from one game to the input of one or more other games. This creates a notion of dependency
between different games. For $(1, n)$ and $(n, 1)$, there are no cyclic dependencies, and we were able to define the joint strategy. However, for $(n, n)$-simulations, there are cyclic dependencies, and it is not clear how the joint strategy should be defined.

The second issue arises from the fact that we further require that the join of two winning strategies is itself a winning strategy. Therefore, the joint strategy needs to carry an invariant which implies the $x$-winning condition, for $x \in \{\forall i, \exists i, \exists i\}$). While such an invariant for $x = \forall i$ is straightforward, it is not clear what the correct invariant should be for either delayed or fair simulation.

5 Quotienting

In the following we discuss how multipebble simulation preorder can be used for state-space reduction of alternating automata, i.e., we discuss under which notions of quotient the quotient automaton recognizes the same language as the original one.

Let $Q = (Q, \Sigma, q_0, \Delta, E, U, F)$ be an alternating automaton, over finite or infinite words. Let $\succeq$ be any binary relation on $Q$, and let $\approx$ be the induced equivalence, defined as $\approx := \succeq \cap (\succeq^{-1})^{-1}$. $[\cdot] : Q \mapsto [Q]$ is the function that maps each element $q \in Q$ to the equivalence class $[q] \in [Q]$ it belongs to, i.e., $[q] := \{q' | q \approx q'\}$. We overload $[P]$ on sets $P \subseteq Q$ by taking the set of equivalence classes.

In all the notions of quotients that will be defined, only the transition relation varies. Let $Q = (Q, \Sigma, q_0, \Delta, E, U, F)$ as follows: $E' := E$, $U' := [Q] \setminus E' = \{[q] | [q] \subseteq U\}$ and $F' = [F]$. We leave $\Delta'$ unspecified at this time, as it will have different concrete instantiations later. Notice that mixed classes, i.e., classes containing both existential and universal states, are declared existential.

The following definitions are borrowed from [4]. We say that $q' \in \Delta(q, a)$ is a $k$-minimal a-successor of $q'$ if there is no strictly $\in_{(1,k)}$-smaller $a$-successor of $q'$, i.e., for any $q'' \in \Delta(q, a)$, $q'' \in_{(1,k)} q'$ implies $q'' \not\in_{(1,k)} q'''$. Similarly, $q' \in \Delta(q, a)$ is a $k$-maximal $a$-successor of $q'$ if for any $q'' \in \Delta(q, a)$, $q'' \in_{(1,k)} q'$ implies $q'' \not\in_{(1,k)} q'$. Let $\min_{k,\sigma}(q)$/$\max_{k,\sigma}(q)$ be the set of minimal/maximal successors.

5.1 Finite words

Let $\preceq$ be any preorder which implies language inclusion over finite words, i.e., $q \preceq s \Rightarrow L^\text{fin}(q) \subseteq L^\text{fin}(s)$. In particular, one can take $\preceq := (\in_{(1,k_1,k_2)})^*$. or even $\preceq$ equal to language inclusion itself. As before, let $\approx$ be the equivalence induced by $\preceq$. It is well known that automata over finite words can be quotiented w.r.t. any preorder which implies language equivalence. Here, we show that not all transitions are needed, and that is sufficient to consider $\preceq$-maximal successors of existential states and $\preceq$-minimal successors of universal states. We define the minimax [4] quotient automaton $Q^m_{\preceq}$ by instantiating the quotient skeleton (see Section 5) with transition relation $\Delta_{\preceq} := \Delta^m_{\preceq}$, where $(([q], a, [q'])) \in \Delta^m_{\preceq}$ iff either

- $[q] \in E'$ and $\exists \tilde{q} \in [q] \cap E, \tilde{q}' \subseteq [q']$ s.t. $(\tilde{q}, a, \tilde{q}') \in \Delta$ and $\tilde{q}' \subseteq \max_{\preceq}(\tilde{q})$, or
- $[q] \in U'$ and $\exists \tilde{q} \in [q], \tilde{q}' \subseteq [q']$ s.t. $(\tilde{q}, a, \tilde{q}') \in \Delta$ and $\tilde{q}' \subseteq \min_{\preceq}(\tilde{q})$. 

10
Notice that transitions from universal states in mixed classes are ignored altogether.

**Lemma 5.** Let $Q$ be any alternating finite automaton, and let $\preceq$ be any preorder which implies finite-language inclusion. Then, for any $q \in Q$, $L^\text{fin}(q) = L^\text{fin}([q]_\preceq)$.

### 5.2 Infinite words

Unlike for finite words, it is well known that quotienting $\omega$-automata w.r.t. $\omega$-language-equivalence does not preserve the $\omega$-language. It has even been shown that quotienting w.r.t. $(1, 1)$-fair (bi)simulation does not preserve the $\omega$-language either [6, 3]. Therefore, one has to look for finer simulations, like delayed or direct simulation. Notice that multiplebble existential-direct simulation cannot be used for quotienting, since it does not even imply $\omega$-language inclusion—see Theorem 3.

**Theorem 7.** For any $k_1, k_2 \in \mathbb{N}^0$ and $x \in \{3\text{d}i, f\}$ there exists an ABA $Q$ s.t. $L^\omega(Q) \neq L^\omega(Q^x)$, with $\approx := \approx^x_{(k_1, k_2)}$. For $x = 3\text{d}i$, $k_1$ and $k_2$ must not be both equal to 1. (Note that $\approx^x_{(1, 1)}$-quotienting does preserve the $\omega$-language.)

Thus, in the following we concentrate on universal-direct and delayed simulation.

**Minimax quotients for universal-direct simulation.** In [4] it has been shown that minimax quotients preserve the $\omega$-language (for direct simulation), and that one can consider just maximal/minimal successors of existential/universal states, respectively. Here, we improve this notion, by showing that, when considering multiple-pebbles, it is not needed to consider every maximal successor of existential states, but it is safe to discard those maximal successors which are $(1, k)$-simulated by a $k$-set of other maximal successors. This suggests the following definition: For $\hat{q} \in E$, $a \in \Sigma$ and $k > 0$, we say that $\hat{q}'$ is a set of $k$-maximal representatives for $a$-successors of $\hat{q}$ iff

$$
\hat{q}' \subseteq \max^{k, \text{d}i}_a(\hat{q}) \wedge \left( \forall q'' \in \left( \max^{k, \text{d}i}_a(\hat{q}) \setminus \hat{q}' \right), \exists q'' \in 2^{\hat{q}'}: q'' \in \approx^{k, \text{d}i}_a \hat{q}' \right)
$$

(4)

Notice that the above definition is non-deterministic, in the sense that there might be different sets of maximal representatives: In this case, one can just take any $\subseteq$-minimal set satisfying Equation 4. In the following, we assume that a set of maximal representatives $\hat{q}'$ has been selected for any $\hat{q} \in E$ and $a \in \Sigma$.

We define the minimax+ quotient automaton $Q^{\text{min+}}$ by instantiating the quotient skeleton (see Section 5) with transition relation $\Delta^\text{min+} := \Delta^{\text{min+}}$, which differs from $\Delta^\text{min}$ just for existential and mixed classes: $([q], a, [q']) \in \Delta^{\text{min+}}$ with $[q] \in E'$ iff

- there exist $\hat{q} \in [q] \cap E$ and $\hat{q}' \in [q']$ s.t. $(\hat{q}, a, \hat{q}') \in \Delta$ and $\hat{q}' \in \hat{q}'$, where $\hat{q}'$ is a fixed set of $k$-maximal representatives for $a$-successors of $\hat{q}$, as defined above.

Our definition of minimax+ quotient differs from the one in [4] also w.r.t. the treatment of mixed classes, as discussed in the following remarks.

**Remark 5.** While in [4] universal states in mixed classes do induce transitions (to minimal elements), in our definition we ignore these transitions altogether. In the setting of $(1, 1)$-simulations these two definitions coincide, as they are shown in [4] to yield exactly the same transitions, but this needs not be the case in our setting: In the context of
multiple-pebbles, one minimal transition from a universal state \( q^U \) might be subsumed by no single transition from some existential state \( q^E \) in the same class, but it is always the case that \( q^E \) has a set of transitions which together subsume the one from \( q^U \) (cf. Lemma 15 in Appendix D.3). In this case, we show that one can in fact always discard the transitions from \( q^U \). Thus, in the context of multiple-pebbles, minimax+ quotients result in less transitions than just minimax quotients from [4].

**Remark 6.** While minimax mixed classes are deterministic when considering (1, 1)-simulations [4], this is not necessarily true when multiple pebbles are used.

**Theorem 8.** \( q \approx^{\forall_{(1,n)}}_{(1,n)} [q]_{\text{in+}}, \text{ where the quotient is taken w.r.t. the transitive closure of } \equiv^{\forall_{(1,k)}}_{(1,k)}, \text{ for any } k \text{ such that } 1 \leq k \leq n. \text{ In particular, } L^\omega(q) = L^\omega([q]_{\text{in+}}). \)

**Semielective quotients for delayed simulation.** It has been shown in [4] that minimax quotients w.r.t (1, 1)-delayed simulation on ABA do not preserve the \( \omega \)-language. The reason is that taking just maximal successors of existential states is incorrect for delayed simulation, since a visit to an accepting state might only occur by performing a non-maximal transition. (This is not the case with direct simulation, where if a simulation-smaller state is accepting, then every bigger state is accepting too.) This motivates the definition of semielective quotients [4], which are like minimax quotients, with the only difference that every transition induced by existential states is considered, not just maximal ones. Except for that, all previous remarks still apply. In particular, in mixed classes in semielective quotients it is necessary to ignore non-minimal transitions from universal states—the quotient automaton would recognize a bigger language otherwise.

While for the (1, 1)-simulations on ABA in [4] it is actually possible to ignore transitions from universal states in mixed classes altogether (see Remark 5), in the context of multiple-pebbles this is actually incorrect, as shown in Figure 5, Appendix D.3. The reason is similar as why non-maximal transitions from existential states cannot be discarded: This might prevent accepting states from being visited. We define the semielective+ quotient automaton \( Q_{\text{se+}} \) by instantiating the quotient skeleton (see Section 5) with \( \Delta_{\text{se+}} := \Delta_{\text{se}} \), where

\[
([q], a, [q']) \in \Delta_{\text{se+}} \iff (q, a, q') \in \Delta \text{ and either } q \in E, \text{ or } q \in U \text{ and } q' \in \min^{n,\text{de}}(q)
\]

**Theorem 9.** \( q \approx^{\text{de}_{(1,n)}}_{(1,n)} [q]_{\text{se+}}, \text{ where the quotient is taken w.r.t. } \equiv^{\text{de}}_{(1,n)}. \text{ In particular, } L^\omega(q) = L^\omega([q]_{\text{se+}}). \)

**Remark 7.** It is surprising that, unlike for NBA [2], quotienting ABA w.r.t. (1, k)-de simulations, for \( 1 < k < n \), does not preserve the language of the automaton in general. The problem is again in the mixed classes, where minimal transitions from universal states can be selected only by looking at the full (1, n)-simulation. See the counterexample in Figure 4, where the dashed transition is present in the (1, k)-quotient, despite being non-(1, n)-minimal.

**Remark 8.** Semielective multiplepebble quotients can achieve arbitrarily high compression ratios relative to semielective 1-pebble quotients, (multiplepebble-)direct minimax quotients and mediated preorder quotients [1] (see Figure 6 in Appendix D.3).
Therefore, actually computes an under-approximation. If the left side is continuously accepting, for example, then the right side might simply have not enough time to satisfy any obligation at all. Therefore, [2] actually computes an under-approximation to delayed simulation.

Fig. 4. (1, k)-semi-elective+ quotients on ABA do not preserve the $\omega$-language for $1 < k < n$ in general. Let $k = 2$. The only two $(1, k)/(1, n)$-equivalent states are $q_u$ and $q_e$, and in the quotient they form a mixed class. $q_t$ is not a $(1, n)$-minimal $a$-successor of $q_u$, but it is a $(1, k)$-minimal successor for $k = 2$. Thus, the only difference between the $(1, n)$- and $(1, k)$-semi-elective+ quotients is that the dashed transition is (correctly) not included in the former, but (incorrectly) included in the latter. Thus the $(1, k)$-semi-elective+ quotient automaton would incorrectly accept the word $w = aaea^* \not\in \mathcal{L}^\omega(q_t) = aaa\{b + c + d\}a^\omega$.

6 Solving Multipebble Simulation Games

In this section we show how to solve the multipebble simulation games previously defined. We encode each simulation game into a 2-player game-graph with an $\omega$-regular winning condition. In the game-graph, Eve will take the role of Duplicator, and Adam the one of Spoiler. A game-graph is a tuple $G = \langle V_E, V_A, \to \rangle$, where nodes in $V_E$ belong to Eve (mimicking Duplicator), and nodes in $V_A$ belong to Adam (mimicking Spoiler). Transitions are represented by elements in $\to \subseteq (V_E \times V_A \cup V_A \times V_E)$, where we write $p \to q$ for $(p, q) \in \to$. Notice that the two players strictly alternate while playing, i.e., the game graph is bipartite. We write $V$ for $V_E \cup V_A$. We introduce the following monotone operator on $2^V$: For any $x \subseteq V_A$, $\text{cpre}(x) := \{v_0 \in V_A \mid \forall v_1 \in V_E. (v_0 \to v_1 \implies \exists v_2 \in x. v_1 \to v_2)\}$, i.e., cpre($x$) is the set of nodes where Eve can force the game into $x$.

We define various game-graphs for solving simulations. We express the winning region of Eve as a $\mu$-calculus fixpoint expression over $V_A$ [8], which can then be evaluated using standard fixpoint algorithms. We derive the desired complexity upper bounds using the following fact:

**Lemma 6.** Let $e$ be a fixpoint expression over a graph $V$, with $|V| \in n^{O(k)}$. Then, for any fixed $k \in \mathbb{N}$, evaluating $e$ can be done in time polynomial in $n$.

For solving direct and fair simulation, we refer the reader to Appendix E. Here, we consider just delayed simulation, which is the most difficult (and interesting).

The natural starting point for defining $G^\text{de}$ is the definition in [2] of the game-graph for computing $(1, k)$-simulations for NBAs. Unfortunately, the game-graph in [2] is actually incorrect: According to the definition of delayed simulation (cf. Section 2), every new obligation encountered when the left side is accepting at some round should be independently satisfied by the right side, which has to be good since that round. Now, the algorithm in [2] just tries to satisfy the most recent obligation, which overrides all the previous ones. This is an issue: If the left side is continuously accepting, for example, then the right side might simply have not enough time to satisfy any obligation at all. Therefore, [2] actually computes an under-approximation to delayed simulation.
We overcome this difficulty by explicitly bookkeeping all pending constraints. This leads to the following definitions. The game-graph for delayed simulation is \( G^{de} = (V^{de}_E, V^{de}_A, \rightarrow^{de}) \), where nodes in \( V^{de}_E \) are of the form \( v_{(q, \text{Bad}, s, \text{Good})} \), and nodes in \( V^{de}_A \) of the form \( v_{(q, \text{Bad}, s, \text{Good}, o, q', s')} \), with \( q, q', s, s' \subseteq Q \). Bad = \((b_1 \supset \cdots \supset b_{m_1})\) and Good = \((g_1 \subset \cdots \subset g_{m_2})\) are two sequences of sets of states from \( Q \), strictly ordered by set-inclusion, which are used to keep track of multiple obligations.

Intuitively, Bad is used to detect when new constraints should be created, i.e., to detect when every \( \text{Left} \) pebble is universally good since some previous round. At each round, a new set of bad pebbles \( b = q \setminus F \) is added to Bad. When accepting states are visited by \( \text{Left} \) pebbles, they are discarded from every set \( b \in \text{Bad} \). When some \( b \) becomes eventually empty, this means that, at the current round, all \( \text{Left} \) pebbles are universally good since some previous round: At this point, \( b \) is removed from Bad, and we say that the red light flashes.

The sequence Good represents a set of constraints to be eventually satisfied. Each \( g \in \text{Good} \) is a set of good pebbles, which we require to “grow” until it becomes equal to \( s \). When Good = \( \emptyset \), there is no pending constraint. Constraints are added to Good when the red light flashes (see above): In this case, we update Good by adding the new empty constraint \( g = \emptyset \). When accepting states are visited by \( \text{Right} \) pebbles, we upgrade every constraint \( g \in \text{Good} \) by adding accepting states. Completed constraints \( g = s \) are then removed from Good, and we say that the green light flashes.

Lemma 7. \(|V^{de}| \leq 2(1 \cdot 2^{(k_1 + k_2) + 1} \cdot (1 + (k_1 + 1)^{k_1 + 1}) \cdot (1 + 2(k_2 + 1)^{k_2 + 1}) \cdot |\Sigma|\).

Transitions in \( G^{de} \) are defined as follows. For any \((q, s, a, q'', s'') \in I^{Sp} \), we have \( v_{(q, \text{Bad}, s, \text{Good})} \rightarrow^{de} v_{(q, \text{Bad}, s, \text{Good}, o, q'', s'')} \), and for \((q, s, a, q'', s', s'') \in I^{Dup} \), we have \( v_{(q, \text{Bad}, s, \text{Good}, o, q'', s'')} \rightarrow^{de} v_{(q', \text{Bad}', s', \text{Good}')}, \) where \( \text{Bad}', \text{Good}' \) are computed according to Algorithm 1 in Appendix E.3.

