?

For simplicity in stating our results, we define $\subseteq^{\mathrm{di}} = \subseteq^{\mathrm{de}} = \subseteq^{\mathrm{f}} = \subseteq^{\mathrm{fw}}$.

Jumping backward containment is the jumping analogue of backward containment from Section 3.3. Intuitively, two states q and s are in *jumping backward containment* $\subseteq^{\mathrm{bw}}(\sqsubseteq_0, \sqsubseteq_1)$ iff initial $\sqsubseteq_0$-jumping paths ending in q are matched by corresponding initial $\sqsubseteq_1$-paths ending in s, and, if the first path takes an accepting transition, then so does the second.

Definition 4.2.4 (Jumping backward containment). *For two states q and s, let $q \subseteq^{\mathrm{bw}} (\sqsubseteq_0, \sqsubseteq_1)\ s$ iff, for any initial $\sqsupseteq_0$-jumping path $q_0 \sqsupseteq_0 \hat{q}_0 \xrightarrow{a_0} \cdots \xrightarrow{a_{n-1}} q_n \sqsupseteq_0 \hat{q}_n$ ending at $\hat{q}_n = q$, there exists an initial $\sqsupseteq_1$-jumping path $s_0 \sqsupseteq_1 \hat{s}_0 \xrightarrow{a_0} \cdots \xrightarrow{a_{n-1}} s_n \sqsupseteq_1 \hat{s}_n$ ending at $\hat{s}_n = s$, s.t., for any $0 \leq i < n$,*

$$q_i \sqsupseteq_0^F \hat{q}_i \textit{ implies } s_i \sqsupseteq_1^F \hat{s}_i$$

4.2.2 Jumping-safe preorders

Jumping-safe preorders are central in this chapter. Intuitively, a preorder is jumping-safe if fair jumping paths do not introduce new words into the language of the automaton.

Definition 4.2.5 (Jumping-safe preorder). *For an automaton $\mathcal{Q}$, a preorder $\sqsubseteq \subseteq Q \times Q$ is* jumping-safe *(w.r.t. $\mathcal{Q}$) iff*

$$\mathcal{L}^{\sqsubseteq}(\mathcal{Q}) = \mathcal{L}(\mathcal{Q})$$

Therefore, if a preorder is jumping-safe, then fair&initial jumping paths can always be replaced by fair&initial non-jumping paths.

In the next section we introduce several PTIME preorder transformers. The main result of this chapter says that transformers map jumping-safe preorders into jumping-safe preorders. (cf. Theorems 4.5.3 and 4.5.4).

4.3 Jumping simulation transformers

In this section we introduce several simulation transformers τ, which map a given preorder $\sqsubseteq$ into a new preorder $\tau(\sqsubseteq)$. The transformer τ is defined game-theoretically via a variant of the usual simulation game.

Fix a preorder $\sqsubseteq$. We introduce a novel class of simulation games where both Spoiler and Duplicator can *jump* during the game. The new position a player can jump to depends on the preorder $\sqsubseteq$: If a player's pebble is on q, then the pebble can

instantaneously jump to any state $\hat{q}$ s.t. $q \sqsubseteq \hat{q}$. We call state $\hat{q}$ a *proxy*, which acts as a dynamic mediator for taking jumping transitions. From the proxy, the pebble can then take an ordinary a-transition to some state q'. Overall, we have a jumping transition $q \sqsubseteq \hat{q} \xrightarrow{a} q'$, which we usually abbreviate as $q \sqsubseteq \xrightarrow{a} q'$ by keeping the proxy implicit.

The winning condition depends on the specific transformer that we consider. We study four transformers τ^x, for $x = \mathrm{di}, \mathrm{de}, \mathrm{f}, \mathrm{bw}$, in analogy with direct, delayed, fair and backward simulation.

Intuitively, the acceptance condition is shifted from states to transitions. A pebble is no longer statically accepting in a given position, but can instead dynamically take *accepting $\sqsubseteq$-transition.* A $\sqsubseteq$-transition is accepting if the pebble can "transit through" an accepting state before reaching the proxy. We write an accepting a-transition from q to q' as $q \sqsubseteq \xrightarrow{a}_F q'$: Expanding the definition, this means that there exist states $q^F \in F$ and $\hat{q}$ s.t. $q \sqsubseteq q^F \sqsubseteq \hat{q} \xrightarrow{a} q'$.

In the formal definition we actually allow Spoiler and Duplicator to jump w.r.t. two different preorders. This more general definiton is used in Section 4.5.1 to analyze what happens when the two players have different (but related) jumping capabilities; except for that, usually Spoiler and Duplicator have the same jumping capability (this is crucial for transitivity; see Section 4.3.3).

4.3.1 Definitions

Fix two preorders $\sqsubseteq_0$ and $\sqsubseteq_1$, controlling the jumping capability of Spoiler and Duplicator, respectively. Formally, the configurations of the basic $(\sqsubseteq_0, \sqsubseteq_1)$-simulation game between state q and state s, $G_{q,s}(\sqsubseteq_0, \sqsubseteq_1)$, are the same as in ordinary simulation. That is, they are pairs of states $\langle q_i, s_i \rangle$. The difference lies in the fact that more transitions are available in the $(\sqsubseteq_0, \sqsubseteq_1)$-game than in the ordinary game. We first discuss forward transformers.

bad notation

Forward transformers Let $x = \mathrm{di}, \mathrm{de}, \mathrm{f}$. Initially, the game is in configuration $\langle q_0, s_0 \rangle$. Subsequently, if in round i the current configuration is $\langle q_i, s_i \rangle$, then the configuration for the next round $i + 1$ is determined as follows:

- First, Spoiler chooses an input symbol a_i and a $\sqsubseteq_0$-jumping transition

$$q_i \sqsubseteq_0 \xrightarrow{a_i} q_{i+1}$$

4. *Backward* $(\sqsubseteq_0, \sqsubseteq_1)$*-simulation*, $x = \mathrm{bw}$. Duplicator wins if two conditions are satisfied, the first regarding final states and the second initial states.

lam

   a) Whenever Spoiler takes an accepting backward $\sqsubseteq_0$-transition, Duplicator takes an accepting backward $\sqsubseteq_1$-transition (in the same round):

$$\forall(i \geq 0) \cdot q_i \sqsubseteq_0 \xleftarrow{a_i}_F q_{i+1} \implies s_i \sqsubseteq_1 \xleftarrow{a_i}_F s_{i+1}$$

   b) Initial states are treated differently, since they do not need to be part of transitions: Whenever Spoiler could possibly $\sqsubseteq_0$-jump to an initial state, Duplicator can $\sqsubseteq_1$-jump to an initial state:

$$\forall(i \geq 0) \cdot q_i \sqsubseteq_0 I \implies s_i \sqsubseteq_1 I$$

The rules for backward jumping simulation are summarized below.