We have that Eve wins iff every red flash is matched by at least one green flash, and different red flashes are matched by different green ones. This can be checked by verifying that infinitely often either Good = \( \emptyset \) or \( s \in \text{Good} \), i.e., it is not the case that Good contains a constraint that it is not eventually “completed” and discarded. Let \( T = \{v_{(q, \text{Bad}, s, \text{Good})} \mid \text{Good} = \emptyset \lor s \in \text{Good} \} \), and define the initial configuration as

\[
\nu_T = \begin{cases} 
\nu_{(q, (F \setminus x), s, o)} & \text{if } q \setminus F \neq \emptyset \\
\nu_{(q, x, o, (s'' \cap F'))} & \text{otherwise}
\end{cases}
\]

\( q \supseteq_{k_1, k_2} s \) iff \( T \) is visited infinitely often iff \( v_T \in W^{de} = \nu x \mu y (cpre(y) \cup T \cap cpre(x)) \).

Theorem 10. For any fixed \( k_1, k_2 \in \mathbb{N} \), \( x \in \{\nu, \exists, \text{di}, \text{di}, \text{de}, \text{f}\} \) and sets \( q, s \subseteq Q \), deciding whether \( q \supseteq_{(k_1, k_2)} s \) can be done in polynomial time.

7 Conclusions and Future Work

Transitivity for \((n, n)\)-simulations. As discussed at the end of Section 4, composing \((n, n)\) (winning) strategies is apparently much more difficult than in the \((1, n)\) and \((n, 1)\) case. We conjecture that all types of \((n, n)\)-simulations discussed in this paper are transitive, and showing this would conceivably solve the join problem as well.
Quotienting with \((n,1)\)- and \((n,n)\)-simulations. While we have dealt with \((1,n)\)-quotients, we have not considered \((n,1)\)- or \((n,n)\)-quotients. For the latter, one should first solve the associated transitivity problem, and, for both, an appropriate notion of semielective-quotient has to be provided. We have shown that this is already a non-trivial task for \((1,n)\)-simulations on ABA.

Future directions. Our work on delayed simulation has shown that several generalizations are possible. In particular, two issues need to be addressed. The first is the complexity of the structure of the game-graph needed for computing delayed simulation. A possible generalization of delayed simulation involving looser “synchronization requirements” between obligations and their satisfaction might result in simpler game-graphs. The second issue concerns Lemmas 3 and 4: We would like to find a weaker delayed-like simulation for which the counterexample shown there does not hold. This would give a better understanding of the MH-construction.

As in [3], it is still open to find a hierarchy of \((k_1,k_2)\)-multipebble simulations converging to \(\omega\)-language inclusion when \(k_1 = k_2 = n\).

Acknowledgment. We thank K. Etessami and C. Fritz for helpful discussions, and an anonymous referee for suggesting the comparison to mediated preorder [1].

References

A Preliminaries (Section 2)

Theorem 1. Let \( x \in \{0, 3, \text{di, } \forall \text{di, de, f} \} \) and \( k'_1 \geq k_1, k'_2 \geq k_2 \).

1. Inclusion: \( \subseteq^{(k_1, k_2)} \subseteq \subseteq^{(k'_1, k'_2)} \). (In particular, \( \subseteq^{x(1, k_2)} \subseteq \subseteq^{x(k'_1, k'_2)} \))

2. Strictness: If \( k'_1 > k_1 \) or \( k'_2 > k_2 \), there exists an automaton \( Q' \) s.t. \( \subseteq^{x(1, k_2)} \neq \subseteq^{x(k'_1, k'_2)} \).

Proof. 1. This follows directly from the definitions, since having more pebbles can only help Duplicator, who is always allowed to take pebbles away.

2. For showing the strictness of the inclusion, consider the example in Figure 1, for any kind of simulation \( x \in \{0, 3, \text{di, } \forall \text{di, de, f} \} \). This example shows that Duplicator wins by “hedging her bets” on both sides, using 2 pebbles on the left and 3 pebbles on the right. Hence, \( q \not\subseteq^{(2,3)} s \) holds.

To see that the 2 Left pebbles are necessary, it suffices to note that if there were only one Left pebble, then Spoiler could choose either \( b_1 \) or \( b_2 \), and every Right pebble would get stuck. But with two Left pebbles, Spoiler can no longer play either \( b_1 \) or \( b_2 \), since one Left pebble would get stuck, which would be winning for Duplicator.

A similar reasoning for the symbols \( \{c_1, c_2, c_3\} \) shows that 3 Right pebbles are necessary and sufficient for Duplicator to win the game.

It is easy to generalize Figure 1 to more pebbles \( (k_1, k_2) \). Moreover, a similar example can be crafted without using the stuckness condition, but using only the acceptance condition.

\[ \square \]

The following two observations will be useful in later proofs: When pebbles are good infinitely often, then it is the case that they are always good, as stated below.

Lemma 8. Let \( \pi = q_0q_1 \ldots \) be any sequence of \( k \)-sets. Then,

\[
\left( \text{good}^3(\pi, \infty) \implies \forall i \geq 0. \text{good}^3(q_i) \right) \land \left( \text{good}^3(\pi, \infty) \implies \forall i \geq 0. \text{good}^3(q_i) \right).
\]

The following is a consequence of König’s Lemma:

Lemma 9. Let \( \pi = s_0s_1 \ldots \) be an infinite sequence of \( k \)-sets. If \( \text{good}^3(\pi, \infty) \), then there exists an infinite path \( \pi_{\text{acc}} = p_0p_1 \ldots \) s.t. \( i \) for any \( i, p_i \in s_i \), and 2) for infinitely many \( i \)'s, \( p_i \in F \).

Proof. We make use of König’s Lemma: We build an infinite tree which is infinitely branching, hence by König’s Lemma there exists an infinite path \( \pi_{\text{acc}} \) starting from the root, and we show that this path contains infinitely many accepting states. First, we extract a subsequence \( \{s'_j\}_{j \geq 0} \) from \( \{s_i\}_{i \geq 0} \), as follows: \( s'_j := s_{i_j} \), where \( i_0 = 0 \) and, inductively, \( i_j \) is the least index \( i > i_{j-1} \) s.t. \( \text{good}^3(s, i_1) \). It follows that, for any \( j > 0 \), \( \text{good}^3(s'_j, i_j-1) \). For any state \( r \in s'_j \), there is a node \( v(r, j) \) at level \( j \geq 0 \) in the tree. (For example, the root of the tree is \( v(s, 0) \), where \( s \) is the only state in \( s'_0 = s_0 = \{s\} \).) The parenthood relation between nodes is defined as follows: We have that, for any \( r \in s'_{j+1} \), there exists \( r' \in s'_j \) s.t. there exists a path from \( r' \) to \( r \) which visits at least one accepting state. In this case, \( v(r, j + 1) \) is a children of \( v(r', j) \). This tree is infinite and finitely branching. Moreover, the infinite path \( \pi_{\text{acc}} \), whose existence is guaranteed by König’s Lemma, visits accepting states infinitely often. \[ \square \]
B Section 3

Lemma 1. For any automaton \( Q \) with \( n \) states, and states \( q, s \in Q \):

1. \( q \overset{\text{di}}{=}_{(k_1, k_2)} s \) implies \( L_{\text{fin}}^n(q) \subseteq L_{\text{fin}}^n(s) \), for any \( k_1, k_2 \in \mathbb{N}_{>0} \).
2. \( q \overset{\text{ci}}{=}_{(k_1, k_2)} s \) implies \( \text{Tr}(q) \subseteq \text{Tr}(s) \), for any \( k_1, k_2 \in \mathbb{N}_{>0} \).
3. \( L_{\text{fin}}^n(q) \subseteq L_{\text{fin}}^n(s) \) implies \( q \overset{\text{ci}}{=}_{(n, n)} s \), provided that \( Q \) is complete.
4. \( \text{Tr}(q) \subseteq \text{Tr}(s) \) implies \( q \overset{\text{di}}{=}_{(n, n)} s \), provided that \( Q \) is complete.

In particular, the last two points above show that existential-direct (resp., ordinary) simulation "reaches" language inclusion (resp., trace inclusion) at \((n, n)\).

Proof. Point 2 follows from Point 1), and Point 4) follows from Point 3), since when the set of accepting state is the full set of states, i.e., \( F = Q \), ordinary and direct simulation coincide, and the trace language equals the finite language in this case.

For Points 1) and 3), we defer their proof at the end of the next section. \(\square\)

B.1 Subset constructions

Below, we give a formal definition for the de-existentialization and de-universalization procedures for AFAs.

Definition 1. Given an AFA \( Q = (Q, \Sigma, q_1, F, E, U) \) with \( |Q| = n \), the existential \( n \)-subset construction yields a purely universal finite automata

\[
S^3(Q) := (Q', \Sigma, \{q_1\}, F', E', U') ,
\]

where \( Q' := 2^Q, F' := \{ P \in Q' \mid P \cap F \neq \emptyset \}, E' := \emptyset, U' := Q' \), and the transition relation \( \Delta' \subseteq 2^Q \times \Sigma \times 2^Q \) satisfies: \( (P, a, R) \in \Delta' \) iff there exists a choice function select : \( P \cap U \times \Sigma \mapsto Q \) which fixes an element in \( \Delta(p, a) \) for any universal state \( p \in P \cap U \), and \( R := \bigcup_{p \in P \cap E} \Delta(p, a) \cup \{ \text{select}(p, a) \mid p \in P \cap U \} \).

Intuitively, the choice function select resolves the universal choice, and then we take the union over all possible resolutions of the existential choice.

Definition 2. Given an AFA \( Q = (Q, \Sigma, q_1, F, E, U) \) with \( |Q| = n \), the universal \( n \)-subset construction yields a purely existential finite automata

\[
S^\forall(Q) := (Q', \Sigma, \{q_1\}, F', E', U') ,
\]

where \( Q' := 2^Q, F' := \{ P \in Q' \mid P \subseteq F \}, E' := Q', U' := \emptyset \), and \( (P, a, R) \in \Delta' \) iff there exists a choice function select : \( P \cap E \times \Sigma \mapsto Q \) which fixes an element in \( \Delta(p, a) \) for any existential state \( p \in P \cap E \), and \( R := \bigcup_{p \in P \cap E} \Delta(p, a) \cup \{ \text{select}(p, a) \mid p \in P \cap E \} \).

Lemma 2. Let \( Q_1, Q_2 \) be two AFAs over the same alphabet \( \Sigma \), with \( |Q_1| = n_1 \) and \( |Q_2| = n_2 \). Then, for any \( k_1 \leq n_1 \) and \( k_2 \leq n_2 \),

\[
Q_1 \overset{\text{di}}{=}_{(k_1, k_2)} Q_2 \quad \iff \quad Q_1 \overset{\text{di}}{=}_{(k_1, k_2)} S^3(Q_2) \quad (S1)
\]

\[
Q_1 \overset{\text{di}}{=}_{(n_1, k_2)} Q_2 \quad \iff \quad S^\forall(Q_1) \overset{\text{di}}{=}_{(1, k_2)} Q_2 \quad (S2)
\]

\[
Q_1 \overset{\text{di}}{=}_{(n_1, n_2)} Q_2 \quad \iff \quad S^\forall(Q_1) \overset{\text{di}}{=}_{(1, 1)} S^3(Q_2) . \quad (S3)
\]
Proof. First notice that Equation (S3) follows by subsequent application of (S1) and (S2). For Equation (S1), the idea is that maximally splitting pebbles on existential states in $Q_2$ is exactly the same as moving the only pebble in $S^2(Q_2)$. More formally, one can show how to maintain the following invariant: $(q_1, q_2)$ is the current configuration in the $(k_1, n_2)$-game on the left iff $(q_1, q_2)$ is the current configuration in the $(k_1, 1)$-game on the right. From the invariant, the winning condition is easily verified: If $q_1 \subseteq F$, then $q_2 \cap F \neq \emptyset$, which is the same as saying $q_2 \in F'$, where $F'$ is the set of accepting states in $S^2(Q_2)$. Equation (S2) is proved similarly. \[ \square \]

Proof (of Lemma 1). We first prove Point 1), i.e., $q \sqsubseteq_{(k_1, k_2)} s$ implies $L^{\text{fin}}(q) \subseteq L^{\text{fin}}(s)$. Let $Q_1$ and $Q_2$ be two disjoint copies of $Q$, where the initial states are, respectively, $q$ and $s$. By definition, $q \sqsubseteq_{(k_1, k_2)} s$ iff $q \sqsubseteq_{(k_1, k_2)} Q_2$. Since $Q_1 \sqsubseteq_{(k_1, k_2)} Q_2$ implies $Q_1 \sqsubseteq_{(n_1, n_2)} Q_2$, then by Equation (S3), one has $S^v(Q_1) \sqsubseteq_{(1, 1)} S^2(Q_2)$. But $\sqsubseteq_{(n_1, n_2)}$ is known to imply language inclusion [4], hence $L^{\text{fin}}(S^v(Q_1)) \subseteq L^{\text{fin}}(S^2(Q_2))$. Finally, since the subset constructions are language-preserving (this follows from their correctness), $L^{\text{fin}}(Q_1) \subseteq L^{\text{fin}}(Q_2)$, implying $L^{\text{fin}}(q) \subseteq L^{\text{fin}}(s)$.

For Point 3), the crucial observation is that $S^v(Q_1) \sqsubseteq_{(1, 1)} S^2(Q_2)$ is equivalent to language inclusion, since $S^v(Q_1)$ is a purely existential automaton and $S^2(Q_2)$ is a purely universal automaton, hence only Spoiler plays. Thus, $L^{\text{fin}}(q) \subseteq L^{\text{fin}}(s)$ implies $S^v(Q_1) \sqsubseteq_{(1, 1)} S^2(Q_2)$, and, by Equation (S3), $Q_1 \sqsubseteq_{(n, n)} Q_2$, i.e., $q \sqsubseteq_{(n, n)} s$. \[ \square \]

C Section 4

Theorem 4. For $x \in \{\forall \text{di}, \text{de, f}\}$, automaton $Q$, $k_1, k_2 \in \mathbb{N}_{>0}$ and states $q$, $s \in Q$,

$$q \sqsubseteq_{(k_1, k_2)} s \quad \text{implies} \quad L^x(q) \subseteq L^x(s).$$

Proof. It suffices to prove the claim for $k_1 = k_2 = n = |Q|$, since, by Theorem 1, $(n, n)$-simulation contains $(k_1, k_2)$-simulation. Similarly, by the containment between universal-direct, delayed and fair simulation established in Theorem 2, it is sufficient to consider just fair simulation, which is the coarsest. Let $w = a_0a_1 \cdots \in L^x(q)$ be a word in $L^x(q)$. We have to show $w \in L^x(s)$. Let $G^{\text{sim}} = G^{(n, n)}(q, s)$ be the simulation game between $q$ and $s$, let $G^{\text{acc}} = G^x(q, w)$ be the acceptance game for $w$ from state $q$, and let $G^{\text{acc}} = G^x(s, w)$ be the acceptance game for $w$ from state $s$.

In order to show $w \in L^x(s)$, we use the winning strategy of Duplicator in $G^{\text{sim}}$ and the information in $G^{\text{acc}}$ to witness the existence of a winning strategy for Automaton in $G^{\text{acc}}$. We use the so-called “puppeteering technique” to coordinate the various games. There are two “real players”, the Automaton ($A_1$) and Pathfinder ($P_1$) players in $G^{\text{acc}}$, and four “puppet players”, which are controlled by $A_1$:

- The Duplicator ($D$) and Spoiler ($S$) puppets in the simulation game $G^{\text{sim}}$.
- The Automaton ($A_0$) and Pathfinder ($P_0$) puppets in the acceptance game $G^{\text{acc}}$.

The orchestration job of $A_1$ is complicated by the fact that in the simulation game $G^{\text{sim}}$ multiple Left and Right pebbles may be present in any given round. Henceforth, $A_1$ maintains a family of acceptance games, depending on the current configuration of
the simulation game. The flow of information between the various acceptance games and the simulation game is shown below.

The meaning of the picture is the following. Recall that in $G^{\text{sim}}$, Duplicator is allowed to “hedge her bets”, i.e., to split pebbles, on universal Left states and on existential Right states. Every time $D$ splits Left universal pebbles, a new acceptance game in $G^0_{\text{acc}}$ is spawned, and $D$’s choices in $G^{\text{sim}}$ are mimicked by $P_0$ in $G^0_{\text{acc}}$. Similarly, when $D$ splits Right existential pebbles, a new acceptance game in $G^1_{\text{acc}}$ is spawned, and $D$’s choice in $G^{\text{sim}}$ is mimicked by $A_1$ in $G^1_{\text{acc}}$. This is represented by the r.h.s. of the figure above. Symmetrically, the other side represents $S$’s behaviour, which moves Left existential pebbles copying $A_0$, and Right universal pebbles copying $P_1$.

The correctness is guaranteed by the fact that, since $A_0$ is playing a winning strategy in every $G^0_{\text{acc}}$ game, then regardless of universal choices in $G^0_{\text{acc}}$, the resulting run will visit accepting states infinitely often. Thus, by construction, Left pebbles in $G^{\text{sim}}$ are universally good infinitely often, and, since $D$ is playing a winning strategy for fair simulation, Right pebbles are existentially good infinitely often. We won’t explicitly define $A_1$’s winning strategy for accepting $w$ in $G^1_{\text{acc}}$, but, using Lemma 9 (which relies on König’s Lemma) we will show that one such strategy does exist: Indeed, since Right pebbles are existentially good infinitely often, by Lemma 9, there exists an accepting run for $w$ in $G^1_{\text{acc}}$, which witnesses the existence of a winning strategy for $A_1$.

For bookkeeping the state of the various simulation games, we use a logbook. Assume that, at round $i$, the current partial play in $G^{\text{sim}}$ is

$$\pi_i = \langle q_0, s_0 \rangle \ldots \langle q_i, s_i \rangle,$$

with $q_0 = \{q\}$ and $s_0 = \{s\}$, and that the remaining input word to be read is $w_i = a_i a_{i+1} \ldots$. Then, a logbook $L_i = (L^0_i, L^1_i)$ for round $i$ is a pair of finite sets of partial plays from $G^0_{\text{acc}}$ and $G^1_{\text{acc}}$, respectively, where $|L^0_i| = j^0_i$, $|L^1_i| = j^1_i$ and

$$L^0_i = \{\pi^0_{i,j} := \langle q_0, w_0 \rangle \ldots \langle q_{j-1}, w_{j-1} \rangle \mid 1 \leq j \leq j^0_i\}, \text{ and}$$

$$L^1_i = \{\pi^1_{i,j} := \langle s_0, w_0 \rangle \ldots \langle s_{j-1}, w_{j-1} \rangle \mid 1 \leq j \leq j^1_i\}$$

(with $(r, w_i)$ we mean that in the language acceptance game the current state is $r$ and the remaining input word is $w_i$). We say that $L_i$ is valid if it further satisfies the following logbook properties:

$$\forall \pi^0_{i,j} \in L^0_i : \pi^0_{i,j} \text{ is a } A_0\text{-conform partial play. (LP2)}$$
i.e., (LP0) and (LP1) say that the logbook is correctly “synchronized” with the simulation game, and (LP2) says that every \( \pi^0_{i,j} \) is built applying \( A_0 \)'s winning strategy.

Inductively, \( A_1 \) ensures that a valid logbook at round \( i \) is updated into a valid logbook in the next round \( i + 1 \). In the first round \( i = 0 \), the initial \( G \text{sim} \)-configuration is \( \langle q_0, s_0 \rangle = (\{q\}, \{s\}) \), and the two acceptance games \( G^\text{acc}_0 \) and \( G^\text{acc}_1 \) are in \( \langle q, w \rangle \) and \( \langle s, w \rangle \), respectively. Clearly, \( L_0 = (L'_0, L''_0) \) with \( L'_0 = \{\langle q, w \rangle \} \) and \( L''_0 = \{\langle s, w \rangle \} \) is a valid logbook.

Assume that \( L_i \) is a valid logbook for round \( i \), that the current configuration of the simulation game is \( \langle q_i, s_i \rangle \), and that these two sets are partitioned into existential and universal states, in the following way:

\[
q_i = \{q_{i,1}^E, \ldots, q_{i,j}^E \} \cup \{q_{i,1}^U, \ldots, q_{i,j}^U \} \quad \text{and} \quad s_i = \{s_{i,1}^E, \ldots, s_{i,j}^E \} \cup \{s_{i,1}^U, \ldots, s_{i,j}^U \}
\]

(Notice that \( j^E_0 + j^U_0 = j^E_1 \) and \( j^E_0 + j^E_1 = j^U_1 \).) The next input symbol in \( G \text{sim} \) is determined by the remaining input word \( w_i = a_1a_{i+1} \ldots \), and it is equal to \( a_i \). I.e., \( A_1 \) makes the \( S \) puppet choose \( a_i \) as the next input symbol in the simulation game \( G \text{sim} \), and the remaining input for the next round is \( w_{i+1} = a_{i+1}a_{i+2} \ldots \). For determining the next \( G \text{sim} \)-configuration \( \langle q_{i+1}, s_{i+1} \rangle \), as well as updating the logbook to \( L_{i+1} = (L'_i, L''_i, L_1') \), \( A_1 \) orchestrates the various puppets according to the following steps.

1. The \( A_0 \) puppet moves from existential \( G^\text{acc}_0 \)-configurations \( \langle q_{i,1}^E, w^i \rangle \ldots \langle q_{i,j}^E, w^j \rangle \) according to her winning strategy in \( G^\text{acc}_0 \). Notice that all such pebbles in \( q_i^E \) can be moved to some \( a_i \)-successor, since 1) \( w \in L^\omega(q) \), and 2) different Left states are the result of different past choices at universal \( L \) states, hence all states \( q_{i,j}^E \in q_i^E \) accept \( w_i \) (in fact, all states in \( q_i \)). In this way, an \( a_i \)-successor \( q_{i+1,j} \) is built for every \( q_{i,j}^E \in q_i^E \). Let \( q_{i+1,j}^E = \{q_{i,j+1}^E \mid 1 \leq j \leq j^E_0 \} \) be the set of such \( a_i \)-successors.

2. Similarly, the \( P_1 \) puppet chooses a successor for every universal \( G^\text{acc}_1 \)-configurations \( \langle s_{i,1}^U, w^i \rangle \ldots \langle s_{i,j}^U, w^j \rangle \). Let \( q_{i+1,j}^U = \{s_{i,j+1}^U \mid 1 \leq j \leq j^U_0 \} \) be the set of such \( a_i \)-successors, if any. Notice that, at this point, one or even all such configurations may get stuck, i.e., with no \( a_i \)-successor, and, consequently, \( q_{i+1}^U \) might be empty. We need not worry about this now.

3. Then, the \( S \) puppet in the simulation game \( G \text{sim} \) copies the \( A_0 \)'s and \( P_1 \)'s moves above, moving from \( q_i^E \) to \( q_{i+1}^E \) and from \( s_i^U \) to \( s_{i+1}^U \).

4. Now, the \( D \) puppet moves according to her winning strategy: Pebbles in \( q_i^U \) are moved to \( q_{i+1}^U \) (possibly being split for \( D \) may hedge her bets), and pebbles in \( s_i^E \) are moved to \( s_{i+1}^E \) (possibly being split or thrown away), and the simulation game goes into state \( \langle q_{i+1}, s_{i+1} \rangle \), where \( q_{i+1} = q_{i+1}^E \cup q_{i+1}^U \), and \( s_{i+1} \subseteq s_{i+1}^E \cup s_{i+1}^U \).
Since $D$’s strategy is winning, at least one Right pebble can be moved to some successor. Thus, $s_{i+1} \neq \emptyset$, $D$’s move above is copied by the puppet $P_0$ and the player $A_1$, as specified below.

5. The $P_0$ puppet copies the $D$ puppet’s move from $q_i^U$ to $q_i^{(U)}$: For any $q_i^U, q_i^{(U)} \in q_i^U \cap \Delta_i$, we add $\pi_{i+1,j} = \pi_i^0 \cdot \langle q_i^{(U)} \rangle_{i+1,j}$ to $L_{i+1}^0$, where $\pi_{i,j}$ is the partial play ending in $\langle q_i^{(U)}, w_i \rangle$, which exists by (LP0). This establishes (LP0) and (LP2) for universal states in $q_i$.

6. Finally, the $A_1$ player in the family of acceptance games $\Gamma^\omega_{1}$ copies the $D$ puppet’s move from $s_{i}^E$ to $s_{i+1}^{(E)}$: For any $s_i^E \in s_i^E \cap \Delta_i$, $s_{i+1,j} \in s_{i+1}^{(E)}$, we add $\pi_{i+1,j} = \pi_i^1 \cdot \langle s_{i+1,j}, w_{i+1} \rangle$ to $L_{i+1}^1$, where $\pi_{i,j}$ is the partial play ending in $\langle s_{i,j}, w_i \rangle$, which exists by (LP1). We also update the logbook for universal states in $s_i$ which were not discarded in the previous step 2.: For any $s_i^U \in s_i^U \cap \Delta_i$, $s_{i+1,j} \in s_{i+1}^{(U)}$, we add $\pi_{i+1,j} = \pi_i^0 \cdot \langle s_{i+1,j}, w_{i+1} \rangle$ to $L_{i+1}^2$, where $\pi_{i,j}$ is the partial play ending in $\langle s_{i,j}, w_i \rangle$, which exists by (LP1). This establishes (LP1) for the next round $i + 1$. (Notice that, since $s_{i+1} \neq \emptyset$, then $L_{i+1}^1$ contains at least one partial play.)

Below we argue about the correctness of this construction. Let $\pi = \langle q_0, s_0 \rangle\langle q_1, s_1 \rangle \ldots$ be the resulting infinite play, and let $\pi^0 = q_0 q_1 \ldots$ and $\pi^1 = s_0 s_1 \ldots$ be its projections. Since every partial play in $L_i^0$ is $\mathcal{A}_0$-conform (by (LP2)), and the $A_0$ puppet is playing according to a winning strategy (which exists, since $w \in \mathcal{L}_5(q)$), it follows that, for every $q_i \in \mathcal{L}_5(q)$, there exists $\pi_i^0 \in L_i^0$ has visited at least $n$ accepting states. By (LP0) and since no Left pebble is thrown away, we have that $\text{good}^a(\pi^0, \omega)$.

Since $D$’s strategy is winning, by the winning condition of fair simulation, we have $\text{good}^a(\pi^1, \omega)$. By Lemma 9, there exists an infinite accepting path $\pi_{\text{acc}}$. Thus, $A_1$ has a strategy $s_i$ for any $P_i$’s strategy, there exists an accepting path $\pi_{\text{acc}}$ which is conform to both strategies. Thus, $w \in \mathcal{L}_5(s_i)$.

**Theorem 5.** For any $x \in \{\forall, \exists, \text{d}, \text{e}, \text{f}\}$, there exist an automaton $\mathcal{Q}$ and states $q_0, s_0 \in \mathcal{Q}$ s.t. $\mathcal{L}_5(q_0) \subseteq \mathcal{L}_5(s_0)$, but $q_0 \not\mathcal{L}_5(n,n) \not\subseteq \mathcal{L}_5(s_0)$.

**Proof.** By the inclusion between simulations (see Theorem 2), it is enough to consider fair simulation. Take the example in Figure 2(b). We have that $\mathcal{L}_5(q_0) = \mathcal{L}_5(s_0) = \alpha^\omega + \alpha \beta^\omega$, hence language inclusion holds between $q_0$ and $s_0$, but, as we shall see, no Duplicator strategy is winning in the $(n,n)$-simulation game $\Gamma^\omega_{(n,n)}(q_0, s_0)$.

This can be seen as follows. Spoiler chooses the $a$ action and we can assume that Duplicator “hedges her bets” by going in $s' = \{s_0, s_1\}$. Now, Spoiler keeps looping on $q_0$ by choosing the $a$ action for an arbitrarily high number of moves. Duplicator can only reply by staying in a subset of $s'$. Notice that Duplicator has to eventually take the pebble on $s_0$ away: Indeed, Spoiler’s pebble in $q_0$ is accepting infinitely often, hence Duplicator would lose if the pebble on $s_0 \not\mathcal{F}$ is not taken away. When Duplicator takes the pebble on $s_0$ away, Spoiler plays the $b$ action and Duplicator loses, his remaining pebble on $s_1$ being stuck. Similar examples may be conceived in which the acceptance condition, instead of the stickiness condition, is used to show that Duplicator loses. □
C.1 Infinite words: ABAs and the Miyano-Hayashi construction

Definition 3. Given an ABA \( Q = (Q, \Sigma, q_I, \Delta, F, E, U) \), the Miyano-Hayashi construction [10] yields a de-universalized NBA

\[ Q_{nd} := (Q', \Sigma, \{(q_I), \{q_I\} \setminus F\}, \Delta', F', E', U') , \]

where the new set of states \( Q' \subseteq 2^Q \times 2^Q \) (called macrostates) consists of pairs of subsets of \( Q \), the set of accepting macrostates \( F' \) satisfies

\[ (P,O) \in F' \iff O = \emptyset, \]

i.e., a macrostate is accepting if no obligation is pending, \( E' = Q' \) and \( U' = \emptyset \), i.e., \( Q_{nd} \) is a purely non-deterministic automaton, and the the transition relation \( \Delta' \subseteq (2^Q \times 2^Q) \times \Sigma \times (2^Q \times 2^Q) \) satisfies: \( ((P,O), a, (P',O')) \in \Delta' \) if there exists a choice function \( \text{select} : P \cap E \times \Sigma \to Q \) s.t. \( \forall p \in P \cap E, a \in \Sigma : (p, a, \text{select}(p, a)) \in \Delta \) (i.e., \( \text{select}(p, a) \) fixes an element in \( \Delta(p, a) \)) such that

\[ P' = \bigcup_{p \in P \cap U} \Delta(p, a) \cup \{\text{select}(p, a) \mid p \in P \cap E\} , \]

and if \( O = \emptyset \), then \( O' = P' \setminus F \), otherwise,

\[ O' = \left( \bigcup_{o \in O \cap U} \Delta(o, a) \cup \{\text{select}(o, a) \mid p \in O \cap E\} \right) \setminus F . \]

Lemma 3. For any ABA \( Q \), let \( Q_{nd} \) be the NBA obtained according to the Miyano-Hayashi de-universalization procedure applied to \( Q \). Then,

a) \( Q \leq^1_{(1,1)} Q_{nd} \) for \( x \in \{f, \forall \} \), and
b) \( Q_{nd} \leq^1_{(1,1)} Q \).

Moreover, there exist automata \( Q^1 \) and \( Q^2 \) s.t.

a') \( Q^1 \leq^{de}_{(n,1)} Q_{nd} \) and
b') \( Q^2 \leq^{de}_{(1,1)} Q_{nd} \), for \( x \in \{de, \forall \} \).

Proof. We first prove Point a), i.e., \( Q \leq^1_{(1,1)} Q_{nd} \). Intuitively, the strategy of Duplicator is to maximally hedge her bets on \( Q \) (i.e., Left universal pebbles), and to select successors in \( Q_{nd} \) (which is a purely existential automaton) by copying Spoiler’s moves from Left existential pebbles in \( Q \). More formally, there exists a strategy for Duplicator which maintains the following invariant: If at round \( k \) the current configuration is \( \langle q_k, (q'_k, o_k) \rangle \), then \( q_k = q'_k \), i.e., Duplicator has a strategy that mimicks exactly the MH-construction.

We now argue that this strategy is winning for Duplicator. If some Left pebble gets stuck, then Duplicator wins. Otherwise, by the properties of the MH-construction, it is the case that the Right pebble can always be moved; in this case, we argue as follows. Let \( \pi = (q_0, (q_0, o_0))(q_1, (q_1, o_1)) \ldots \) be the resulting sequence of configurations.
For $x = \forall \ell_i$, assume that $q_k \subseteq F$. By the definition of the MH-construction $o_k \subseteq q_k \cap F$, hence $o_k = \emptyset$, i.e., $(q_k, \emptyset) \in F'$. For $x = f$, assume that there are infinitely many $i$'s s.t. $q_i$ is universally good since some previous round $j_i$. Now consider the sequence of indices $\{k_i\}_{i \geq 0}$ defined as follows: $k_0 = 0$, and, inductively, $k_{i+1}$ is s.t. $q_{k_{i+1}}$ is good since round $k_i$. (Notice that this sequence is well-defined and infinite: Since there are infinitely many $i$'s s.t. $q_i$ is universally good since some previous round $j_i$, by Lemma 8, this implies that for any $i$, there exists $j_i \geq i$ s.t. $q_j$ is universally good since round $i$.) We have that, by the definition of the MH-construction, $o_{k_i} = \emptyset$ for any $i > 0$. Hence, $(q_{k_i}, \emptyset) \in F''$ for infinitely many $k_i$'s.

We now prove Point b), i.e., $Q_{nd} \subseteq_{\{1, 1\}} Q$. We can assume that the Left pebble never gets stuck, otherwise Duplicator wins trivially. Here, the strategy for Duplicator is to maintain the following invariant: If at round $k$ the current configuration is $\langle (q_k, o_k), q_k \rangle$, then $q_k \in q_k$, i.e., Duplicator can force the Right pebble to be somewhere in $q_k$. Clearly, the invariant holds for the initial configuration: For $i = 0$, $(q_0, o_0) = \langle \{q_f\}, \{q_f\} \setminus F \rangle$ and $q_0 = q_f$. Inductively, assume that the invariant $q_i \in q_i$ holds for $k \geq 0$. We show how Duplicator can ensure it in the next round $k + 1$. Assume that Spoiler moves the Left pebble to $(q_{k+1}, o_{k+1})$ and that the next input symbol is $a_{k+1}$. We have two cases to consider:

- If $q_k \in E$, then, by the MH-construction, there exists $q' \in q_{k+1}$ s.t. $q' \in \Delta(q_k, a_k)$.
  In this case, Duplicator moves the Right pebble from $q_k$ to $q_{k+1} := q'$.
- If $q_k \in U$, then, by the MH-construction, it is the case that $\Delta(q_k, a_k) \subseteq q_{k+1}$.