Simulation game for $\tau^{\mathrm{bw}}(\sqsubseteq_0, \sqsubseteq_1)$

Current configuration		Next configuration
$\langle q_i, s_i \rangle$	$\dashrightarrow$	$\langle q_{i+1}, s_{i+1} \rangle$

Spoiler chooses $q_i \sqsubseteq_0 \xleftarrow{a_i} q_{i+1}$

Duplicator replies with $s_i \sqsubseteq_1 \xleftarrow{a_i} s_{i+1}$

Winning condition

1) $\forall(i \geq 0) \cdot q_i \sqsubseteq_0 \xleftarrow{a_i}_F q_{i+1} \implies s_i \sqsubseteq_1 \xleftarrow{a_i}_F s_{i+1}$
2) $\forall(i \geq 0) \cdot q_i \sqsubseteq_0 I \implies s_i \sqsubseteq_1 I$

As usual for simulation games, the binary relation $\tau^x(\sqsubseteq_0, \sqsubseteq_1)$ is defined as follows: $q\, \tau^x(\sqsubseteq_0, \sqsubseteq_1)\, s$ iff Duplicator wins the jumping x-simulation game $G^x_{q,s}(\sqsubseteq_0, \sqsubseteq_1)$. Moreover, when Spoiler and Duplicator have the same jumping capabilities w.r.t. a given preorder $\sqsubseteq$, we simply write $\tau^x(\sqsubseteq)$ instead of $\tau^x(\sqsubseteq, \sqsubseteq)$; this is very important, since $\tau^x(\sqsubseteq)$ is also a preorder (cf. Corollary 4.3.7 in Section 4.3.3).

lam

4.3.2 Basic properties

We investigate some basic properties of τ^x, which are immediate consequences of jumping games.

Inclusions Forward simulations for $x = \mathrm{di}, \mathrm{de}, \mathrm{f}$ are linearly ordered by inclusion, direct simulation being the finest and fair simulation the coarsest. This follows immediately from the fact that "easier" winning conditions favour Duplicator.

Lemma 4.3.1. *For* $x = \mathrm{di}, \mathrm{de}, \mathrm{f}$ *and preorders* $\sqsubseteq_0, \sqsubseteq_1$, *let* $\preceq^x = \tau^x(\sqsubseteq_0, \sqsubseteq_1)$. *Then,*

$$\preceq^{\mathrm{di}} \subseteq \preceq^{\mathrm{de}} \subseteq \preceq^{\mathrm{f}}$$

Monotonicity and non-decreasingness When we give more jumping power to Duplicator, τ^x clearly grows. Symmetrically, when Spoiler can perform "longer" jumps, τ^x shrinks. Therefore, the two-arguments transformer $\tau^x(\cdot, \cdot)$ is *antitone* in the first argument and *isotone* in the second argument. This means that, for preorders $\sqsubseteq_0, \sqsubseteq_1, \sqsubseteq_0', \sqsubseteq_1'$,

$$\sqsubseteq_0 \subseteq \sqsubseteq_0' \text{ and } \sqsubseteq_1 \subseteq \sqsubseteq_1' \quad \text{implies} \quad \tau^x(\sqsubseteq_1', \sqsubseteq_0) \subseteq \tau^x(\sqsubseteq_1, \sqsubseteq_0')$$

In particular, when we only allow Duplicator to jump, we get a reflexive relation (not necessarily transitive) coarser than ordinary simulation.

Lemma 4.3.2. *Let* $x = \mathrm{di}, \mathrm{de}, \mathrm{f}, \mathrm{bw}$, *and let* $\sqsubseteq^x = \tau^x(Id)$ *be the corresponding ordinary simulation preorder. Then, for any preorder* $\sqsubseteq$ *and states* q *and* s,

$$q \sqsubseteq^x s \quad \textit{implies} \quad q\, \tau^x(Id, \sqsubseteq)\, s$$

Otherwise, when we give the same jumping power to both players at the same time, the resulting one-argument transformer $\tau^x(\cdot)$ is no longer monotone. The lack of such a basic property might seem unfortunate. However, we can prove that $\tau^x(\cdot)$ enjoys another related useful property, that of being *non-decreasing* up to transposition. That is, $\tau^x(\sqsubseteq)$ is at least as coarse as the transpose of $\sqsubseteq$.

Lemma 4.3.3 (Non-decreasingness). *Let* $x = \mathrm{di}, \mathrm{de}, \mathrm{f}, \mathrm{bw}$. *For any preorder* $\sqsubseteq$ *and states* q *and* s,

$$q \sqsubseteq s \quad \textit{implies} \quad s\, \tau^x(\sqsubseteq)\, q$$

Proof. First, notice that, since $\sqsubseteq_1 \subseteq \sqsubseteq_1'$, also $\sqsubseteq_1^F \subseteq [\sqsubseteq_1']^F$, which we often use below.

For $x = \mathrm{di}$, assume $q_i \sqsubseteq_0 \xrightarrow{a_i}_F q_{i+1}$. Since σ_0 is winning, $r_i \sqsubseteq_1 \xrightarrow{a_i}_F r_{i+1}$, thus $r_i \sqsubseteq_1' \xrightarrow{a_i}_F r_{i+1}$. Since σ_1 is winning, $s_i \sqsubseteq_2 \xrightarrow{a_i}_F s_{i+1}$. This shows that $\sigma_0 \bowtie \sigma_1$ is winning for $x = \mathrm{di}$.

For $x = \mathrm{bw}$, final states are propagated as above (by flipping transitions). For initial states, assume $q_i \sqsubseteq_0 I$. Since σ_0 is winning, $r_i \sqsubseteq_1 I$, therefore $r_i \sqsubseteq_1' I$. Since σ_1 is winning, $s_i \sqsubseteq_2 I$. Therefore, $\sigma_0 \bowtie \sigma_1$ is winning for $x = \mathrm{bw}$.

For $x = \mathrm{de}$, assume $q_i \sqsubseteq_0 \xrightarrow{a_i}_F q_{i+1}$. Since σ_0 is winning in G_0, there exists $k \geq i$ s.t. $r_k \sqsubseteq_1 \xrightarrow{a_k}_F r_{k+1}$, thus $r_k \sqsubseteq_1' \xrightarrow{a_k}_F r_{k+1}$. Since σ_1 is winning in G_1, there exists $j \geq k \geq i$ s.t. $s_j \sqsubseteq_2 \xrightarrow{a_j}_F s_{j+1}$. Thus, $\sigma_0 \bowtie \sigma_1$ is winning for $x = \mathrm{de}$.

Finally, for $x = \mathrm{f}$, assume $q_i \sqsubseteq_0 \xrightarrow{a_i}_F q_{i+1}$ for infinitely many i's. Since σ_0 is winning, $r_i \sqsubseteq_1 \xrightarrow{a_i}_F r_{i+1}$ for infinitely many i's, which implies $r_i \sqsubseteq_1' \xrightarrow{a_i}_F r_{i+1}$ for infinitely many i's. Since σ_1 is winning, $s_i \sqsubseteq_2 \xrightarrow{a_i}_F s_{i+1}$ for infinitely many i's. Hence, $\sigma_0 \bowtie \sigma_1$ is winning for $x = \mathrm{f}$. □

As an immediate corollary, we have the following closure property of transformers.

Corollary 4.3.6. *Let* $x = \mathrm{di}, \mathrm{de}, \mathrm{f}, \mathrm{bw}$. *For preorders* $\sqsubseteq_0, \sqsubseteq_1, \sqsubseteq_1', \sqsubseteq_2$, *assume* $\sqsubseteq_1 \subseteq \sqsubseteq_1'$. *Then,*

Not

$\tau^x \ldots \circ \tau^x \subseteq \tau^x$

$$q \, \tau^x(\sqsubseteq_0, \sqsubseteq_1) \, r \ \wedge \ r \, \tau^x(\sqsubseteq_1', \sqsubseteq_2) \, s \quad \Longrightarrow \quad q \, \tau^x(\sqsubseteq_0, \sqsubseteq_2) \, s$$

By taking $\sqsubseteq = \sqsubseteq_0 = \sqsubseteq_1 = \sqsubseteq_1' = \sqsubseteq_2$ in the corollary above, we have that $\tau^x(\sqsubseteq)$ is transitive. Since it is also clearly reflexive, it is a preorder.

Corollary 4.3.7. *Let* $x = \mathrm{di}, \mathrm{de}, \mathrm{f}, \mathrm{bw}$. *For any preorder* $\sqsubseteq$, $\tau^x(\sqsubseteq)$ *is a preorder.*

4.4 Language containment and inclusion

In this section, we link jumping simulations to jumping containment and inclusion. In Section 4.4.1, we establish that jumping simulations are sound underapproximations of jumping containment (cf. Lemma 4.4.1). In Section 4.4.2, we show that jumping containment can be used to prove jumping language inclusion (cf. Lemma 4.4.2). Moreover, if the input preorder is jumping-safe, then also non-jumping language inclusion—that is, ordinary language inclusion—can be underapproximated (see Lemma 4.4.3 in Section 4.4.3). Thus, jumping simulations meet the simulation desiderata (Da).

Going from backward containment to forward language inclusion is a non-trivial task, since it requires filling the gap between finite paths (as in backward containment)

and infinite ones (as in ω-language inclusion). In Section 4.4.4, we develop a technique for converting a sequence of longer and longer finite paths into a single infinite path; we discuss *coherence*, which is a sufficient condition for the infinite path to be fair.

Not clear!!!

4.4.1 Jumping simulation implies jumping containment

We start off by establishing that jumping simulations imply jumping containment. This is an analogue of the corresponding classic result about ordinary simulation preorders.

Lemma 4.4.1 (Simulation implies containment). *For* $x = \mathrm{di}, \mathrm{de}, \mathrm{f}, \mathrm{bw}$, *let* $\preceq^x = \tau^x(\sqsubseteq_0, \sqsubseteq_1)$. *Then, for two states* q *and* s,

$$q \preceq^x s \quad \text{implies} \quad q \ \subseteq^x(\sqsubseteq_0, \sqsubseteq_1) \ s$$

Proof. We first prove the lemma for forward transformers $x = \mathrm{di}, \mathrm{de}, \mathrm{f}$. By the inclusions in Lemma 4.3.1, it suffices to cosider $x = \mathrm{f}$. Let $q \preceq^{\mathrm{f}} s$, and assume $w \in \mathcal{L}^{\sqsubseteq_0}(q)$. That is, there exists a fair $\sqsubseteq_0$-jumping path

$$\pi = q_0 \sqsubseteq_0 \xrightarrow{a_0} q_1 \sqsubseteq_0 \xrightarrow{a_1} \cdots$$

Lam

starting at $q_0 = q$. In the simulation game, from the initial configuration $\langle q, s \rangle$, we let Spoiler play as to follow π. That is, in round i, Spoiler plays transition $q_i \sqsubseteq_0 \xrightarrow{a_i} q_{i+1}$. Since $q \preceq^{\mathrm{f}} s$, Duplicator has a winning strategy to reply with. Thus, Duplicator builds a $\sqsubseteq_1$-jumping path

$$\pi' = s_0 \sqsubseteq_1 \xrightarrow{a_0} s_1 \sqsubseteq_1 \xrightarrow{a_1} \cdots$$

starting at $s_0 = s$. Since π is fair, Spoiler plays accepting transitions infinitely often. But Duplicator is winning for $x = \mathrm{f}$, therefore also Duplicator plays accepting transitions infinitely often. Thus π' is fair as well, and $w \in \mathcal{L}^{\sqsubseteq_1}(s)$.

For $x = \mathrm{bw}$, let $q \preceq^{\mathrm{bw}} s$ and let

$$\pi = q_0 \sqsupseteq_0 \hat{q}_0 \xrightarrow{a_0} q_1 \sqsupseteq_0 \hat{q}_1 \xrightarrow{a_1} \cdots \xrightarrow{a_{n-1}} q_n \sqsupseteq_0 \hat{q}_n, \text{ with } \hat{q}_n = q$$

be an initial $\sqsupseteq_0$-jumping path ending in q. The argument is the same as above. The simulation game starts from configuration $\langle q, s \rangle$, and Spoiler plays by choosing backward transitions according to π. Since $q \preceq^{\mathrm{bw}} s$, Duplicator has a winning strategy in the simulation game. Therefore, Duplicator builds a matching initial $\sqsupseteq_1$-jumping path π' ending in s,

$$\pi' = \hat{s}_0 \xrightarrow{a_0} s_1 \sqsupseteq_1 \hat{s}_1 \xrightarrow{a_1} \cdots \xrightarrow{a_{n-1}} s_n \sqsupseteq_1 \hat{s}_n, \text{ with } \hat{s}_n = s$$

By the definition of backward jumping simulation,

4.4.4 Coherent sequences of jumping paths

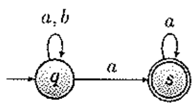
In this section, we present a general method to show the existence of fair paths based on the existence of certain sequences of *finite* paths. Contextually, we apply this technique to prove Lemma 4.4.2 from Section 4.4.2.

Fix an infinite word $w = a_0 a_1 \cdots \in \Sigma^\omega$. The idea is to start with a sequence of longer and longer finite initial paths $\Pi := \pi_0, \pi_1, \ldots$ over suitable prefixes of w. We are interested in finding a sufficient condition for the existence of an initial and fair infinite path over w. Since fair paths have infinitely many accepting states, a necessary condition is that the number of accepting states in paths π_i's grows unboundedly.