Hence, Spoiler moves the Right universal pebble $q_k$ to any successor $q_{k+1} \in \Delta(q_k, a_k)$. For every Spoiler’s move, $q_{k+1} \in q_{k+1}$.

We now argue that this invariant-preserving strategy is winning for Duplicator. Assume that, for infinitely many $i$'s, $(q_i, o_i)$ is accepting, i.e., $o_i = \emptyset$. Hence, we can build an infinite sequence $\{i_j\}_{j \geq 0}$ of indices s.t. $o_{i_j} = \emptyset$ for any $j > 0$. From the structure of the MH-construction, it follows that for any $j \geq 0$, there exists $k$ s.t. $i_{j-1} < k \leq i_j$ and $q_k \in F$. Hence, $q_k$ is accepting for infinitely many indices $k$’s.

Points a’) and b’) are actually shown in Figure 3, where we give two automata $Q^1$ and $Q^2$ which result in the same de-universalized automaton, i.e., $Q^1_{nd} = Q^2_{nd} =: Q_{nd}$. For Point a’), the simulation game $G^{\text{sim}}_{\{n, 1\}}(Q^1, Q^2_{nd})$ results in the (unique) sequence of configurations

$$\pi = \langle (q_0), (q_0), (q_0) \rangle \langle (q_1, q_{12}), (q_1, q_{12}) \rangle \langle (q_2, q_{22}), (q_{22}, q_{22}) \rangle \langle (q_{31}, q_{32}), (q_{31}, q_{32}) \rangle \omega,$$

but at round $k = 3$, the Left pebbles on $q_{31}, q_{32}$ are universally good since round 2 (since $q_2$ and $q_3$ are in $F$), but the Right pebble is never accepting for $k \geq 3$. For Point b’), the reasoning is similar, but now $q_{31} \not\in F$. We have that Spoiler can force the game $G^{\text{sim}}_{\{n, 1\}}(Q^1_{nd}, Q^2_{nd})$ in the following sequence of configurations:

$$\pi = \langle s_0, q_0 \rangle \langle s_1, q_{11} \rangle \langle s_2, q_{21} \rangle \langle s_3, q'_{31} \rangle \omega,$$

s.t., at round $k = 2$, $s_2 \in F'$, but 1) $q_{21} \not\in F$ (hence, $Q^2_{nd} \not\subseteq_{\{1, 1\}} Q^2$ for $x = \forall \ell_i$), and 2) for no later round $k' \geq 2$, the Right pebble is accepting (in fact, it is trapped in $q'_{31} \not\in F$), hence $Q^2_{nd} \not\subseteq_{\{1, 1\}} Q^2$ for $x = \text{de}$. \qed
Lemma 4. Let $Q$, $S$ be two alternating Büchi automata. Then,

$$Q \preceq_{(n,1)} S \iff Q_{ad} \preceq_{(1,1)} S_{ad}.$$  

Proof. We make use of transitivity, which will established later.

“Only if”. Assume $Q \preceq_{(n,1)} S$. Then, by a double application of Lemma 3,

$$Q_{ad} \preceq_{(1,1)} Q \preceq_{(n,1)} S \preceq_{(n,1)} S_{ad},$$  

and, since $(1,1)$-simulation is contained in $(n,1)$-simulation (see Theorem 1), by transitivity, we obtain $Q_{ad} \preceq_{(1,1)} S_{ad}$. But $Q_{ad}$ is a purely existential automaton, hence $(n,1)$-simulation reduces to $(1,1)$-simulation in this case. Thus, $Q_{ad} \preceq_{(1,1)} S_{ad}$.

“If”. Assume $Q_{ad} \preceq_{(1,1)} S_{ad}$. By Lemma 3,

$$Q \preceq_{(n,1)} Q_{ad} \preceq_{(1,1)} S_{ad} \preceq_{(1,1)} S,$$

thus, by “upgrading” $(1,1)$-simulation to $(n,1)$, and by transitivity, $Q \preceq_{(n,1)} S$. □

C.2 Transitivity

Theorem 6. Let $x \in \{\forall, \exists, \delta, \epsilon\}$. Then,

$$q \preceq_{(1,n)}^x r \land r \preceq_{(n,1)}^x s \implies q \preceq_{(1,n)}^x s$$

$$q \preceq_{(n,1)}^x r \land r \preceq_{(n,1)}^x s \implies q \preceq_{(n,1)}^x s.$$  

Proof. Directly from Lemma 10 and Lemma 11 below. □

Definition of the join of two strategies for $(1, n)$ simulations. Let $G_0 = G^{(1,n)}(q, r)$ and $G_1 = G^{(1,n)}_1(r, s)$ be the basic simulation games between $q$ and $r$, and between $r$ and $s$, respectively. Let $\sigma_0$ and $\sigma_1$ be two Duplicator’s strategies in $G_0$ and $G_1$, respectively. We construct a joint strategy $\sigma_0 \bowtie \sigma_1$ for Duplicator in the basic simulation game $G = G^{(1,n)}(q, s)$. In the definition of the join, we assume that the automaton is complete, hence it is always possible to select successors and the simulation game never halts.

We keep track of the current state of the $G_0$ game and at most $n$ games in $G_1$. In the picture above, $G_0$ is shown in the center, where games in $G_1$ are shown at the top/bottom of the picture, where, for $X \in \{E, U\}$, with $G^{X, (1,n)}_1(q, s)$ we mean those games in $G_1$ where the (only) Left pebble is existential/universal. An arrow of the form $P_{X,Y} \rightarrow P_{X,Y'}$,
means that $X$-pebbles on side $Y$ moved by player $P$ in the source game induce a move by player $P'$ in the destination game on side $Y'$, where $P, P' \in \{S, D, 0S, 0D, \overset{\circ}{S}, \overset{\circ}{D}\}$, $X \in \{E, U\}$ and $Y, Y' \in \{L(\text{left}), R(\text{right})\}$. (Notice that the kind of pebbles $X$ does not change across an arrow.)

The necessary bookkeeping is done by using a logbook. At round $k \geq 0$, the current logbook is a triple $L_k = (L^0_k, L^1_k, w_k)$, where $w_k = a_0a_1 \ldots a_{k-1}$ is the input word constructed so far (and we let $w_0 = \varepsilon$ if $k = 0$), $L^0_k = \pi^0_k$, with $\pi^0_k = \langle q^0_k, r^0_k, s^0_k \rangle \ldots \langle q^0_{k+1}, r^0_{k+1}, s^0_{k+1} \rangle$, is a partial play in $G_0$ of length $k + 1$, which is $\sigma_0$-conform to $w_k$, and $L^1_k = \{\pi^{-1}_{k,0}, \pi^{-1}_{k,1}, \ldots, \pi^{-1}_{k,l(k)}\}$ is a set of partial plays in $G_1$, which are $\sigma_1$-conform to $w_k$, of length $k + 1$, and of the form

$$\pi^{-1}_{k,i} = \langle r^0_{k,i}, s^0_{k,i} \rangle \langle r^1_{k,i}, s^1_{k,i} \rangle \cdots \langle r^1_{k,i}, s^1_{k,i} \rangle, \text{ for } i \in 0, \ldots, l(k).$$

Assume that at round $k \geq 0$ the current partial play in $G$ is $\pi_k$, where

$$\pi_k = \langle q_0, s_0 \rangle \langle q_1, s_1 \rangle \cdots \langle q_k, s_k \rangle.$$  \hfill (*)

We say that a logbook $L_k$ is valid if it satisfies the logbook properties below.

\begin{align*}
q^0_k &= q_k \quad \text{(P0)} \\
r^0_k &= \bigcup_{i(k)} \{r^1_{k,i}\} \quad \text{(P1)} \\
s_k &= \bigcup_{i(k)} s^1_{k,i} \quad \text{(P2)}
\end{align*}

Notice that (P1) entails the following property: (P1') For every $r \in r^0_k$, there exists $i$ s.t. $\pi^{-1}_{k,i} = \langle r^0_{k,i}, s^0_{k,i} \rangle \cdots \langle r^1_{k,i}, s^1_{k,i} \rangle$ with $r = r^1_{k,i}$, i.e., $r$ is the r.h.s. of the last configuration of some partial play in $L^1_k$.

We inductively show how to build a valid logbook and we define the joint strategy $\sigma_0 \bowtie \sigma_1$. The initial configuration in $G_0$ is $\langle q, r \rangle$, the one in $G_1$ is $\langle r, s \rangle$, and the one in $G$ is $\langle q, s \rangle$. Hence, the initial logbook $L_0 := (L^0_0, L^1_0)$, with $L^0_0 = \langle q^0_0, r^0_0 \rangle$ and $L^1_0 = \{\langle r^1_0, s^1_0 \rangle\}$, is clearly valid, where $q^0_0 = q$, $r^0_0 = r$, and $s^0_0 = s$.

Inductively assume that, at round $k$, $L_k$ is a valid logbook, and that the current (partial) play in $G$ is $\pi_k$, with $\pi_k = \langle q_0, s_0 \rangle \langle q_1, s_1 \rangle \cdots \langle q_k, s_k \rangle$ (as in (*)). We show how to build a new logbook $L_{k+1} = (L^0_{k+1}, L^1_{k+1}, w_{k+1})$ for the next round, and we prove it valid. Assume that Spoiler moves as follows:

$$(\{q_k\}, s_k, a_k, (q_k)', (s_k)') \in I^{Sp}_{G},$$  \hfill (S)

i.e., the next input symbol is $a_k$ and Spoiler moves universal-r.h.s. pebbles from $s_k$ to $(s_k)'$. Notice that, if $q_k$ is existential, then Spoiler moves the only l.h.s. pebble from $q_k := \{q_k\}$ to $(q_k)' := \{(q_k)\}'$, otherwise $(q_k)' := \emptyset$. Notice that we can already update the next input word to $w_{k+1} := w_k \cdot a_k$, which defines the third component of the next logbook.
For any universal state \(r_{k,i}^U \in r_k^0 \cap U\), let \(\pi_{k,i}^1\) be the path in \(L_k^1\) s.t. \(\pi_{k,i}^1 = \langle r_{0,i}^1, s_{0,i}^1 \rangle^1 \cdots \langle r_{k,i}^1, s_{k,i}^1 \rangle^1\) with \(r_{k,i}^1 = r_{k,i}^U\), which is guaranteed to exist by property (P1'). Spoiler’s move (S) above induces the \(G_1\)-Spoiler’s move from \(\langle r_{k,i}^1, s_{k,i}^1 \rangle^1\) below:

\[
((r_{k,i}^1), s_{k,i}^1, a_k, \{s_{k,i}^1\}) \in \Gamma_{G_1}^{S_0},
\]

i.e., \(a_k\) is fixed by move (S) above, and \(\{s_{k,i}^1\}\) is the subset of \(s_{k,i}^1\) obtained by restricting Spoiler’s move’s (S) to \(s_{k,i}^1 \subseteq s_k\), the containment following by property (P2). We now apply \(G_1\)-Duplicator’s strategy \(\sigma_1\), obtaining

\[
\sigma_1((r_{k,i}^1), s_{k,i}^1, (r_{k,i}^1), s_{k,i}^1) = ((r_{k,i}^1), s_{k,i}^1)^{\prime \prime}. \tag{1-D}\]

By the completeness condition, it is always possible to select some successor \((r_{k,i}^1), s_{k,i}^1)^{\prime \prime}\) and \((s_{k,i}^1)^{\prime \prime} \neq \emptyset\).

The move (1-D) above fixes a successor \((r_{k,i}^1), s_{k,i}^1)^{\prime \prime}\) for each universal state \(r_{k,i}^U \in r_k^0\).

We now consider these moves as adversarial in \(G_0\), i.e., they induce a \(G_0\)-Spoiler’s move

\[
((q_0^0), q_0^0, a_k, (q_0^0), (r_0^0)) \in \Gamma_{G_0}^{S_0},
\]

where \((r_0^0)\) is the set of elements \((r_{k,i}^1)\) above. We then apply \(G_0\)-Duplicator’s strategy \(\sigma_0\), obtaining

\[
\sigma_0((q_0^0), q_0^0, (q_0^0), (r_0^0)) = (q_{k+1}^0, r_{k+1}^0). \tag{0-D}
\]

Notice that, if \(q_0^0\) is existential, then \(\{q_0^0, q_{k+1}^0\} = (q_0^0)\) as determined in (S), otherwise \(q_{k+1}^0\) is determined by (0-D) above. The new configuration of the game \(G_0\) is \((q_{k+1}^0, r_{k+1}^0)\), and, accordingly, the first component \(L_{k+1}^0\) of the new logbook is defined as \(L_{k+1}^0 := \pi_{k+1}^0 \cdot \langle q_{k+1}^0, r_{k+1}^0 \rangle\). By the completeness condition, it is always possible to select some successor \(q_{k+1}^0\) and \(r_{k+1}^0 \neq \emptyset\).

The \(G_0\)-Duplicator’s move (0-D) above fixes a successor \((r_{k,i}^E)\) for any existential \(r_{k,i}^E \in r_k^0 \cap E\) By the logbook property (P1’), for any such \(r_{k,i}^E\), there exists a path \(\pi_{k,i}^1\) in \(L_k^1\) s.t. \(\pi_{k,i}^1 = \langle r_{0,i}^1, s_{0,i}^1 \rangle^1 \cdots \langle r_{k,i}^1, s_{k,i}^1 \rangle^1\). The \(G_0\)-Duplicator’s move (0-D) above is interpreted adversarially in \(G_1\):

\[
((r_{k,i}^E), s_{k,i}^1, a_k, (r_{k,i}^E), (s_{k,i}^1)) \in \Gamma_{G_1}^{S_0}. \tag{1-S'}
\]

We then apply \(G_1\)-Duplicator’s winning strategy \(\sigma_1\):

\[
\sigma_1((r_{0,i}^1, s_{0,i}^1) \cdots (r_{k,i}^1, s_{k,i}^1)) = ((r_{k,i}^E), (s_{k,i}^1)^{\prime \prime}). \tag{1-D}'
\]

Once again, the completeness condition entails that it is always possible to select some successor \((r_{k,i}^E), (s_{k,i}^1)^{\prime \prime}\) and \((s_{k,i}^1)^{\prime \prime} \neq \emptyset\).

We are now ready to define \(L_{k+1}^1\). Let \((r_{k,i}^1)\) be any state \((r_{k,i}^1)\) defined in (0-D), or any state \((r_{k,i}^E)\) defined in (1-D'), which is not discarded in (0-D), i.e., \((r_{k,i}^1) \notin r_{k+1}^0\).

(Notice that \((r_{k,i}^1) \neq \emptyset\) by the completeness property, as already noticed above. Hence, there exists at least one such \((r_{k,i}^1)\). We then add \(\pi_{k,i}^1 \cdot \langle (r_{k,i}^1), (s_{k,i}^1)^{\prime \prime}\) to the second
component \( L_{k+1}^1 \) of the new logbook. It is easy to check that, by construction, property (P1) holds at round \( k + 1 \).

Finally, we define Duplicator’s move in \( G \) as

\[
(\{q_k, s_k, a_k, (q'_k, s'_k)\}, \{q_{k+1}, s_{k+1}\}) \in I_{G}^{\text{Dup}},
\]

where \( q_{k+1} := q^0_{k+1} \) is fixed by move (0-D) if \( q_k = q^0_k \) is existential, and it is fixed by (S) if it is universal. (Notice that this establishes property (P0).) Moreover, \( s_{k+1} \) is taken to be the union of all sets \((s_{k,i}^1)^n \) constructed in \( (1-D_i) \) and \( (1-D'_i) \) above, i.e.,

\[
s_{k+1} := \bigcup_i (s_{k,i}^1)^n, \text{ which, in turn, establishes property (P2). Hence,}
\]

\[
(\sigma_0 \triangleright \sigma_1) (\langle q_0, s_0 \rangle \cdots \langle q_k, s_k \rangle) (a_k, (q'_k, s'_k)) :=\langle q_{k+1}, s_{k+1} \rangle.
\]

The following theorem shows that \( \sqsubseteq_{(1,n)} \) is transitive, i.e., it shows that when \( \sigma_0 \) and \( \sigma_1 \) are both winning, then \( \sigma_0 \triangleright \sigma_1 \) is winning as well.

**Lemma 10.** Let \( x \in \{\forall i, d, e, f\} \). Then,

\[
q \sqsubseteq_{(1,n)} r \land r \sqsubseteq_{(1,n)} s \implies q \sqsubseteq_{(1,n)} s.
\]

**Proof.** We refer to the logbook \( L_k \) at round \( k \) as defined above. We first deal with the case in which the game \( G \) never ends prematurely.

For \( x = \forall i \), we have to show that, whenever \( q_k \) is accepting, so is every pebble in \( s_k \). Assume \( q_k \in F \). Since \( \sigma_0 \) is a winning strategy, then \( r^0_k \subseteq F \). By the logbook property (P1'), the current configuration of every game in \( G_1 \) is of the form \( \langle r, s \rangle \), for some \( r \in F \) and \( s \subseteq Q \). But \( \sigma_1 \) is winning, hence every such \( s \) is contained in \( F \). By (P2), \( s_k \subseteq F \).