In the case of deterministic automata, this condition is also sufficient: Indeed, in a deterministic automaton there exists a unique run over w, which is accepting precisely when the number of accepting stated visited by its prefixes goes to infinity. In this case, we say that the π_i's are *strongly coherent*, since longer paths conservatively extends shorter ones.

However, in the general case of nondeterministic automata it is quite possible to have paths that visit arbitrarily many final states, and, still, no accepting run exists. This occurs because final states can appear arbitrarily late in the path, as shown in the next example.

Example 4.4.1 - Visiting arbitrarily many final states is not sufficient



Consider the automaton $\mathcal{Q}$ above. Take the infinite word $w = aba^2ba^3b\cdots$. For every prefix of the form $w_i = aba^2b\cdots a^{i-1}ba^i$, there exists a w_i-path

$$\pi_i = q \xrightarrow{ab} q \xrightarrow{a^2b} \cdots \xrightarrow{a^{i-1}b} \underbrace{q \xrightarrow{a} s \xrightarrow{a} s \xrightarrow{a} \cdots \xrightarrow{a} s}_{i \text{ times}}$$

visiting the final state s i times. Still, no fair path exists over w. Therefore $w \notin \mathcal{L}(\mathcal{Q})$.

The issue is that accepting states appear just in the tail of the path, and they never "stabilize" in any prefix of bounded length. To prevent this, we require accepting states

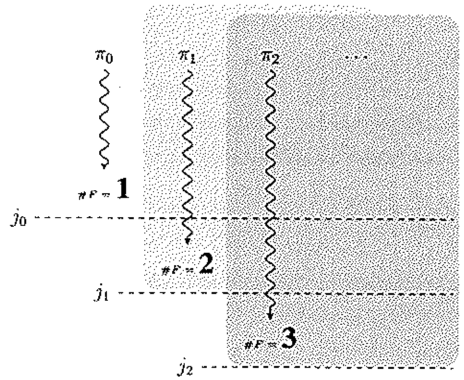


Figure 4.2: Coherent paths

to spread uniformiformly across the path. We split the infinite time horizon into slices $0 < j_0 < j_1 < \cdots$, and we require that for each interval $[0, j_i]$ and index $k \geq i$, path π_k visits at least i final states within the first j_i steps. See Figure 4.2. When this condition is satisfied, we say that the sequence of paths Π is *coherent*.

We give the formal definitions for the more general case of jumping paths. For a (finite or infinite) $\sqsubseteq$-jumping path $\pi = q_0 \sqsubseteq \xrightarrow{a_0} q_1 \sqsubseteq \xrightarrow{a_1} \cdots$, let $\#F(\pi, j)$ be the number of accepting transitions within the first j transitions:

$$\#F(\pi, j) := \sum_{0 \leq i < j} [q_i \sqsubseteq \xrightarrow{a_i}_F q_{i+1}]$$

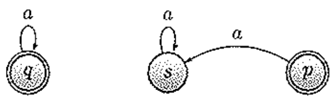
Definition 4.4.5. *Fix a word $w \in \Sigma^\omega$. Let $\Pi := \pi_0, \pi_1, \ldots$ be an infinite sequence of finite $\sqsubseteq$-jumping w-paths. Π is* coherent *if the following property holds:*

$$\forall i \cdot \exists (j > i) \cdot \forall (k \geq i) \cdot \#F(\pi_k, j) \geq i\,. \tag{4.1}$$

Let Π be defined as above. Sometimes we are interested in infinite subsequences of Π, obtained by removing some of its paths (possibly infinitely many). Formally, Π' is an *infinite subsequence of* Π iff $\Pi' := \pi_{f(0)}, \pi_{f(1)}, \ldots$ for some $f : \omega \mapsto \omega$ with $f(0) < f(1) < \cdots$. Coherent sequences are obviously preserved under the operation of taking infinite subsequences.

Lemma 4.4.6. *If Π is coherent, then any infinite subsequence Π' thereof is coherent.*

Example 4.5.1 - The transformer $\tau^{\mathrm{f}}(\succeq^{\mathrm{f}}, \sqsubseteq)$ is not reflexive



Consider the unary automaton above. Let $\sqsubseteq = Id$ and let $\preceq^{\mathrm{f}}$ be ordinary fair simulation $\preceq^{\mathrm{f}} = \tau^{\mathrm{f}}(Id)$. Since $\Sigma = \{a\}$, the only possible infinite word is a^{ω}. Notice that there is no fair a^{ω} path, neither from s nor from p. Therefore, s fairly simulates p, $s \succeq^{\mathrm{f}} p$, for the simple reason that Spoiler cannot build a fair path from p. However, since $s \succeq^{\mathrm{f}} p$, if we consider the fair $(\succeq^{\mathrm{f}}, Id)$-jumping game starting from configuration $\langle s, s \rangle$, Spoiler can now play the following fair $\succeq^{\mathrm{f}}$-jumping path from s:

$$\pi = s \succeq^{\mathrm{f}} p \xrightarrow{a}_F s \succeq^{\mathrm{f}} p \xrightarrow{a}_F \cdots$$

Since no ordinary fair path exists from s, Duplicator loses. Therefore, $s\ \tau^{\mathrm{f}}(\succeq^{\mathrm{f}}, Id)\ s$ does not hold.

The corollary below is an immediate consequence of Lemma 4.5.1, and it is at the heart of the central preservation property of Theorems 4.5.3 and 4.5.4 of Section 4.5.2.

Corollary 4.5.2. *For* $x = \mathrm{di}, \mathrm{de}, \mathrm{bw}$ *and any state* q, $q \subseteq^x(\succeq^x, \sqsubseteq)\ q$.

Proof. By Lemma 4.5.1, $q\ \tau^x(\succeq^x, \sqsubseteq)\ q$, and, by Lemma 4.4.1, $q \subseteq^x(\succeq^x, \sqsubseteq)\ q$. □

4.5.2 Preserving jumping-safe preorders

In this section we state the central result of this chapter: *direct, delayed and backward simulation transformers preserve jumping-safe preorders.* See Theorems 4.5.3 and 4.5.4. The proofs are elementary; they crucially use the reflexivity property of Corollary 4.5.2.

This opens the possiblity of repeatedly applying simulation transformers to the identity relation (which is trivially jumping-safe) to get coarser and coarser jumping-safe preorders; this is explored in Section 4.6.

Forward transformers

Theorem 4.5.3. *For* $x = \mathrm{di}, \mathrm{de}$, *let* $\preceq^x = \tau^x(\sqsubseteq)$. *If* $\sqsubseteq$ *is a jumping-safe preorder, then* $\succeq^x$ *is a jumping-safe preorder coarser than* $\sqsubseteq$.

In analogy with forward simulations, it is the transpose of $\preceq^x$ which is jumping-safe, not $\preceq^x$ itself.

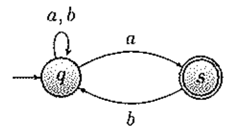
Proof. Let $\sqsubseteq$ be jumping-safe. By Lemma 4.3.3, $\succeq^x$ is coarser than $\sqsubseteq$. We show that $\succeq^x$ is jumping safe, i.e., $\mathcal{L}^{\succeq^x}(Q) = \mathcal{L}(Q)$. The right-to-left direction is immediate. For the other direction,

$$\mathcal{L}^{\succeq^x}(Q) = \bigcup_{q \in I} \mathcal{L}^{\succeq^x}(q) \overset{(0)}{\subseteq} \bigcup_{q \in I} \mathcal{L}^{\sqsubseteq}(q) = \mathcal{L}^{\sqsubseteq}(Q) \overset{(1)}{=} \mathcal{L}(Q)$$

where inclusion (0) follows from $q \subseteq^x(\succeq^x, \sqsubseteq)\ q$ (by Corollary 4.5.2), and equality (1) holds since $\sqsubseteq$ is jumping-safe. □

Theorem 4.5.3 fails for the fair transformer. In the example below we show that this holds already by taking as input the identity relation $\sqsubseteq = Id$.

Example 4.5.2 - The fair transformer does not preserve jumping-safe preorders



It is well-known that ordinary fair simulation $\sqsubseteq^{\mathrm{f}} = \tau^{\mathrm{f}}(Id)$ is not a GFQ preorder [44]. It turns out that this happens precisely because $\sqsupseteq^f$ is not jumping-safe. Consider the automaton above. State q recognizes the language $(\Sigma^* ab)^{\omega}$, and $q \sqsupseteq^{\mathrm{f}} s$. Consider $w = b^{\omega}$; w admits the following fair $\sqsupseteq^f$-jumping path from q:

$$\pi = q \sqsupseteq^f s \xrightarrow{b} q \sqsupseteq^f s \xrightarrow{b} \cdots$$

Therefore, $w \in \mathcal{L}^{\sqsupseteq^f}(q)$. But $w \notin \mathcal{L}(q)$, since there is no ordinary fair path from q over w. Therefore, $\sqsupseteq^f$ is not jumping-safe.

Backward transformer

Theorem 4.5.4. *Let* $\preceq^{\mathrm{bw}} = \tau^{\mathrm{bw}}(\sqsubseteq)$. *If* $\sqsupseteq$ *is a jumping-safe preorder, then* $\preceq^{\mathrm{bw}}$ *is a jumping-safe preorder coarser than* $\sqsupseteq$.

Second, de-winning strategies propagate final states even under composition on the left.

Lemma 4.5.8. *Let σ^0, σ^1 be two composable strategies for Duplicator in $G^{\mathrm{de}}_{q_0,r_0}(\sqsubseteq_0, \sqsubseteq_1)$ and $G^{\mathrm{de}}_{r_0,s_0}(\sqsubseteq_1', \sqsubseteq_2)$, respectively. If $r_0 \in F$ and σ^1 is a winning strategy, then, for any $(\sigma^0 \bowtie \sigma^1)$-conform play $\pi = \pi_0 \times \pi_2$ from $\langle q_0, s_0 \rangle$,*

$$\pi_0 = q_0 \sqsubseteq_0 \xrightarrow{a_0} q_1 \sqsubseteq_0 \xrightarrow{a_1} \cdots \qquad \pi_2 = s_0 \sqsubseteq_2 \xrightarrow{a_0} s_1 \sqsubseteq_2 \xrightarrow{a_1} \cdots$$

Duplicator is eventually accepting, *i.e., there exists $i \geq 0$ s.t. $s_i \sqsubseteq_2 \xrightarrow{a_i}_F s_{i+1}$.*

Proof. Let $\pi = \pi_0 \times \pi_2$ be any $(\sigma^0 \bowtie \sigma^1)$-conform play. Moreover, let $\pi_1 = r_0 \sqsubseteq_1 \xrightarrow{a_0} r_1 \sqsubseteq_1 \xrightarrow{a_1} \cdots$ be the intermediate play. It follows that $\pi_1 \times \pi_2$ is σ^1-conform. Since $r \in F$ and σ^1 is winning, there exists $i \geq 0$ s.t. $s_i \sqsubseteq_2 \xrightarrow{a_i}_F s_{i+1}$. □

We can now finish the proof for the last case of Lemma 4.5.1.

Proof of Lemma 4.5.1 (for $x = \mathrm{de}$). The idea is to split the play in $G^{\mathrm{de}}_{q,q}(\succeq, \sqsubseteq)$ into stages $k_0 < k_1 < \cdots$ (starting at $k_0 = 0$), and define a sequence of strategies $\sigma_0, \sigma_1, \ldots$, s.t., during the i-th stage, Duplicator plays according to the i-th strategy. Stage i starts when, in round k_i, Spoiler is accepting. Therefore, during stage i, there is a pending obligation for Duplicator to visit an accepting state. When this obligation is eventually fulfilled by σ_i, the next stage can start as soon as Spoiler is accepting again. When this does not happen, there are only finitely many stages. Otherwise, a new stage $i+1$ starts when Spoiler is accepting in round k_{i+1}, and Duplicator switches to the next strategy σ_{i+1}.

In round i, we stipulate that the current configuration of the game is $\langle q_i, s_i \rangle$, where, initially, $\langle q_0, s_0 \rangle = \langle q, q \rangle$. Moves of Spoiler take the form of $\succeq$-jumping transitions $q_i \succeq \xrightarrow{a_i} q_{i+1}$, whereas Duplicator's responses are $\sqsubseteq$-jumping transitions of the form $s_i \sqsubseteq \xrightarrow{a_i} s_{i+1}$.

Formally, we define two sequences of indices $\{k_i\}_{i \geq 0}$ and $\{h_i\}_{i \geq 0}$ by induction. Initially, let $k_0 = h_0 = -1$. For $i \geq 0$,

$$k_{i+1} = \min(\{j > h_i \mid q_j \succeq \xrightarrow{a_j}_F q_{j+1}\} \cup \{\omega\})$$
$$h_{i+1} = \min(\{j \geq k_{i+1} \mid s_j \sqsubseteq \xrightarrow{a_j}_F s_{j+1}\} \cup \{\omega\})$$

Intuitively, h_i is the time Duplicator matches the i-th pending obligation and k_{i+1} is the time Spoiler raises the next, $(i+1)$-th pending obligation. If the i-th pending obligation is not eventually fulfilled, then $k_i < h_i = k_{i+1} = \cdots = \omega$. If the i-th pending obligation is the last one to be raised and it is eventually fulfilled, then $h_i < k_{i+1} = h_{k+1} = \cdots = \omega$. Otherwise, when infinitely many obligations are raised and fulfilled, the sequence does not converge. In general, the two sequences are thus interleaving:

$$k_0 \leq h_0 < k_1 \leq h_1 < \cdots$$

At any stage $i \geq 0$,

- Let $q_0^F = q_0$ and, for $i > 0$ and $k_i < \omega$, let $q_i^F \in F$ be the accepting state s.t.

  $$q_{k_i} \succeq q_i^F \succeq \xrightarrow{a_{k_i}} q_{k_i+1}$$

  which exists by the definition of k_i.