For \( x = d \), assume that \( q_k \) is accepting. Since \( \sigma_0 \) is winning, there exists \( j \geq k \) s.t. \( \text{good}^2(r^0_j, k) \). Thus, for any \( r^1_{j,i} \in r^0_{j,i} \), there exists \( j(i) \) s.t. \( r^1_{j,i} \in F \). By (P1), let \( (r^1_{j(i)}, s^1_{j(i)}) \) be the configuration at round \( j(i) \) of some \( G_1 \)-game. Since \( \sigma_1 \) is winning, there exists \( j(i) \) s.t. \( \text{good}^2(s^1_{j(i)}, i) \). Let \( j^* = \max_j (j(i)) \). Then, for all \( i \), every state in \( s^1_{j^*} \) has seen an accepting state since round \( k \). By (P2), \( s^1_{j^*} = \bigcup_i s^1_{j^*} \). Thus, every state in \( s^1_{j^*} \) has seen an accepting state since round \( k \). Therefore, there exists a minimal \( k^* \) s.t. \( k \leq k^* \leq j^* \) and \( \text{good}^2(s^1_{k^*}, k) \).

For \( x = f \), the reasoning is entirely similar to the previous paragraph.

We now deal with the case in which the game \( G \) ends prematurely. If the Left pebble on \( q_k \) is stuck, then Duplicator wins, and we are done. Otherwise, assume that the Left pebble is never stuck. We show that, in this case, the game actually never stops. In fact, since \( \sigma_0 \) is a winning strategy in \( G_0 \), then there always exists some Right pebble \( r_{k,i} \in r_k \) which can proceed. By (P1'), there exists some configuration \( \langle r_{k,i}, s_{k,i} \rangle \) in \( G_1 \) which can go on, and, being \( \sigma_1 \) winning in such a game, then some pebble in \( s_{k,i} \) can be moved, and, therefore, some pebble in \( s_k \) can be moved as well (by (P2)). Thus, \( G \) never stops.

\( \square \)
Definition of the join of two strategies for \((n, 1)\) simulations. The definitions for \((1, n)\) simulations are dual to the \((1, n)\) case in the previous section. Let \(G_0 = G^{(n, 1)}(q, r)\) and \(G_1 = G^{(n, 1)}(r, s)\) be two basic simulation games. Let \(\sigma_0\) and \(\sigma_1\) be two Duplicator’s strategies in \(G_0\) and \(G_1\), respectively. We construct a joint strategy \(\sigma_0 \bowtie \sigma_1\) for Duplicator in the basic simulation game \(G = G^{(n, 1)}(q, s)\).

We keep track of the current state of the \(G_1\) game and (at most) \(n\) games in \(G_0\) using the logbook technique. At round \(k \geq 0\), the current logbook is a triple \(L_k = (L_0^k, L_1^k, w_k)\), where \(w_k = a_0a_1 \ldots a_{k-1}\) is the input word constructed so far, \(L_0^k = \pi^k_0 = (q_0, s_0^1)(r_1, s_1^1) \ldots (r_k, s_k^1)\) is a partial \(\sigma_1\)-conform to \(w_k\) play of length \(k + 1\) in \(G_1\), and \(L_1^k = \{\pi^0_{k,0}, \pi^0_{k,1}, \ldots, \pi^0_{k,l(k)}\}\) is a set of partial \(\sigma_0\)-conform (w.r.t. \(w_k\)) plays of length \(k + 1\) in \(G_0\) with \(\pi^0_{k,i} = (q_{0,i}^0, r_{0,i}^0)(q_{1,i}^0, r_{1,i}^0) \ldots (q_{k,i}^0, r_{k,i}^0)\), for \(i \in \{0, \ldots, l(k)\}\). Every logbook \(L_k\) will satisfy an invariant, which consists of the logbook properties (P0)-(P2) specified below. Assume that at round \(k \geq 0\) the current play in \(G\) is \(\pi^k\), where \(\pi^k = (q_0, s_0^0)(q_1^0, s_1^0) \ldots (q_k^0, s_k^0)\). Then, \(L_k\) is a valid logbook if

\[
\begin{align*}
q_k &= \bigcup_{i=0}^{l(k)} q_{k,i}^0 & \text{(P0)} \\
r_k &= \bigcup_{i=0}^{l(k)} r_{k,i}^0 & \text{(P1)} \\
s_k &= s_k^0 & \text{(P2)}
\end{align*}
\]

We inductively show how to build a valid logbook and how to define the joint strategy \(\sigma_0 \bowtie \sigma_1\). The initial configuration in \(G_0\) is \(\{q\}, \{r\}\) (there is only one such game initially), the one in \(G_1\) is \(\{\{r\}, \{s\}\}\), and the one in \(G\) is \(\{\{q\}, \{s\}\}\). Let \(q_0 = \{q\}\), \(r_0 = r\), and \(s_0 = s\). Hence, the initial logbook \(L_0 := (L_0^0, L_1^0, w_0)\), with \(L_0^0 = \{\{(q_0), \{r_0\}\}\}\), \(L_1^0 = \{\{\{r_0\}, \{s_0\}\}\}\), and \(w_0 = e\), is clearly valid.

Inductively assume that, at round \(k\), the current partial play in \(G\) is \(\pi^k\), and that \(L_k = (L_0^k, L_1^k, w_k)\) is a valid logbook, where \(L_0^k\), \(L_1^k\) and \(w_k\) are defined as above. By (P2),

\[
\begin{align*}
\pi^k &= (q_0, s_0^0)(q_1^0, s_1^0) \ldots (q_k^0, s_k^0) \\
\pi^k_{k,i} &= (q_{0,i}^0, r_{0,i}^0)(q_{1,i}^0, r_{1,i}^0) \ldots (q_{k,i}^0, r_{k,i}^0), \text{ for } i \in \{0, \ldots, l(k)\} \\
\pi^1_k &= (r_0, s_0^0)(r_1, s_1^0) \ldots (r_k, s_k^0)
\end{align*}
\]
We show how to build a new, valid logbook $L_{k+1} = (L^0_{k+1}, L^1_{k+1}, w_{k+1})$ for the next round. Assume that $G$-Spoiler moves as follows:

\[
(q_k, \{s_k\}, a_k, (q_k)') \in \Gamma^\text{Sp}_G,
\]

i.e., the next input symbol is $a_k$ and Spoiler moves existential-Left pebbles from $q_k$ to $(q_k)’$. We take $w_{k+1} = w_k \cdot a_k$. Notice that, if $s_k$ is existential, then Spoiler moves the only Right pebble from $s_k := \{s_k\}$ to $(s_k)’ := \{(s_k)’\}$, otherwise $(s_k)’ := \emptyset$.

For any existential state $r^E \in r^E_k$, let $\pi^0_{k,i}$ be the path in $L^0_k$ ending in $(q_k, i, r^E)$ (i.e., $r^0_{k,i} = r^E$), which is guaranteed to exist by property (P1). Spoiler’s move (S) above induces the $G_0$-Spoiler’s move below

\[
(q_k, \{s_k\}, r^E_0, a_k, (q_k)’, \{\}\} \in \Gamma^\text{Sp}_{G_0},
\]

which is obtained by restricting to $q_k^0, i \subseteq q_k$ the transition from $q_k$ to $(q_k)’$, the inclusion following from (P0). We apply $G_0$-Duplicator’s strategy $\sigma_0$, obtaining

\[
\sigma_0((q_k^0, r^0_{0,i})a_0 \cdots a_{k-1}(q_k^0, r^E)) (a_k, (q_k)’, \{\}) = (q_{k+1}^0, r^0_{k+1,i}).
\]  

(0-D)

The move (0-D) above fixes a successor $r^0_{k+1,i}$ for each existential state $r^E$ in $r^E_k$ (in $G_0$). We now consider these moves as adversarial in $G_1$, i.e., they induce a $G_1$-Spoiler’s move

\[
(r_k, \{s_k\}, a_k, (r_k)’, (s_k)’) \in \Gamma^\text{Sp}_{G_1},
\]

where $(r_k)’$ is the set of elements $r^0_{k+1,i}$ defined above. We then apply $G_1$-Duplicator’s strategy $\sigma_1$, obtaining

\[
\sigma_1((r_0, s_0) a_0 \cdots a_{k-1}(r_k, s_k))(a_k, (r_k)’, (s_k)’) = (r_{k+1}, \{s_{k+1}^0\}).
\]  

(1-D)

Notice that, if $s_k$ is universal, then $s_{k+1}^0 = (s_k)’$, as determined in (S), otherwise $s_{k+1}^0$ is determined by (1-D) above. The second component $L^1_{k+1}$ of the new logbook is $L^1_{k+1} = \pi^1_k \cdot (r_{k+1}, s_{k+1}^0)$.

Call a state $r \in r_k$ useful iff it has not been discarded by move (1-D), i.e., iff it has some successor in $r_{k+1}$. By the logbook property (P1), for each useful universal state $r^U \in r^U_k$, there exists a play $\pi^0_{k,i} \in L^0_k$ s.t. $\pi^0_{k,i} = \langle q_0^0_i, r_{0,i}^0 \rangle \cdots \langle q_{k-1}^0_i, r_{k-1,i}^0, r^U \rangle$ with $r_{k,i}^0 = r^U$. The $G_1$-Duplicator’s move (1-D) is then interpreted adversarially in $G_0$:

\[
(q_k^0, \{s_k\}, r^U) \in \Gamma^\text{Sp}_{G_0},
\]

and we apply $G_0$-Duplicator’s winning strategy $\sigma_0$, yielding

\[
\sigma_0((q_0^0_i, r_{0,i}^0) a_0 \cdots a_{k-1}(q_{k-1}^0_i, r^U))(a_k, (q_k)’, r^0_{k+1,i}) = (q_{k+1}^0, r^0_{k+1,i}).
\]  

(0-D’)

We now update the first component $L^0_k$ of the logbook. For any useful existential or universal state $r^0_{k,i} \in r_k$ with corresponding play $\pi^0_{k,i} \in L^0_k$ (as above), let $r^0_{k+1,i}$ be as determined in (0-D) or (0-S’), respectively. Then, we add $\pi^0_{k,i} \cdot (q_{k+1}^0, r^0_{k+1,i})$ to $L^0_{k+1}$. Since every element in $r_{k+1}$ arises as a successor of some useful element in $r_k$, we have that (P1) holds at round $k + 1$. 

29
Finally, we define Duplicator’s move in $\mathcal{G}$ as

$$(\sigma_0 \triangleright \sigma_1) ((q_0, s_0) a_0 \cdots a_{k-1} (q_k, s_k)) (a_k, (q_k')', (s_k')') := (q_{k+1}, s_{k+1}) \text{.}$$  \hspace{1cm} (D)$$

where $q_{k+1}$ is as union over all sets $q_{i+1}^{0,i}$ defined by equations (0-D) and (0-D'), and $s_{k+1} := s_k^{1,k+1} \cap s_k'$ is defined according to (S) or (1-D), depending on whether $s_k$ was universal or existential, respectively. Notice that, by definition of $q_{k+1}$ and $s_{k+1}$, properties (P0) and (P2) hold for the new logbook. This completes the description of the joint strategy $\sigma_0 \triangleright \sigma_1$.

The following theorem shows that $\equiv^*_{(n,1)}$ is transitive, i.e., it shows that when $\sigma_0$ and $\sigma_1$ are both winning, then $\sigma_0 \triangleright \sigma_1$ is winning as well.

**Lemma 11.** Let $x \in \{\forall \mathfrak{d}, \mathfrak{d}, \mathfrak{e}, \mathfrak{f}\}$. Then,

$$q \equiv^*_{(n,1)} r \land r \equiv^*_{(n,1)} s \implies q \equiv^*_{(n,1)} s \text{.}$$

**Proof.** We refer to the logbook $L_k$ at round $k$ as defined above. For $x = \forall \mathfrak{d}$, we have to show that, whenever some pebble in $q_k$ is accepting, so is $s_k$. Assume $q_k \cap F \neq \emptyset$. Then, there exists $q^F \subseteq q_k \cap F$ and, by (P0), there exists $q^F_0 \subseteq q_k$ s.t. $q^F \in q^F_0$ and $(q^F_0, r^F_0)$ is the current configuration in some $\mathcal{G}_0$ game. Since $q^F_0 \cap F \neq \emptyset$ and $\sigma_0$ is a winning strategy, then $r^F_0 \subseteq F$. Hence, by the logbook property (P1), $r_k = \bigcup \{r^F_0\} \cap F \neq \emptyset$, where $(r_k, s_k)$ is the current configuration in $\mathcal{G}_1$. Since $r_k \cap F \neq \emptyset$ and $\sigma_1$ is winning, we have $s_k \in F$.

For $x = \mathfrak{d}$, assume that at round $k$ every pebble in $q_k$ is universally good since some previous round, i.e., $\text{good}^u(q_k)$ holds. Let $(q^F_0, r^F_0)$ be any configuration in $\mathcal{G}_0$. By (P0), $q^F_0 \subseteq q_k$, thus $\text{good}^u(q^F_0)$. Since $\sigma_0$ is winning, then, for every $i$, there exists $k(i) \in \text{good}^u(q^F_0)$, i.e., $k(i)$ is the least index $k' \subseteq F$. Let $i^*$ be the index for which $r^F_0 \subseteq F$ is the last pebble being accepting for the first time since round $k$, i.e., $i^* = \text{argmax}(k(i))$. Hence, at round $k(i^*) \geq k$, every pebble in $r^F_0$ has been universally good since round $k$. Since $\sigma_1$ is winning, then there exists $k' \geq k(i^*) \geq k \in F$. Let $k^* \leq k'$ be the minimal $k'' \subseteq F$. s.t. $s_k'' \subseteq F$. Therefore, $\text{good}^u(s_k'')$, i.e., $\{s_k''\}$ is existentially good since round $k$.

For $x = \mathfrak{e}$, assume that $q_k$ is universally good since some previous round for infinitely many $k$’s. By reasoning as above for delayed simulation, since $\sigma_0$ is winning, then $r_k$ is universally good since some previous round for infinitely many $k$’s. Finally, being $\sigma_1$ winning, we conclude that $s_k \in F$ for infinitely many $k$’s.

\[ \square \]

**D Section 5**

**D.1 Finite words**

**Lemma 5.** Let $Q$ be any alternating finite automaton, and let $\preceq$ be any preorder which implies finite-language inclusion. Then, $L^\text{fin}(q) = L^\text{fin}([q])$.

**Proof.** We proceed by induction on the length of $w \in \Sigma^*$. Let $q$ any state in $Q$, and let $[q]$ be its equivalence class. Notice that, $L^\text{fin}(q) = L^\text{fin}(q')$ for any $q' \in [q]$.

Assume \( w = \varepsilon \). If \( w \in L^{\text{fin}}(q) \) then \( q \in F \), hence \([q] \in F'\) and \( w \in L^{\text{fin}}([q]) \) as well. Conversely, if \( w \in L^{\text{fin}}([q]) \) then \([q] \in F'\), hence there exists \( q' \in [q] \cap F \). But \( L^{\text{fin}}(qF') = L^{\text{fin}}(q) \), hence \( w \in L^{\text{fin}}(q) \).

Assume \( w = a_0 \ldots a_{k-1} \) is a word of length \( k \), and let \( w' = a_1 \ldots a_{k-1} \). We proceed by case analysis on the type of \([q]\).

- Case 1: \([q] \in E'. \) We prove \( L^{\text{fin}}(q) \subseteq L^{\text{fin}}([q]) \), distinguishing two subcases.

  Subcase 1.1: \( q \in E \). Assume \( w \in L^{\text{fin}}([q]) \). Then, there exists \((q, a_0, q') \in \Delta \) s.t. \( w' \in L^{\text{fin}}(q') \), and, w.l.o.g., we may assume that \( q' \) is an \( a_0 \)-maximal successor of \( q \). (If not, then there exists a state \( q'' \supseteq q' \) which actually is an \( a_0 \)-maximal successor of \( q \), thus \( w' \in L^{\text{fin}}(q'') \) and then one can proceed from \( q'' \).) By induction hypothesis, \( w' \in L^{\text{fin}}([q']) \). Hence, by the definition of quotient, there exists \(([q], a_0, [q']) \in \Delta \), thus \( w \in L^{\text{fin}}([q]) \).

  Subcase 1.2: \( q \in U \). Let \( qE' \in [q] \cap E \) and, by definition of quotient, \( L^{\text{fin}}(q) = L^{\text{fin}}(qE') \), and then one can proceed as above from \( qE' \). Thus, \( L^{\text{fin}}(q) \subseteq L^{\text{fin}}([q]) \).

We now prove \( L^{\text{fin}}([q]) \subseteq L^{\text{fin}}(q) \). If \( w \in L^{\text{fin}}([q]) \), then \(([q], a_0, [q']) \in \Delta \) s.t. \( w' \in L^{\text{fin}}([q']) \). By the definition of quotient, there exist \( qE' \in [q] \) and \( q' \in [q'] \) s.t. \((qE', a_0, q') \in \Delta \). By induction hypothesis, \( w' \in L^{\text{fin}}(q') \) (we do not use the maximality of \( q' \) here), hence \( w \in L^{\text{fin}}(qE') = L^{\text{fin}}(q) \).

- Case 2: \([q] \in U' \). We prove \( L^{\text{fin}}(q) \subseteq L^{\text{fin}}([q]) \). Assume \( w \in L^{\text{fin}}([q]) \). Let \([q']\) be any element in \([Q]\) s.t. \(([q], a_0, [q']) \in \Delta \). We have to show that \( w' \in L^{\text{fin}}([q']) \).