- Let σ_i^0 be a $\preceq$-respecting strategy in the game $G^{\mathrm{de}}_{q_i^F, q_i^F}(\succeq, \sqsubseteq)$, which exists by Corollary 4.5.6.

- Assuming $q_i^F \preceq s_{k_i}$, let σ_i^1 be a *winning strategy* in the game $G^{\mathrm{de}}_{q_i^F, s_{k_i}}(\sqsubseteq, \sqsubseteq)$.

Let k be the current round, and let $\pi = \pi_0 \times \pi_1$ be the current partial play, where

$$\pi_0 = q_0 \succeq \xrightarrow{a_0} q_1 \succeq \xrightarrow{a_1} q_2 \succeq \xrightarrow{a_2} \cdots \succeq \xrightarrow{a_{k-1}} q_k$$
$$\pi_1 = s_0 \sqsubseteq \xrightarrow{a_0} s_1 \sqsubseteq \xrightarrow{a_1} s_2 \sqsubseteq \xrightarrow{a_2} \cdots \sqsubseteq \xrightarrow{a_{k-1}} s_k$$

Let Spoiler play the jumping transition $q_k \succeq \xrightarrow{a_k} q_{k+1}$. Assume the game is in stage i, that is, $k_i \leq k < k_{i+1}$. Let $\pi[k_i, k] = \pi_0' \times \pi_1'$ be the modified suffix of π starting in round k_i, where

$$\pi_0' = q_i^F \succeq \xrightarrow{a_{k_i}} q_{k_i+1} \succeq \xrightarrow{a_{k_i+1}} \cdots \succeq \xrightarrow{a_{k-1}} q_k$$
$$\pi_1' = s_{k_i} \sqsubseteq \xrightarrow{a_{k_i}} s_{k_i+1} \sqsubseteq \xrightarrow{a_{k_i+1}} \cdots \sqsubseteq \xrightarrow{a_{k-1}} s_k$$

Then, Duplicator plays according to the following strategy σ_i:

$$\sigma_i(\pi) := [\sigma_i^0 \bowtie \sigma_i^1](\pi[k_i, k])$$

Notice that π_0' starts at q_i^F (and not at q_{k_i}), since σ_i^0 is a strategy starting from configuration $\langle q_i^F, q_i^F \rangle$.

σ_i^0 and σ_i^1 are composable. To show that σ_i is well-defined, we need to ensure that σ_i^1 always exists, i.e., that $q_i^F \preceq s_{k_i}$ holds throughout the game. This holds initially, since both q_0^F and s_{k_0} equal q, and $\preceq$ is reflexive. Inductively, assume $q_i^F \preceq s_{k_i}$

Therefore, we restrict ourselves to composing forward and backward transformers in a strictly alternating fashion, as we explore in the next section.

4.6.2 Proxy simulations

For two transformers τ and τ', let their composition be $\tau;\tau'$, where τ is applied first:

$$\tau;\tau'(\sqsubseteq) = \tau'(\tau(\sqsubseteq))$$

As basic building blocks we define the four composite transformers below, obtained by composing a forward transformer with the backward one:

$$\tau^{\mathrm{di+bw}} := \tau^{\mathrm{di}};\tau^{\mathrm{bw}} \qquad \tau^{\mathrm{bw+di}} := \tau^{\mathrm{bw}};\tau^{\mathrm{di}}$$
$$\tau^{\mathrm{de+bw}} := \tau^{\mathrm{de}};\tau^{\mathrm{bw}} \qquad \tau^{\mathrm{bw+de}} := \tau^{\mathrm{bw}};\tau^{\mathrm{de}}$$

Forward proxy simulations are obtained by applying a cascade of $\tau^{\mathrm{di+bw}}/\tau^{\mathrm{de+bw}}$ operations starting from the identity relation. See Figure 4.3(a), where each node in the tree is a forward proxy simulation. Backward proxy simulations are defined similarly, but w.r.t. transformers $\tau^{\mathrm{bw+di}}/\tau^{\mathrm{bw+de}}$. See Figure 4.3(b).

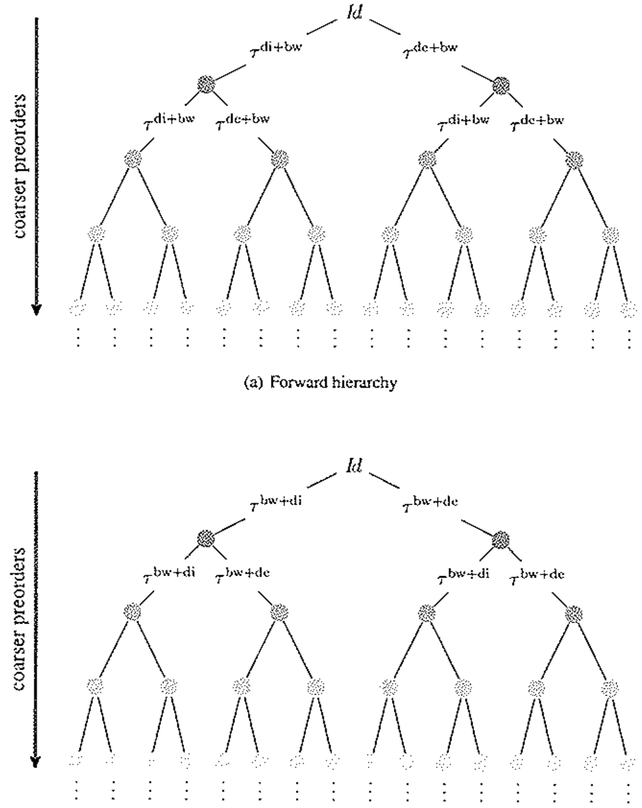
Definition 4.6.3. $\preceq$ *is a* forward proxy simulation *iff there exists a sequence of transformers* $\tau_0\tau_1\ldots\tau_n \in \{\tau^{\mathrm{di+bw}},\tau^{\mathrm{de+bw}}\}^*$ *s.t.* $\preceq = \tau_0;\tau_1;\cdots;\tau_n(Id)$. *Similarly,* $\preceq$ *is a* backward proxy simulation *iff there exists a sequence of transformers* $\tau_0\tau_1\ldots\tau_n \in \{\tau^{\mathrm{bw+di}},\tau^{\mathrm{bw+de}}\}^*$ *s.t.* $\preceq = \tau_0;\tau_1;\cdots;\tau_n(Id)$.