By the definition of quotient, there exist \( \hat{q} \in [q] \) and \( q' \in [q'] \) (we do not use the minimality of \( q' \) here) s.t. \((\hat{q}, a_0, \hat{q}') \in \Delta \). We have that \( L^{\text{fin}}(q) = L^{\text{fin}}(\hat{q}) \), hence \( w \in L^{\text{fin}}(\hat{q}) \). Since \( \hat{q} \in U \), then every \( a_0 \)-successor of \( \hat{q} \) accepts \( w' \). In particular, \( w' \in L^{\text{fin}}(\hat{q}') \), and, by induction hypothesis, \( w' \in L^{\text{fin}}([q']) \). But \([q']\) was arbitrary, thus \( w \in L^{\text{fin}}([q]) \).

We prove \( L^{\text{fin}}([q]) \subseteq L^{\text{fin}}(q) \). Assume \( w \in L^{\text{fin}}([q]) \). Let \( q' \) be any element in \( Q \) s.t. \((q, a_0, q') \in \Delta \), and we have to show \( w' \in L^{\text{fin}}(q') \) for any such \( q' \). In particular, it is sufficient to show \( w' \in L^{\text{fin}}(q') \) for any \( a_0 \)-minimal \( q' \), since \( L^{\text{fin}}(q') \subseteq L^{\text{fin}}(q'' \approx q') \) for any \( q'' \supseteq q' \). Hence, we assume that \( q' \) is an \( a_0 \)-minimal successor of \( q \). Being \([q] \in U' \), we have that \( w' \in L^{\text{fin}}([q]) \) for any \( a_0 \)-successor \([q']\) of \([q] \). As \((q, a_0, q') \in \Delta \) and by the definition of quotient, there exists \((q, a_0, [q']) \in \Delta \). Thus, \( w' \in L^{\text{fin}}([q']) \), and, by induction hypothesis, \( w' \in L^{\text{fin}}(q') \). But \( q' \) was arbitrary, thus \( w \in L^{\text{fin}}(q) \). \( \square \)

D.2 Infinite words: Direct simulation

The two directions in Theorem 8 are proved, resp., by Lemma 12 and Lemma 13 below.

Lemma 12. If \( q \equiv_{(1, n)} s \), then \( [q] \equiv_{(1, n)} [s] \), where the quotient is taken w.r.t. \( \equiv_{(1, k)} \).

Proof. Let \( G = G^{(1, n)}(\{q_i\}, s_i) \) and, at round \( i \), if the current configuration of \( G \) is \( \langle q_i, s_i \rangle \), let \( G_i = G^{(1, n)}(q_i, s_i) \). We maintain the following invariant: At round \( i \), \( q_i \equiv_{(1, i)} s_i \). Notice that the invariant implies the lemma: The crucial observation is that \([q_i] \subseteq F' \) implies \([q_i] \subseteq F \), i.e., if one state in the quotient is accepting, then, by the definition of direct simulation, all states in the quotient are accepting as well, and,
in particular, \( q_i \in F \). By the invariant and by the definition of \( \forall \delta \)-simulation, \( q_i \in F \) implies \( s_i \subseteq F \).

Assume the current configuration in \( G \) is \( \langle [q_i], s_i \rangle \), and \( q_i \subseteq (1,n) s_i \). Let Spoiler choose the next input symbol \( a_i \). We consider two cases, depending on whether \( [q_i] \) is existential or universal.

First case: \( [q_i] \in E' \). Let Spoiler choose an \( a_i \)-successor \([q_i+1]\) of \([q_i]\), i.e., Spoiler chooses transition

\[
([q_i], s_i, a_i, \{[q_{i+1}]\}, s') \in I^\text{Sp}_{G}.
\]

By the definition of minimax quotient, there exist \( \hat{q} \in [q_i] \cap E \) and \( q' \in [q_{i+1}] \) s.t. \( q' \in s(\hat{q}, a) \). (Note that we do not use the maximality of \( q' \) in this proof.) We have \( \hat{q} \equiv_{\forall \delta}(1,k) S_q \equiv_{\forall \delta}(1,n) s_i \). But \( (1,k) \)-simulation implies \((1,n)\)-simulation (Theorem 1), therefore \( \hat{q} \equiv_{\forall \delta}(1,n) q_i \), and, by transitivity, \( \hat{q} \equiv_{\forall \delta}(1,n) s_i \). We let \( G(\hat{q}, s_i) \)-Duplicator’s winning strategy, obtaining transition \( \langle \hat{q}, s_i, a_i, \{q', s'\}, \{q', \{s'\}, s_{i+1}\}\rangle = I^\text{Dup}_{G(\hat{q}, s_i)} \). Clearly, \( q' \in [q_{i+1}] \), \( q' \equiv_{\forall \delta}(1,n) s_{i+1} \), and the invariant is preserved.

We define \( G \)-Duplicator’s response as

\[
([q_i], s_i, a_i, [q_{i+1}], s', [q_{i+1}], s_{i+1}) \in I^\text{Dup}_{G}.
\]

Second case: \( [q_i] \in U' \). By the definition of quotient, \( q_i \in U \). Let Spoiler choose transition

\[
([q_i], s_i, a_i, \{\}, s') \in I^\text{Sp}_{G}.
\]

We let \( G_i \)-Spoiler choose transition \( ([q_i], s_i, a_i, \{\}, s') \in I^\text{Sp}_{G_i} \), and then we apply \( G_i \)-Duplicator’s winning strategy, obtaining \( ([q_i], s_i, a_i, \{\}, s', [q_{i+1}], s_{i+1}) \in I^\text{Dup}_{G_i} \). The crucial point is that we can assume \( \forall \delta \)-minimal \( a_i \)-successor of \( q_i \). In particular, it is also \( k \)-\( \forall \delta \)-minimal. This implies that there exists a \( a_i \)-transition in the quotient automaton from \([q_i]\) to \([q_{i+1}]\). Thus, \( G \)-Duplicator’s response is defined as

\[
([q_i], s_i, a_i, \{\}, s', [q_{i+1}], s_{i+1}) \in I^\text{Dup}_{G}.
\]

Clearly, \( q_{i+1} \equiv_{\forall \delta}(1,n) s_{i+1} \), and the invariant is preserved also in this case. \( \square \)

Lemma 13. If \( q \equiv_{\forall \delta}(1,n) s \), then \( q \equiv_{\forall \delta}(1,n) [s]_{m^+} \), where the quotient is taken w.r.t. \( \equiv_{\forall \delta}(1,k) \).

Proof. Let \( G = G(q, [s]) \) and, at round \( i \), if the current configuration of \( G \) is \( \langle q_i, [s_i] \rangle \), let \( G_i = G(q_i, s_i) \). We maintain the following invariant: \( q_i \equiv_{\forall \delta}(1,n) s_i \). The invariant implies the lemma: if \( q_i \in F \), then, by the definition of \( \forall \delta \)-simulation, \( s_i \subseteq F \), thence, by the definition of quotient, \([s_i] \subseteq F^+ \).

Assume the current configuration in \( G \) is \( \langle q_i, [s_i] \rangle \), and \( q_i \equiv_{\forall \delta}(1,n) s_i \). Let Spoiler choose the next input symbol \( a_i \) and a transition

\[
([q_i], [s_i], a_i, q', [s']) \in I^\text{Sp}_{G}.
\]

where \([s']\) is obtained by fixing a successor \([s']\) for any \([s'']\) \in \([s_i] \cap U' \). (Notice that, if \( q_i \in E \), then \( q' = \{q_{i+1}\} \) is just a singleton, for some \( q_{i+1} \in D(q_i, a) \), otherwise, when

...
Since 13.

Lemma 14. Infinite words: Delayed simulation

\( q_i \in U \), we have that \( q' = \{ \} \) as Left universal pebbles are under Duplicator’s control. By the definition of minimax quotient, \( ([s^{U}], a_i, [s']) \in \Delta_{m+} \) implies that there exist \( s^{U} \in [s'] \) and \( s' \in [s'] \) s.t. \( s' \in \Delta(s^{U}, a_i) \). (*) Let \( s' \) be the set of states \( s' \) obtained above. (We do not use the minimality of \( s' \) in this proof.) For any mixed class \( [s^{U}_{\text{mix}}] \in [s_i] \), for which its representative \( s^{U}_{\text{mix}} \in s_i \) is universal, let \( s^{E}_{\text{mix}} \in [s^{U}_{\text{mix}}] \cap E \) be a \((1, k)\)-\( \forall \text{di} \)-equivalent existential representative, for which, in particular, \( s^{U}_{\text{mix}} \equiv_{\forall \text{di}} s^{E}_{\text{mix}} \). Let \( p_i \) be equal to \( s_i \), but where each \( s^{U}_{\text{mix}} \) is replaced by \( s^{E}_{\text{mix}} \). By the invariant, \( q_i \equiv_{\forall \text{di}} (1, n) \) \( s_i \) and, by the definition of \( p_i \) and by transitivity, \( q_i \equiv_{\forall \text{di}} p_i \). The game then proceeds by using \( p_i \) in place of \( s_i \). Notice that universal states in \( p_i \) are exactly those universal states in \( s_i \) which belong to a purely universal quotient. We let \( G(q_i, p_i) \)-spoiler choose transition \( ([q_i], p_i, a_i, q', s') \in I^{\text{3p}}_{G(q_i, p_i)}, \) where \( s' \) is obtained from \( p_i \cap U \subseteq s_i \cap U \) by fixing successors as prescribed in (*) above. We then apply \( G(q_i, p_i) \)-Duplicator’s winning strategy, yielding transition \( ([q_i], p_i, a_i, q', s', [q_{i+1}], p_{i+1}) \in I^{\text{3p}}_{G(q_i, p_i)} \). It might be the case that some transition \( (p^E, a_i, p') \in \Delta \) induced above, for \( p^E \in p_i \cap E \) and \( p' \in p_{i+1} \). (If any) does not induce a transition in the quotient, i.e., there exists no corresponding transition \( ([p^E], a_i, [p']) \in \Delta_{m+} \). This happens when, in definition of minimax+ quotient, \( p' \) is not selected as a \( k \)-maximal representative for \( a_i \)-successors of \( p^E \). If this is the case, then, by the definition of minimax+ quotient, there exists some \( k \)-maximal \( a_i \)-successor \( s'' \subseteq \max^{E}_{a_i}([p^E]) \) s.t. \( p' \equiv_{\forall \text{di}} s'' \) and, for all \( s'' \in s'' \), \( ([p^E], a_i, [s'']) \in \Delta_{m+} \). We define \( s_{i+1} \) as \( p_{i+1} \), where elements \( p' \) are replaced by \( s'' \), as specified above. Then, \( G \)-Duplicator’s response is defined as

\[
([q_i], [s_i], a_i, q', [s'], [q_{i+1}], [s_{i+1}]) \in I^{\text{3p}}_{G}.
\]

Since \( q_{i+1} \equiv_{\forall \text{di}} p_{i+1} \), and, by the definition of \( s_{i+1} \) and transitivity, \( q_{i+1} \equiv_{\forall \text{di}} (1, n) \) \( s_{i+1} \), the invariant is preserved. \( \square \)

**Theorem 8.** \( q \equiv_{\forall \text{di}} (1, n) [q]_{m+} \), where the quotient is taken w.r.t. \( \equiv_{\forall \text{di}} \). In particular, \( L^w(q) = L^w([q]_{m+}) \).

Proof. Since \( q \equiv_{\forall \text{di}} (1, n) \) \( \{q\} \) trivially holds, the theorem follows from previous Lemma 12 and 13. \( \square \)

**D.3 Infinite words: Delayed simulation**

**Lemma 14.** Let \( q, s \in U \). If \( q \approx_{(1, n)} s \), then, for any \( q' \in \min^{n,x}_{a} (q) \), there exists \( s' \in \min^{n,x}_{a} (s) \) s.t. \( q' \approx_{(1, n)} s' \).

**Proof.** We actually prove the following richer statement.

**Claim.** Let \( q, s \in U \) s.t. \( q \approx_{(1, n)} s \). Then, for any \( q' \in \min^{n,x}_{a} (q) \), there exists \( s' \in \Delta (s, a) \) s.t. \( s' \subseteq_{(1, n)} q' \), and, for any \( s'' \in \Delta (s, a) \), \( s'' \equiv_{(1, n)} q' \) implies both 2.1) \( s'' \in \min^{n,x}_{a} (s) \), and 2.2) \( q' \approx_{(1, n)} s'' \).

Let \( q \approx_{(1, n)} s \), and let \( q' \in \min^{n,x}_{a} (q) \). Point 1) follows from the definition of simulation, i.e., there exists \( s' \in \Delta (s, a) \) s.t. \( s' \subseteq_{(1, n)} q' \).
We now show Points 2.1) and 2.2), i.e., we show that any such \( s' \) is in fact an \( x \)-minimal \( a \)-successor of \( s \). Let \( s'' \in \Delta(s,a) \) be any other \( a \)-successor of \( s \) s.t. \( s'' \not\subseteq (1,n) \) \( s' \). We have to show \( s' \preceq^\pi(1,n) s'' \) as well. Since \( q \preceq^\pi(1,n) s \), from the definition of simulation, there exists \( q'' \in \Delta(q,a) \) s.t. \( q'' \preceq^\pi(1,n) s'' \). Hence, we have the following chain of inclusions: \( q'' \preceq^\pi(1,n) s'' \preceq^\pi(1,n) s' \preceq^\pi(1,n) q' \). By the transitivity of \( \preceq^\pi(1,n) \) established in Theorem 6, we have \( q'' \preceq^\pi(1,n) q' \), and, by the minimality of \( q' \), \( q' \preceq^\pi(1,n) q'' \). By transitivity, all states in \( \{q'', s'', s', q'\} \) are \( x \)-simulation equivalent. In particular, \( s' \preceq^\pi(1,n) s'' \), which establishes Point 2.1), and \( q' \preceq^\pi(1,n) s' \), which establishes Point 2.2).

**Lemma 15.** Let \( s \in U \) and \( q \in E \). If \( q \preceq^\pi(1,n) s \), then there exists \( q' \subseteq \Delta(q,a) \) s.t., for any \( s' \in \min_{n \times}(s) \), \( s' \preceq^\pi(1,n) q' \).

**Proof.** Let \( s \in U \), \( q \in E \), and \( q \preceq^\pi(1,n) s \). From \( s \preceq^\pi(1,n) q \) and by the definition of simulation, there exists \( q' \subseteq \Delta(q,a) \) and \( s'' \in \Delta(s,a) \) s.t. \( s'' \preceq^\pi(1,n) q' \).

Let \( s' \) be any element in \( \min_{n \times}(s) \). From \( q \preceq^\pi(1,n) s' \) and by the definition of simulation, it follows that, for any \( q'' \in q' \), we have \( q'' \preceq^\pi(1,n) s' \). Since \( s'' \preceq^\pi(1,n) q' \), and any element in \( q' \) is simulated by \( s' \), we obtain, by transitivity (Theorem 6), \( s'' \preceq^\pi(1,n) s' \), and, by the minimality of \( s', s' \preceq^\pi(1,n) s'' \preceq^\pi(1,n) q' \). By transitivity, \( s' \preceq^\pi(1,n) q' \).

**Lemma 16.** If \( q \preceq^\text{de}(1,n) s \), then \( [q]_{\text{se}+} \preceq^\text{de}(1,n) s \), where the quotient is taken w.r.t. \( \preceq^\text{de}(1,n) \).

**Proof.** In the following, we simply write \( \preceq \) instead of \( \preceq_{\text{de}(1,n)} \). Then, when we write \( q \preceq_{\sigma} s \), we mean that \( \sigma \) is Duplicator’s winning strategy in \( G(q,s) \), i.e., the one witnessing \( q \preceq s \). In the proof, we need the following definitions: For any Duplicator’s strategy \( \sigma : PP_1 \mapsto (P_0 \mapsto P_1) \) and for any \( \pi \in P \), we define a new Duplicator’s strategy \( \sigma^\pi \) in the following way: For any \( \pi' \in PP_1 \), \( \sigma^\pi(\pi') := \sigma(\pi \cdot \pi') \). Given any Duplicator’s strategy \( \sigma \), we say that a sequence \( \pi^1 = s_k s_{k+1} \ldots \) is \( \sigma \)-right-conform starting at \( p_k \) iff there exist sequences \( \pi^0 = p_k p_{k+1} \ldots \) and \( \pi = \langle p_k, s_k \rangle \langle p_{k+1}, s_{k+1} \rangle \ldots \) is \( \sigma \)-conform w.r.t. \( \pi \). We will use the following fact:

**Claim.** Assume \( \pi^1 = s_k s_{k+1} \ldots \) is \( \sigma \)-right-conform starting at \( p_k \). If \( p_k \in F \) and \( \sigma \) is a winning strategy, then there exists \( i \geq k \) s.t. good\(^i\)(\( s_i, k \)).

We are now ready for proving the lemma. Let \( G = G([q], s) \) and, at round \( k \), if the current configuration of \( G \) is \( \langle [q], s_k \rangle \), let \( G_k = G(q_k, s_k) \). We build a sequence of winning strategies \( \sigma_0, \sigma_1, \ldots \), s.t., at round \( k \), \( \sigma_k \) is a winning strategy in \( G_k \), i.e., \( q_k \preceq_{\sigma_k} s_k \). Then, we define a strategy \( \sigma \) for Duplicator in \( G \), which, at round \( k \), is defined in terms of \( \sigma_k \). Finally, we prove that \( \sigma \) is winning.

Assume the current configuration in \( G \) is \( \langle [q], s \rangle \), and that \( s_k \) is a winning strategy in \( G_k \) s.t. \( q_k \preceq_{\sigma_k} s_k \). Let Spoiler choose the next input symbol \( a_k \). We consider two cases, depending on whether \( [q] \) is existential or universal.

First case: \( [q] \in E \). Let Spoiler choose an \( a_k \)-successor \( [q_{k+1}] \) of \( [q] \), i.e., Spoiler chooses transition \((\langle [q], s_k, a_k, [q_{k+1}] \rangle, s') \in I^\text{Sp}_G \).
By the definition of semielective quotient, there exist \( \tilde{q} \in [q_k] \) and \( q' \in [q_{k+1}] \) s.t. \( q' \in \Delta(\tilde{q}, a) \). If \( [q_k] \cap F \neq \emptyset \), then let \( q^F \) be any accepting state in \([q_k]\), otherwise let \( q^F \) be just \( \tilde{q} \). We distinguish two subcases, depending on whether \( \tilde{q} \) is in \( E \) or in \( U \).

- First subcase: \( \tilde{q} \in E \). We have that

  \[
  \tilde{q} \subseteq \sigma \quad q^F \subseteq_{\sigma^F} \quad q_k \subseteq_{\sigma_k} \quad s_k .
  \]

  We let \( G(\tilde{q}, s_k) \)-Spoiler choose transition \( (\{\tilde{q}\}, s_k, a_k, \{q'\}, s') \in \Gamma^\text{Sp}_{G(\tilde{q}, s_k)} \). Let \( \tilde{\sigma} = \tilde{\sigma} \bowtie \sigma^F \bowtie \sigma_k \), and let \( s_{k+1} \) be the result of \( G(\tilde{q}, s_k) \)-Duplicator playing according to \( \tilde{\sigma} \), i.e., \( \tilde{\sigma}(\{\tilde{q}\}, s_k)(a_k, \{q'\}, s') = \langle \{q'\} \rangle s_{k+1} \). Clearly, \( q' \subseteq \sigma^\pi \) \( s_{k+1} \), with \( \pi = (\tilde{q}, s_k) \). But \( q' \in [q_{k+1}] \), thus \( q_{k+1} \subseteq_{\sigma^\pi} q' \subseteq_{\sigma^\pi} s_{k+1} \) for some \( \sigma' \). By transitivity, \( q_k \subseteq_{\sigma_k \bowtie \sigma^\pi} s_{k+1} \). We let \( \sigma_{k+1} := \sigma' \bowtie \tilde{\sigma}^\pi \).

- Second subcase: \( \tilde{q} \in U \). By the definition of semielective quotient, \( q' \subseteq \min_{a^\text{de}}(\tilde{q}) \).

  (Notice that, although \( \tilde{q} \) is a universal state in this case, it is still Spoiler who has to choose a successor \( q' \) of \( \tilde{q} \), since \([q_k]\) is an existential state in the quotient automaton). Since \([q_k]\) is a mixed class, there exists \( q^E \subseteq [q_k] \cap E \) s.t.

  \[
  \tilde{q} \subseteq \sigma \quad q^E \subseteq_{\sigma^E} \quad q^F \subseteq_{\sigma^F} \quad q_k \subseteq_{\sigma_k} \quad s_k .
  \]

  Since \( \tilde{q} \subseteq q^F \), by the minimality of \( q' \) and Point 2) of Lemma 15, there exists \( q' \subseteq \Delta(q^E, a) \) s.t. \( q' \subseteq q^E \). Hence, w.l.o.g. \( \tilde{\sigma} \) can be taken s.t. \( \tilde{\sigma}(\{\tilde{q}\}, \{q^E\})(a_k, \{\} \{\}) = \langle \{q'\} \rangle \), where \( q' \) is fixed by \( G([q]), s \)-Spoiler, and not under the control of \( G(\tilde{q}, q^E) \)-Duplicator.

  Let \( \sigma := \tilde{\sigma} \bowtie \sigma^E \bowtie \sigma^F \bowtie \sigma_k \). Similarly to the previous point, \( G(\tilde{q}, s_k) \)-Spoiler chooses a transition \( (\{\tilde{q}\}, s_k, a_k, \{\}, \{\}) \in \Gamma^\text{Sp} \). We let \( G(\tilde{q}, s_k) \)-Duplicator answer with \( \sigma(\{\tilde{q}\}, s_k)(a_k, \{\}, \{\}) = \langle \{q'\} \rangle s_{k+1} \), where \( q' \) is the a \( a_k \)-successor fixed by \( G([q]), s \)-Spoiler above. As before, \( q_{k+1} \subseteq_{\sigma^\pi} q' \subseteq_{\sigma^\pi} s_{k+1} \), where \( \pi = (\tilde{q}, s_k) \).

  Hence, by transitivity, \( q_{k+1} \subseteq_{\sigma_k \bowtie \sigma^\pi} s_{k+1} \). We let \( \sigma_{k+1} := \sigma' \bowtie \tilde{\sigma}^\pi \).

In both cases, \( q_{k+1} \subseteq_{\sigma_{k+1}} s_{k+1} \). We define \( G \)-Duplicator’s winning strategy \( \sigma \) as

\[
\sigma(\pi_k(\{[q_k]\}, s_k))(a_k, \{[q_{k+1}]\}, s') = \langle \{q_{k+1}\} \rangle s_{k+1} ,
\]

Second case: \( [q_k] \in U' \). By the definition of quotient, \( q_k \in U \). Let Spoiler choose transition

\[
(\{[q_k]\}, s_k, a_k, \{\}, \{\}) \in \Gamma^\text{Sp} .
\]

If \( [q_k] \cap F \neq \emptyset \), let \( q^F \) be any accepting state in \([q_k]\), otherwise let \( q^F \) be just \( q_k \).

Then, we have

\[
q^F \subseteq_{\sigma^F} q_k \subseteq_{\sigma_k} s_k .
\]

Let \( \tilde{\sigma} = \sigma^F \bowtie \sigma_k \). Let \( G(q^F, s_k) \)-Spoiler choose transition \( (\{q^F\}, s_k, a_k, \{\}, \{\}) \in \Gamma^\text{Sp}_{G(q^F, s_k)} \), and let \( G(q^F, s_k) \)-Duplicator choose transition \( (\{q^F\}, s_k, a_k, \{\}, s', \{q_{k+1}\}, s_{k+1}) \in \Gamma^\text{Dup} \) \( G(q^F, s_k) \) according to \( \tilde{\sigma} \). We let \( \sigma_{k+1} := \tilde{\sigma}^\pi \), with \( \pi = \langle \{q^F\} \rangle s_k \). The crucial point is that we can assume w.l.o.g. that \( q_{k+1} \) is a de-minimal \( a_k \)-successor of \( q^F \). This implies

35
that there exists a $a_k$-transition in the quotient automaton from $[q_k]$ to $[q_{k+1}]$. Thus, $\mathbb{G}$-Duplicator’s response is defined as

$$\sigma(\pi_k([q_k], s_k))(a_k, \{\}, s') = ([q_{k+1}], s_{k+1}).$$

This concludes the description of the second case.

We now argue about the correctness of the construction above, showing that Duplicator’s strategy is winning in $\mathbb{G}$. If the Left pebble in $\mathbb{G}$ gets stuck, then Duplicator wins, and we are done. Otherwise, assume the Left pebble never gets stuck. By construction, since we are taking joins of winning strategies, it follows that some $Right$ pebble can always be moved, and the game does not halt prematurely. Thus, an infinite path $\pi = ([q_0], s_0)⟨[q_1], s_1⟩\ldots$ results, where $q_0 = q$ and $s_0 = s$. Assume $[q_k] \in F'$, for some $k$. There exists $q^F \in [q_k]$ s.t. $q^F \in F$ and, in any of the cases above, there exists a winning strategy $\sigma^F$ s.t. $q^F \sqsubseteq_{\sigma^F} q_k \sqsubseteq_{\sigma_k} s_k$. Let $\tilde{\sigma} := \sigma^F \bowtie \sigma_k$ be a winning strategy in $\mathbb{G}(q^F, s_k)$. By construction, the sequence $\pi^+ = s_k, s_{k+1}, \ldots$ is $\tilde{\sigma}$-right-conform starting at $q^F$. By the above claim, there exists $i \geq k$ s.t. $\text{good}^+ (s_i, k)$. \hfill $\square$

**Corollary 1.** $[q]_{\text{se+}} \sqsubseteq_{\text{di}}^{\text{de}} \langle 1, n \rangle q$.

The lemma below implies $q \sqsubseteq_{\text{di}}^{\text{de}} [q]_{\text{se+}}$. Notice that we actually prove the much stronger claim that $[q]_{\text{se+}}$ di-simulates $q$.

**Lemma 17.** For any $q \in Q$, $q \sqsubseteq_{\text{di}}^{\text{de}} \langle 1, 1 \rangle [q]_{\text{se+}}$.

**Proof.** We maintain the following invariant: If $(s_k, [q_k])$ is the current configuration in $\mathbb{G}(q, [q])$, then $s_k \in [q_k]$. Clearly, the invariant implies that the winning condition for direct simulation is satisfied: If $s_k \in F$, then $[q_k] \in F'$.

The initial configuration is $(s_0, [q_0])$ with $s_0 = q$, and $[q_0] = [q]$, and the invariant clearly holds since $s_0 \in [q_0]$.

Inductively, assume the current configuration is $(s_k, [q_k])$ and the invariant $s_k \in [q_k]$ holds. We distinguish three different cases.

- Case 1: $s_k \in E$. Then $[q_k] \in E'$. Assume Spoiler chooses transition

$$\langle \{s_k\}, \{[q_k]\}, a_k, \{s_{k+1}\}, \{\}\rangle \in \Gamma^{\text{Sp}}.$$

From $(s_k, a_k, s_{k+1}) \in \Delta$, the invariant $s_k \in [q_k]$ and by the definition of semielective quotient, there exists a transition $\langle [q_k], a_k, [s_{k+1}] \rangle \in \Delta^{\text{se+}}$. Thus, Duplicator can select transition

$$\langle \{s_k\}, \{[q_k]\}, a_k, \{s_{k+1}\}, \{\}, \{s_{k+1}\}, \{[s_{k+1}]\} \rangle \in \Gamma^{\text{Dup}}.$$

Clearly $s_{k+1} \in [s_{k+1}]$, and the invariant is preserved.

- Case 2: $s_k \in U$ and $[q_k] \in E'$. In this case, Spoiler only chooses $a_k$:

$$\langle \{s_k\}, \{[q_k]\}, a_k, \{\}, \{\} \rangle \in \Gamma^{\text{Sp}}.$$

36
If $s_k$ has no $a_k$-successor, then Duplicator wins. Otherwise, let $s_{k+1} \in \min_{de}^{n,de}(s_k)$ be a de-minimal $a_k$-successor of $s_k$. By the definition of semielective quotient and by the minimality of $s_{k+1}$, there exists a transition $([q_k], a_k, [s_{k+1}]) \in \Delta_{se+}^{n,de}$, thus

$$([s_k], [q_k], a_k, \emptyset, \emptyset, [s_{k+1}], ([s_{k+1}])) \in I^{\text{Dup}}.$$ 

Clearly $s_{k+1} \in [s_{k+1}]$, and the invariant is preserved.

– Case 3: $s_k \in U$ and $[q_k] \in U'$. In this case, we use the minimality of successors of universal states in universal classes. Assume Spoiler chooses transition

$$([s_k], [q_k], a_k, \emptyset, \emptyset, [q_{k+1}], ([q_{k+1}])) \in I^{\text{Sp}}.$$ 

From the definition of quotient, there exists a transition $(q_k, a_k, q_{k+1}) \in \Delta$ s.t. $q_{k+1} \in \min_{de}^{n,de}(q_k)$. From the invariant $s_k \in [q_k]$, we have $s_k \approx_{de}^{(1,n)} q_k$. By Lemma 14, there exists $s_{k+1} \in \min_{de}^{n,de}(s_k)$ s.t. $s_{k+1} \approx_{de}^{(1,n)} q_{k+1}$. Therefore, Duplicator can select transition

$$([s_k], [q_k], a_k, \emptyset, \emptyset, [s_{k+1}], ([s_{k+1}])) \in I^{\text{Dup}}$$

s.t. $s_{k+1} \in [q_{k+1}]$, thus preserving the invariant. \qed

**Remark 9.** Lemma 17 above is even true when quotienting w.r.t. fair simulation, or even ordinary simulation. Notice that requiring minimal transitions from universal states in mixed semielective-classes not only is required for correctness (see Section 5.2), but it also makes the proof much easier.

**Theorem 9.** $q \approx_{de}^{(1,n)} [q]_{se+}$, where the quotient is taken w.r.t. $\subseteq^{de}_{(1,n)}$. In particular, $L^w(q) = L^w([q]_{se+})$.

**Proof.** Directly from Corollary 1 and Lemma 17, and from the fact that simulation implies language inclusion (Theorem 4). \qed
Fig. 5. An example showing that (minimal) transitions from universal states in mixed classes are needed in semielective quotients. The only two \((1, n)\)-simulation equivalent states in \(Q\) are \(q_U\) and \(q_E\). (In fact, \(n = 2\) suffices.) The resulting mixed class in \(Q_{\approx}\) is \([\ ] = \{q_U, q_E\}\). The dashed \(a\)-transition on the right (due to \(q_U\)) is needed and cannot be discarded: Indeed, \(\mathcal{L}_\omega(Q) \neq \emptyset\), while removing the dashed transition from \(Q_{\approx}\) would make \(\mathcal{L}_\omega(Q_{\approx}) = \emptyset\).

Fig. 6. An example showing that multipebble-semielective quotients can achieve arbitrarily high compression ratios. The NBA above has \(k + 4\) states, and the \(p_i\)'s are \((1, 1)\)-delayed simulation incomparable: Thus, the \((1, 1)\)-semielective quotient has \(k + 4\) states. However, the \(p_i\)'s are all \((1, n)\)-delayed simulation equivalent (and \(n = 2\) suffices), therefore the \((1, n)\)-semielective quotient has only 4 states. Moreover, the \(p_i\)'s are incomparable also w.r.t. \((1, n)\)-universal direct simulation, which shows that semielective quotients can achieve arbitrarily high compression ratios relative to minimax quotients. Finally, notice that direct backward simulation does not help either: In fact, any two \(p_i, p_j\), with \(i \neq j\), are backward-simulation incomparable, as there is just one way of backward reaching the unique initial state \(p_0\). (Remember that backward simulations should be compatible with the initial states, at least.) Therefore, quotienting methods which employ backward simulations, like mediated preorder [1], do not result in a smaller automaton.
We give upper-bounds on the size of game-graphs necessary for computing multipebble simulations. When considering the size of the those game-graphs, we will make use of the following counting function:

\[ \text{sub}_n(k) = \sum_{i=0}^{k} \binom{n}{i}, \]

which counts the number of subsets of size \( \leq k \) of a given set of size \( n \), and we will approximate its value from above by using the following rough upper bound

\[ \text{sub}_n(k) \leq (n+1)^k. \]

Intuitively, the bound above may be seen as follows: Instead of counting sets of size \( \leq k \), one counts ordered sets; each ordered set can be represented as a \( k \)-string over an alphabet of size \( n+1 \), where we use an extra end-of-string symbol. We also give a formal calculation.

\[
\sum_{i=0}^{k} \binom{n}{i} = \sum_{i=0}^{k} \frac{n}{i} \cdot \frac{n-1}{i-1} \cdot \cdots \cdot \frac{n-i+1}{1} \\
\leq \sum_{i=0}^{k} \frac{n}{i} \cdot \frac{n}{i-1} \cdot \cdots \cdot \frac{n}{1} \\
\leq \sum_{i=0}^{k} n^i \cdot \left( \frac{k}{i} \right) \cdot \left( \frac{k-1}{i-1} \right) \cdot \cdots \cdot \left( \frac{k-i+1}{1} \right) \\
= \sum_{i=0}^{k} \left( \frac{k}{i} \right) \cdot n^i = (n+1)^k.
\]

### E.1 Solving existential and universal direct simulation

For computing the winner for direct simulation, we construct a 2-player game \( G_{di}^{di} = \langle V_{A}^{di}, V_{E}^{di}, \rightarrow_{di} \rangle \), where Eve has a safety objective. The game-graph is the same for both existential and universal direct simulation, but the safety objective is different.

Nodes in \( V_{A}^{di} \) take the form \( v(q,s) \), while nodes in \( V_{E}^{di} \) take the form \( v(q,s,a,q',s') \), with \( q, q' \in 2^{Q_{k_1}} \) and \( s, s' \in 2^{Q_{k_2}} \), and \( a \in \Sigma \).