Proxy simulations are jumping-safe by an immediate inductive argument from Theorems 4.5.3 and 4.5.4.

Lemma 4.6.4. *If* $\preceq$ *is a forward or backward proxy simulation, then* $\succeq$ *(resp.,* $\preceq$*) is jumping-safe. In particular,* $\preceq$ *is GFQ.*

Remark 4.6.5. In previous work [28], we introduced two preorders, which we ~~have~~ called direct and delayed proxy simulations. They coincide with the first level $\tau^{\mathrm{bw+di}}(Id)$ and $\tau^{\mathrm{bw+de}}(Id)$ of the backward hierarchy, respectively. Here, we obtain an entire hierarchy of coarser jumping-safe PTIME preorders, thus improving upon [28].

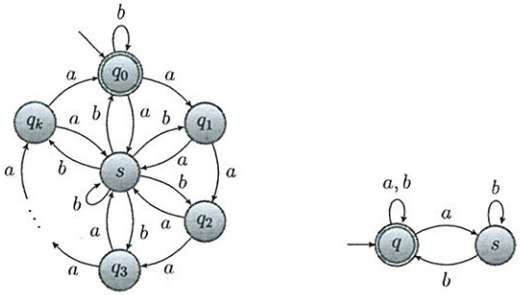
Proxy simulations along a branch of the tree form a non-decreasing chain of jumping-safe preorders. That is, if $\tau_0\tau_1\cdots\tau_n$ is a prefix of $\tau_0\tau_1\cdots\tau_{n'}$, with $n' \geq n$, then $\tau_0;\tau_1;\cdots;\tau_n(Id) \subseteq \tau_0;\tau_1;\cdots;\tau_{n'}(Id)$. On finite automata, these chains eventually



(a) Forward hierarchy

(b) Backward hierarchy

Figure 4.3: Proxy simulations hierarchies



a) Family of automata $\mathcal{Q}_k$ b) The quotients $\mathcal{Q}_k/\approx$

We show that backward proxy simulation quotients can outperform many other quotient techniques based on simulation-like preorders. Consider the family of automata $\mathcal{Q}_k$ above. Each automaton has $k+2$ states. We consider quotienting $\mathcal{Q}_k$ w.r.t. various simulation-like preorders.

Direct and delayed simulation No two different states are comparable w.r.t. delayed simulation (and, thus, w.r.t. direct simulation). Indeed, no state can simulate q_0: From q_0, Spoiler can play both actions a and b, whereas from states $s, q_1, q_2, \ldots, q_k$ only one of these actions is available. This implies that no two different states in the ring are comparable: From configuration $\langle q_i, q_j\rangle$ with $i \neq j$, Spoiler repeatedly plays a until forcing configuration $\langle q_0, q_h\rangle$, for some $h \neq 0$; from the latter configuration, Spoiler wins as above. (We assume that Duplicator does not go to s, which would make her lose early.) Finally, s cannot simulate any state q_i (since action a is unavailable from s), and no state q_i can simulate s: If $i \neq 0$, q_i cannot do b, and if $i = 0$, Spoiler goes from s to q_1 with action b and then wins as above.

Multipebble and fixed-word simulations Duplicator does not benefit from either having multiple pebbles, or from knowing the input word in advance. Indeed, Spoiler's choices are actually independent of Duplicator's moves (if Duplicator does not lose early). In other words, Duplicator loses even if she knows in advance what transition will be played next. Consequently, no two states are in multipebble or fixed-word simulation. This also shows that proxy simulations are incomparable with the latter.

Backward simulation Also, no two different states are backward simulation equivalent. For example, states in the ring are backward incomparable, since different states

? ?

can only reach the initial state via a different number of a's. Also, no state in the ring can simulate s: From configuration $\langle s, q_i\rangle$, it suffices for Spoiler to take transition $s \xleftarrow{a} q_i$, and then we are in the previous case. So, backward simulation quotienting does not help either.

While no previous quotienting method managed to reduce the size of $\mathcal{Q}_k$, we finally show that backward proxy simulations can.

Backward proxy simulations We have observed above that there are no two backward simulation-equivalent states. However, backward simulation is not the identity itself. Indeed, s backward simulates all the states in the ring except q_0:

$$q_1, q_2, \ldots, q_k \sqsubseteq^{\mathrm{bw}} s$$

Indeed, if Spoiler takes any transition $q_{i+1} \xleftarrow{a} q_i$, then Duplicator can reply with $s \xleftarrow{a} q_i$, and similarly if Spoiler goes to s via action b. This gives more power to Duplicator in the delayed proxy simulation game, with the consequence that any two states in the ring are $\tau^{\mathrm{bw+de}}(Id)$-equivalent. To see why, notice that the ability of $\sqsubseteq^{\mathrm{bw}}$-jumping to s before taking a transition effectively adds an edge b between the following pair of states:

$$q_1, q_2, \ldots, q_k \xrightarrow{b} q_0, q_1, \ldots, q_k$$

Could this be a general approach

Let $\approx$ be the equivalence induced by $\tau^{\mathrm{bw+de}}(Id)$. To show $q_i \approx q_j$, we describe how Duplicator can force infinitely many visits to the accepting state q_0. We distinguish two cases.

- In the first case, consider configurations of the form $\langle s, q_j\rangle$. Spoiler has to play action b, and Duplicator takes a jumping b-transition $q_j \sqsubseteq^{\mathrm{bw}} s \xrightarrow{b} q_0$ to q_0, which is accepting. Duplicator stays in q_0 as long as Spoiler plays transition $s \xrightarrow{b} s$. If at any point Spoiler plays transition $s \xrightarrow{b} q_0$, then we are in configuration $\langle q_0, q_0\rangle$, from which Duplicator clearly wins.
- In the second case, consider configurations of the form $\langle q_i, q_j\rangle$, with $i \neq j$. As long as Spoiler plays action a, Duplicator does the same and stays in the ring. In this way, she will eventually visit q_0. Otherwise, if at any point Spoiler plays action b (from some configuration of the form $\langle q_0, q_k\rangle$ or $\langle s, q_k\rangle$), then Duplicator

$Q/\approx$. *For* $x = \text{di}, \text{de}, \text{bw}$,

$$\underbrace{q\ \tau^x(\approx)\ s}_{in\ Q} \iff \underbrace{[q]\ \tau^x(Id)\ [s]}_{in\ Q/\approx}$$

The following lemma say that the i-th automaton after iterated quotienting corresponds to Q quotiented by the i-th equivalence in the hierarchy of Equations (4.2) and (4.3).