**Lemma 18.** \( |V_{A}^{di}| \leq 2 \cdot (n+1)^{2(k_1+k_2)} \cdot |\Sigma| \).

**Proof.** \( |V_{A}^{di}| = \text{sub}_n(k_1) \cdot \text{sub}_n(k_2) \), and \( |V_{E}^{di}| \leq |\text{sub}_n(k_1)|^2 \cdot |\text{sub}_n(k_2)|^2 \cdot |\Sigma|. \) Hence, \( |V_{A}^{di} \cup V_{E}^{di}| \leq 2 \cdot |\text{sub}_n(k_1)|^2 \cdot |\text{sub}_n(k_2)|^2 \cdot |\Sigma| \leq 2 \cdot (n+1)^{2(k_1+k_2)} \cdot |\Sigma|. \) \( \square \)

Transitions in \( V_{A}^{di} \times V_{E}^{di} \) model choices of Spoiler: For any \( (q,s,a,q',s') \in I_{Sp} \), there is a transition for Adam \( v(q,s) \rightarrow v(q,s,a,q',s') \). Similarly, for any Duplicator’s
move \((q, s, a, q', s', q'', s'') \in T^{\text{Dup}}\), there exists a transition for Eve \(v_{(q, s, a, q', s')} \rightarrow v_{(q'', s'')}\) in \(V^E_{\text{di}} \times V^E_{\text{di}}\).

The winning criterion for existential direct simulation induces a set \(T^E = \{v(q, s) \in V^{A}_{E} \mid q \subseteq F \text{ implies } s \cap F \neq \emptyset\}\). Similarly, the safe set for universal direct simulation is \(T^V = \{v(q, s) \in V^{A}_{U} \mid q \cap F \neq \emptyset \text{ implies } s \subseteq F\}\). For \(x \in \{E, V\}\), we have that \(q \subseteq x_{\text{di}}(k_1, k_2)\) s iff Eve can ensure never leaving \(T^x\) when starting from \(v(q, s)\). This can be verified by checking whether \(v(q, s) \in W^x_{\text{di}}\), where

\[
W^x_{\text{di}} = \nu y. T^x \cap \text{cpre}(y).
\]

### E.2 Solving fair simulation

The game-graph for fair simulation is similar to the previous one for direct simulation, but with the difference that we need extra bookkeeping for recording whether each pebble has visited an accepting state or not. We let \(G^f = (V^f_{E}, V^f_{A}, \rightarrow^f)\), where Adam’s nodes in \(V^f_{E}\) are of the form \(v_{(q, \text{bad}, s, \text{good}, q', s', s'')}\) and Eve’s nodes in \(V^f_{A}\) are of the form \(v_{(q, \text{good}, a, q', s', s'')}\), where \(q, q', s, s', s'' \subseteq Q\). The sets \(\text{bad} \subseteq q\) and \(\text{good} \subseteq s\) record the current “badness/goodness” of states in \(q\) and \(s\), respectively, and they are used to detect events like “being good since some previous round”. Specifically, the event \(\text{bad} = \emptyset\) is used to detect when \(\text{good}^E(q)\), and similarly for \(\text{good} = s\) and \(\text{good}^V(s)\).

**Lemma 19.** \(|V^f_{E}| \leq 2 \cdot (n + 1)^{2(k_1 + k_2)} \cdot 2^{k_1 + k_2} \cdot |\Sigma|\).

**Proof.** \(|V^f_{A}| = \text{sub}_n(k_1) \cdot \text{sub}_n(k_2) \cdot 2^{k_1 + k_2}\), and \(|V^f_{E}| \leq [\text{sub}_n(k_1) \cdot \text{sub}_n(k_2)]^2 \cdot 2^{k_1 + k_2} \cdot |\Sigma|\). Hence, \(|V^f_{A} \cup V^f_{E}| \leq 2 \cdot [\text{sub}_n(k_1) \cdot \text{sub}_n(k_2)]^2 \cdot 2^{k_1 + k_2} \cdot |\Sigma| \leq 2 \cdot (n + 1)^{2(k_1 + k_2)} \cdot 2^{k_1 + k_2} \cdot |\Sigma|\) □

For any \((q, s, a, q''', s'') \in I^{\text{SP}}, v_{(q, \text{bad}, s, \text{good})} \rightarrow v_{(q, \text{bad}, s, \text{good}, a, q''', s'')}\), and, for any \((q, s, a, q''', s'', q', s') \in I^{\text{Dup}}, v_{(q, \text{bad}, s, \text{good}, a, q''', s'')} \rightarrow v_{(q', \text{bad}', s', \text{good}')}\), where

\[
good' = \begin{cases} s' \cap F & \text{if } \text{good} = s \\ \{s \in s' \mid s \in F \vee s \in \Delta(\text{good}, a)\} & \text{otherwise} \end{cases}
\]

\[
\text{bad}' = \begin{cases} q' \setminus F & \text{if } \text{bad} = \emptyset \\ \{q \in q' \mid q \notin F \land q \in \Delta(\text{bad}, a)\} & \text{otherwise} \end{cases}
\]

We notice the striking similarity of the update rule for bad pebbles and the updating rule for the second component in the MH-construction (Section 4). Intuitively, states in \(\text{bad}'\) are those states in \(q'\) which are not accepting and with some bad predecessor. Similarly, states in \(\text{good}'\) are those states in \(s'\) which are either accepting, or with some good predecessor. The correctness follows from the following simple fact:

**Claim.** Let \(\pi = v_0 v_1 \ldots\) be an infinite sequence of vertices, with \(v_i = v_{(q_i, \text{bad}, s_i, \text{good})}\) and s.t. \(v_i \rightarrow v_{i+1}\). Let \(\pi_1 = q_0 q_1 \ldots\) and \(\pi_3 = s_0 s_1 \ldots\) be the projections of \(\pi\) to the first and third component, respectively. Then, \(\text{good}^E(\pi_1, \infty)\) iff \(\text{bad}_d = \emptyset\) for infinitely many \(i\)'s, and \(\text{good}^V(\pi_3, \infty)\) iff \(\text{good}_d = s_i\) for infinitely many \(i\)'s.
Let \( T_1 \) be the set of states of the form \( v(q, \text{Bad}, s, \text{good}) \) with \( \text{bad} = \emptyset \), and let \( T_2 \) be the set of states of the form \( v(q, \text{Bad}, s, \text{good}) \) with \( \text{good} = s \). The winning criterion for fair simulation is translated in the following 1-pair Street condition (also known as a reactivity condition [9]): If \( T_1 \) is visited infinitely often, then \( T_2 \) is visited infinitely often. Therefore, winning nodes for Eve are those in \( W^f = \nu x \cdot \mu y \cdot \nu z \cdot (T_2 \cap \text{cpre}(x)) \cup (T_1 \cap \text{cpre}(y)) \cup \overline{T_1} \cap \text{cpre}(z) \).

### E.3 Solving delayed simulation

We recall the definition of the game-graph for computing delayed simulation: \( \mathcal{G}^{\text{de}} = \langle V^{\text{de}}_E, V^{\text{de}}_A, \rightarrow^{\text{de}} \rangle \), where

\[
V^{\text{de}}_A = \{ v(q, \text{Bad}, s, \text{Good}) \mid q, s \subseteq Q \}
\]

\[
V^{\text{de}}_E = \{ v(q, \text{Bad}, s, \text{Good}, o, q', s') \mid q, q', s, s' \subseteq Q \}
\]

and \( \text{Bad} \) and \( \text{Good} \) are two sequences of sets of states from \( Q \), strictly ordered by set-inclusion. More precisely, \( \text{Bad} = (b_1, \ldots, b_{m_1}) \) with \( 0 \leq m_1 \leq k_1 \), satisfies, for any \( i \in \{1, \ldots, m_1\} \),

\[
b_i \subseteq q \quad \text{(B1)}
\]

\[
b_{i+1} \subseteq b_i, \text{ when } i < m_1 \quad \text{(B2)}
\]

\[
b_{m_1} \neq \emptyset, \quad \text{(B3)}
\]

and \( \text{Good} = (g_1, \ldots, g_{m_2}) \) with \( 0 \leq m_2 \leq k_2 \), satisfies, for any \( i \in \{1, \ldots, m_2\} \),

\[
g_i \subseteq s \quad \text{(G1)}
\]

\[
g_{i+1} \subseteq g_i, \text{ when } i < m_2 \quad \text{(G2)}
\]

We also denote with \( \text{Bad} \) the set \( \{b_1, \ldots, b_{m_1}\} \), and similarly for \( \text{Good} \).

The following lemma states that Algorithm 1 preserves the definition of \( \text{Bad} \), \( \text{Good} \):

**Lemma 20.** If \( \text{Bad} \) and \( \text{Good} \) satisfy properties (B1)-(B3) and (G1)-(G2), resp., then the same holds for the sets \( \text{Bad}' \) and \( \text{Good}' \) as computed by Algorithm 1.

**Proof.** Properties (B1) and (G1) are preserved by how \( b' \) and \( g' \) are constructed, on lines 4 and 15, respectively. Similarly, the strictness of the order, i.e., (B2) and (G2), is preserved by removing duplicate elements (lines 6 and 17). Finally, property (B3) follows by the check at line 8, which enforces that empty elements are removed from \( \text{Bad}' \), if any (line 9).

\[
|V^{\text{de}}| \leq 2 \cdot (n+1)^2(k_1+k_2) \cdot (1+k_1+1)^{k_1+1} \cdot (1+2(k_1+1)^{k_2+1}) \cdot |\Sigma|.
\]

**Proof.** We first count the number of pairs \((s, \text{Good})\). Assume \( |g_{m_2}| = h \leq k_2 \) (notice that \( m_2 \leq h \)). We consider two cases, depending on whether \( g_1 \neq \emptyset \) or not. First case: \( g_1 \neq \emptyset \). Then, we can represent the strictly increasing sequence \( g_1 \subseteq g_2 \subseteq \cdots \subseteq g_{m_2} \) by the sequence \( \{d_i\}_{1 \leq i \leq m_2} \) of non-empty differences, defined as \( d_1 = g_1 \) and
Algorithm 1: Updating the sequences Good and Bad

Input: The sequences \( \text{Good} = \langle g_1 \supset \cdots \supset g_{m_2} \rangle \) and \( \text{Bad} = \langle b_1 \supset \cdots \supset b_{m_1} \rangle \) to be updated, the current input symbol \( a \in \Sigma \) and the next configuration \( \langle q', s' \rangle \).

Output: The updated sequences \( \text{Good}' \) and \( \text{Bad}' \)

1. Add \( q \) to \( \text{Bad} \), i.e.,\[ \text{Bad} = \langle q \supset b_1 \supset \cdots \supset b_{m_1} \rangle \]
2. \( \text{Bad}' = () \);
3. \textbf{foreach (} \( b \in \text{Bad} \) \textbf{do}
4. \[ b' = \{ s \in q' \mid s \not\in F \land s \in \Delta(b, a) \} \]
5. Add \( b' \) to \( \text{Bad}' \);
6. Remove duplicate elements from \( \text{Bad}' \);
7. Assume \( \text{Bad}' = \langle b'_1 \supset \cdots \supset b'_{m_1} \rangle \);
8. \textbf{if (} \( b'_{m_1} = () \) \textbf{then}
9. Remove \( b'_{m_1} \) from \( \text{Bad}' \);
10. Add \( \emptyset \) to the front of \( \text{Good} \), i.e.,\[ \text{Good} = \langle \emptyset \subseteq g_1 \subset \cdots \subset g_{m_2} \rangle \]
11. \textbf{if (} \( g_{m_2} = s \) \textbf{then}
12. Remove \( g_{m_2} \) from \( \text{Good} \);
13. \( \text{Good}' = () \);
14. \textbf{foreach (} \( g \in \text{Good} \) \textbf{do}
15. \[ g' = \{ s \in s' \mid s \in F \lor s \in \Delta(g, a) \} \]
16. Add \( g' \) to \( \text{Good}' \);
17. Remove duplicate elements from \( \text{Good}' \);
18. \textbf{return (} \( \text{Good}' \) \textbf{and} \( \text{Bad}' \) \textbf{)};

\[ d_{i+1} = g_{i+1} \setminus g_i \text{ for } i > 1. \] (We have that \( g_i = \bigcup_{j=1}^{i} d_i \), so no information is lost.) Notice that, by definition, \( D = \{ d_1, \ldots, d_{m_2} \} \) is a partition of \( g_{m_2} = \{ g_1, \ldots, g_h \} \), hence \( D \) may be represented by a surjective function \( f \) from \( g_{m_2} \) to \( D \) s.t. \( f(g_i) = d_j \) iff \( g_i \in d_j \). Let \( \{ h \} \) be the Stirling number of the second kind. Then, the number of sequences \( \emptyset \neq g_1 \subset g_1 \subset \cdots \subset g_{m_2} \) is\[ \text{seq}_h(m_2) := \{ h \} \cdot m_2! . \]

Second case: \( g_1 = \emptyset \). Thus \( g_2 \neq \emptyset \), hence the number of sequences \( \emptyset = g_1 \subset g_2 \subset \cdots \subset g_{m_2} \) is just as before, but with one less element in the sequence, i.e., \( \text{seq}_h(m_2 - 1) \). Hence, the number of pairs (\( s, \text{Good} \)) is\[ f_2(n, k_2) = \sum_{i=1}^{k_2} \binom{n}{i} \left( 1 + \sum_{h=1}^{i} \binom{h}{h} \sum_{m_2=1}^{k_2} (\text{seq}_h(m_2 + \text{seq}_h(m_2 - 1))) \right) . \]
I.e., we sum over all sizes \( i \) for sets \( s \), and either Good is empty, or Good is non-empty. In this second case, we sum over all possibilities for the size \( h \in \{1, \ldots, i\} \) of the largest \( g_{m_2} \) (by (G1), every \( g \in \text{Good} \) is \( \subseteq s \)), and over all possibilities for the number of elements \( m_2 \) in Good: For each such combination of indices, we have \( \text{seq}_h(m_2) + \text{seq}_h(m_2 - 1) \) sequences. We now proceed to derive a bound on \( f_2(n, k_2) \). As \( \text{seq}_h(m_2) \) represents the number of surjective functions from a set of size \( h \) to a set of size \( m_2 \), clearly \( \text{seq}_h(m_2) \leq m_2^h \), just considering all such functions. Then,

\[
f_2(n, k_2) \leq \sum_{i=1}^{k_2} \binom{n}{i} \left( 1 + \sum_{h=1}^{i} \binom{i}{h} \sum_{m_2=1}^{h} 2 \cdot m_2^h \right)
\]

\[
\leq \sum_{i=1}^{k_2} \binom{n}{i} \left( 1 + \sum_{h=1}^{i} \binom{i}{h} \sum_{m_2=1}^{h} 2 \cdot h^h \right)
\]

\[
\leq \sum_{i=1}^{k_2} \binom{n}{i} \left( 1 + 2 \cdot \sum_{h=1}^{i} \binom{i}{h} h^{h+1} \right)
\]

\[
\leq \sum_{i=1}^{k_2} \binom{n}{i} \left( 1 + 2 \cdot i \cdot \sum_{h=1}^{i} \binom{i}{h} h^h \right)
\]

\[
\leq \sum_{i=1}^{k_2} \binom{n}{i} \left( 1 + 2 \cdot \cdot (i + 1)^i \right)
\]

\[
\leq \sum_{i=1}^{k_2} \binom{n}{i} \left( 1 + 2 \cdot (i + 1)^i \right)
\]

\[
\leq \sum_{i=1}^{k_2} \binom{n}{i} \left( 1 + 2 \cdot (k_2 + 1)^{k_2+1} \right)
\]

\[
\leq (n + 1)^{k_2}. (1 + 2 \cdot (k_2 + 1)^{k_2+1}).
\]

We now count the number of pairs \((q, \text{Bad})\). Assume \(|b_1| = h \leq k_1\) (notice that \(m_1 \leq h\)). By an argument similar to the one in the previous paragraph, we have that the number of non-empty sequences \(b_1 \supset b_2 \supset \cdots \supset b_{m_1} \neq \emptyset\) is \( \text{seq}_h(m_1) \). Hence, the number of pairs \((q, \text{Bad})\) is

\[
f_1(n, k_1) = \sum_{i=1}^{k_1} \binom{n}{i} \left( 1 + \sum_{h=1}^{i} \binom{i}{h} \sum_{m_1=1}^{h} \text{seq}_h(m_1) \right),
\]

which, with a similar calculation to the above, can be shown to be bounded by

\[
f_1(n, k_1) \leq (n + 1)^{k_1} \cdot (1 + (k_1 + 1)^{k_1+1}).
\]

Finally, \(|V_{\text{A}_{\text{de}}}^\text{de}| \leq f_1(n, k_1) \cdot f_2(n, k_2)\), and \(|V_{\text{E}_{\text{de}}}^\text{de}| \leq f_1(n, k_1) \cdot f_2(n, k_2) \cdot (n + 1)^{k_1+k_2} \cdot |\Sigma|\), thus \( |V_{\text{A}_{\text{de}}}^\text{de} \cup V_{\text{E}_{\text{de}}}^\text{de}| \leq 2 \cdot f_1(n, k_1) \cdot f_2(n, k_2) \cdot (n + 1)^{k_1+k_2} \cdot |\Sigma| \leq 2 \cdot (n + 1)^{2(k_1+k_2)} \cdot (1 + (k_1 + 1)^{k_1+1}) \cdot (1 + 2 \cdot (k_2 + 1)^{k_2+1}) \cdot |\Sigma|\). \(\square\)