Lemma 4.6.7. *For any* $i \geq 0$,

$$Q_i = Q/\approx_i^{\text{fw-}x} \qquad \textit{and} \qquad S_i = Q/\approx_i^{\text{bw-}x}$$

Proof. By induction. For $i = 0$ it holds trivially. For $i \geq 0$, we have

$$\begin{aligned} Q_{i+1} &= S_i/\approx^{\text{fw-}x} \quad \text{(by induction hypothesis)} \\ &= (Q/\approx_i^{\text{bw-}x})/\approx^{\text{fw-}x} \\ &= Q/\approx_{i+1}^{\text{fw-}x} \end{aligned}$$

where the last step follows from Lemma 4.6.6, since computing forward simulation on $Q/\approx_i^{\text{bw-}x}$ is the same as computing $\tau^x(\approx_i^{\text{bw-}x})$ on Q, whose induced equivalence is $\approx_{i+1}^{\text{fw-}x}$. The calculation for S_{i+1} is analogous. □

4.7 Proxy simulations vs Mediated preorder

In this section we compare proxy simulations against another GFQ preorder which has been studied in literature called *mediated preorder* [4]. In Section 4.7.1 we recall the definition of mediated preorder for nondeterministic automata. In Section 4.7.2 we compare in detail quotienting w.r.t. mediated preorder and proxy simulations; in general, the two approaches are incomparable. Finally, in Section 4.7.3 we show how mediated preorder can be interpreted as a variant of proxy simulation.

; before ?

4.7.1 Mediated preorder

Mediated preorder has been originally introduced and studied in the context of alternating Büchi automata [4]. It arises as a combination of direct and backward simulation, in a spirit not unlike jumping simulations. In the context of nondeterministic automata, it can be defined as follows. (With "$\circ$" we denote relational composition: Given two binary relations R_0 and R_1, $(x, z) \in (R_0 \circ R_1)$ iff $\exists y \cdot (x, y) \in R_0 \wedge (y, z) \in R_1$.)

Definition 4.7.1. *Let* $\sqsubseteq^{\text{di}}$ *be direct simulation and let* $\sqsubseteq^{\text{bw}}$ *be backward simulation. A binary relation* R *is a* mediated simulation *iff*

1) $R \subseteq \sqsubseteq^{\text{di}} \circ \sqsupseteq^{\text{bw}}$ *(where* $\sqsupseteq^{\text{bw}}$ *is the transpose of* $\sqsubseteq^{\text{bw}}$*), and*

2) $R \circ \sqsubseteq^{\text{di}} \subseteq R$.

no good style

In other words, if R is a mediated simulation and $q\ R\ s$ for two states q and s, then

1) There exists a state $\hat{q}$ s.t. $q \sqsubseteq^{\text{di}} \hat{q}$ and $s \sqsubseteq^{\text{bw}} \hat{q}$. State $\hat{q}$ is called a *mediator*, and depends in general on q and on s.

2) For any state $\hat{s}$ s.t. $s \sqsubseteq^{\text{di}} \hat{s}$, we have $q\ R\ \hat{s}$.

Mediated simulations are closed under union, and *mediated preorder* $\sqsubseteq^{\text{M}}$ is defined as the union of all mediated simulations, and, therefore, the largest such simulation.

$\sqsubseteq^{\text{M}}$ is correctly called a preorder. First, it is clearly reflexive, since the identity relation is a mediated simulation. Second, is is also transitive: Indeed, the composite relation $\preceq := \sqsubseteq^{\text{M}} \circ \sqsubseteq^{\text{M}}$ is included in $\sqsubseteq^{\text{M}}$. This is established by showing that $\preceq$ is itself a mediated simulation, which can be done with the following calculations: Condition 2) $\preceq \circ \sqsubseteq^{\text{di}} \subseteq \preceq$ follows from

$$\begin{aligned} \preceq \circ \sqsubseteq^{\text{di}} &= (\sqsubseteq^{\text{M}} \circ \sqsubseteq^{\text{M}}) \circ \sqsubseteq^{\text{di}} && \text{by associativity} \\ &= \sqsubseteq^{\text{M}} \circ (\sqsubseteq^{\text{M}} \circ \sqsubseteq^{\text{di}}) && \text{by 2) on } \sqsubseteq^{\text{M}} \\ &\subseteq \sqsubseteq^{\text{M}} \circ \sqsubseteq^{\text{M}} = \preceq \end{aligned}$$

and condition 1) $\preceq \subseteq \sqsubseteq^{\text{di}} \circ \sqsupseteq^{\text{bw}}$ follows from

$$\begin{aligned} \preceq &= \sqsubseteq^{\text{M}} \circ \sqsubseteq^{\text{M}} && \text{by 1) on } \sqsubseteq^{\text{M}} \\ &\subseteq \sqsubseteq^{\text{M}} \circ (\sqsubseteq^{\text{di}} \circ \sqsupseteq^{\text{bw}}) && \text{by associativity} \\ &= (\sqsubseteq^{\text{M}} \circ \sqsubseteq^{\text{di}}) \circ \sqsupseteq^{\text{bw}} && \text{by 2) on } \sqsubseteq^{\text{M}} \\ &\subseteq \sqsubseteq^{\text{M}} \circ \sqsupseteq^{\text{bw}} && \text{by 1) on } \sqsubseteq^{\text{M}} \\ &\subseteq (\sqsubseteq^{\text{di}} \circ \sqsupseteq^{\text{bw}}) \circ \sqsupseteq^{\text{bw}} && \text{by associativity} \\ &= \sqsubseteq^{\text{di}} \circ (\sqsupseteq^{\text{bw}} \circ \sqsupseteq^{\text{bw}}) && \text{since } \sqsupseteq^{\text{bw}} \text{ is transitive} \\ &\subseteq \sqsubseteq^{\text{di}} \circ \sqsupseteq^{\text{bw}} \end{aligned}$$

Finally, mediated preorder is at least as coarse as forward direct simulation. This follows directly from the fact that $\sqsubseteq^{\text{di}}$ is itself a mediated simulation: 1) $\sqsubseteq^{\text{di}} \subseteq \sqsubseteq^{\text{di}} \circ \sqsupseteq^{\text{bw}}$, and 2) $\sqsubseteq^{\text{di}} \circ \sqsubseteq^{\text{di}} \subseteq \sqsubseteq^{\text{di}}$ (by transitivity of $\sqsubseteq^{\text{di}}$). [4] establishes that mediated preorder can be used for quotienting.

Theorem 4.7.2. $\sqsubseteq^{\text{M}}$ *is a GFQ preorder coarser than* $\sqsubseteq^{\text{di}}$.