Simulation Relations for Alternating Büchi Automata

Extended Technical Report

Carsten Fritz* Thomas Wilke†
CAU, Kiel, Germany
{fritz, wilke}@ti.informatik.uni-kiel.de

July 2002

Abstract

We adapt direct, delayed, and fair simulation to alternating Büchi automata. Unlike with nondeterministic Büchi automata, naive quotients do not preserve the recognized language. As a remedy, we present specifically designed definitions of quotients, namely minimax and semi-elective quotients: minimax quotients, which are simple and have a minimum number of transitions, preserve the recognized language when used with direct but not with delayed simulation, while semi-elective quotients, which are more complicated and have more transitions, preserve the recognized language when used with direct simulation as well as delayed simulation. Just as in the case of nondeterministic Büchi automata, fair simulation cannot be used for quotienting. We show that all three types of simulations imply language containment in the sense that if one automaton simulates another automaton with respect to any of the three simulations, then the language recognized by the latter is contained in the language recognized by the former. Our approach is game-theoretic; the proofs rely on a specifically tailored join operation for strategies in simulation games which is interesting in its own right.

Computing all three types of simulation relations and the described quotients is not more difficult than computing the corresponding relations and the naive quotient for nondeterministic Büchi automata. For weak alternating Büchi automata, which are known to recognize all regular ω-languages, we present a particularly efficient algorithm running in time quadratic in the input size.

*Supported by Deutsche Forschungsgemeinschaft.
†Partially supported by Deutsche Forschungsgemeinschaft.

Contents

1 Introduction

An obvious task of theory is to provide reasonable and practically useful notions for comparing automata. For this purpose, simulation relations, which capture the intuitive notion that the moves of one automaton can be mimicked by the moves of another automaton, were introduced and have been used successfully, especially in automated verification. For instance, it is often crucial to check whether the language of a given automaton (describing a system) is contained in the language of another automaton (describing the allowed computations); a sufficient condition for this to hold is that the second automaton simulates the first automaton, and therefore algorithms computing simulation relations are used for checking language containment, see, e. g., [DHW91]. Also, it is often necessary to reduce the state space of a large transition system or automaton (modeling the system or the specification considered) before space and time consuming algorithms are applied; one way to do this is to replace the transition system or automaton in question by a quotient in which states which mutually simulate each other are identified; for this purpose algorithms for computing simulation relations and quotients have been applied as well, see, e. g., [EH00, SB00].

In previous work, simulation relations have been introduced for ordinary and alternating transition systems, see, e. g., [Mil89, HRHK95, AHKV98], and used for checking trace containment. In addition, there is a series of papers studying simulation relations for (nondeterministic) Büchi automata, see, e. g., [HKR97, EH00, ESW01], and nondeterministic ω-automata with other acceptance conditions. In this paper, we combine what has been done for alternating transition systems and nondeterministic Büchi automata: we introduce and study simulation relations for alternating Büchi automata, the motivation being threefold. First, alternation, in general, is a natural and powerful concept, and simulation relations for alternating automata have only been studied for transition systems without acceptance conditions (yet in a more general setting, see [Mil89, HRHK95, AHKV98]). Second, alternating Büchi automata are a generalization of Büchi games (more precisely, two-player infinite games on finite graphs with a Büchi winning condition) in the sense that such a game can be viewed as a Büchi automaton over a one-letter alphabet; thus, simulation relations for alternating Büchi automata cover Büchi games as well. Third, over the last decade, alternating automata have proved to be the right devices to study modal and temporal logics from an automata-theoretic point of view, in particular, new automata-theoretic methods for automated verification based on alternating automata have been developed, see, e. g., [MSS88, Var94, KVW00], so that simulation relations for alternating Büchi automata are of practical interest.

Our definitions of the various simulation relations for alternating Büchi automata are game-based and follow closely the approach of [ESW01]. The main technical difficulty to deal with are the two different types—existential and universal—of states present in alternating automata. Our definitions of the simulation relations are most general with respect to this distinction as we allow that a universal state simulates an existential state and vice versa. This yields smaller automata after quotienting, but, as we prove, does not increase the complexity of the algorithms.

Treating existential and universal states at the same time makes the situation much more complicated. The naive quotient construction, which was also used in [ESW01] for nondeterministic Büchi automata, does not work with alternating Büchi automata. For this reason, we introduce new quotients, which we call minimax and semi-elective quotients and show can replace the naive quotient in the context of alternating Büchi automata: minimax quotients with respect to direct simulation and semi-elective quotients with respect to direct as well as delayed simulation preserve the recognized languages. (For fair simulation, the situation is hopeless, since it was already argued in [ESW01] that no reasonable fair quotient can preserve the recognized language.) We also show that all three types of simulation relations can be used for checking language containment.

Most of our results, especially the more complicated ones, rely on a specific construction to compose strategies in simulation games, which is reminiscent of intruder-in-the-middle attacks known from cryptography. Most of the technical work goes into analyzing this strategy composition method.

Our paper is organized as follows. In Section 2, we review the basic definitions on alternating automata and two-player games on graphs, which are the main tool of the paper. In Section 3, we present our definitions of the various simulation relations and prove that simulation implies language containment. Sections 4 and 5 are the technical core of the paper and lay the ground for proving that direct and delayed quotients preserve the language recognized. In Sections 6 and 7 the definitions of direct and delayed quotient are presented and it is shown that these quotients preserve the language recognized. Section 9 presents efficient algorithms for computing the simulation relations introduced.

2 Alternating Büchi Automata and Games

In this section, we fix basic notation and definitions. We describe the games which all our simulation relations for alternating Büchi automata are based on, and we review the definition of alternating Büchi automata we work with in this paper.

The set of natural numbers is denoted ω. As usual, given a set Σ, we denote the set of finite, finite but nonempty, and infinite sequences over Σ by Σ^*, Σ^+, and Σ^ω, respectively. We set $\Sigma^\infty = \Sigma^* \cup \Sigma^\omega$. Words over Σ are viewed as functions from an initial segment of ω or ω itself to Σ, so when w is a word, then $w(i)$ denotes the letter at its ith position where the first letter is in position 0, and $w[i..j]$ denotes the substring extending from position i through position j.

When R is a binary relation, then uR denotes $\{v \mid (u,v) \in R\}$; similarly, $Rv = \{u \mid (u,v) \in R\}$. When t is an n-tuple, $\mathrm{pr}_i(t)$ is the ith component of t (for $1 \leq i \leq n$).

2.1 Games

For our purposes, a *game* is a tuple

$$G = (P, P_0, P_1, p_I, Z, W) \tag{1}$$

where

- P is the the set of all positions of G,
- $\{P_0, P_1\}$ is a partition of P into the *positions* of Player 0 and Player 1, respectively, where $P_0 = \emptyset$ and $P_1 = \emptyset$ are allowed,
- $p_I \in P$ is the *initial position* of G,
- $Z \subseteq P \times P$ is the set of *moves* of G, and
- $W \subseteq P^\omega$ is the *winning set* of G.

The directed graph (P, Z) is called the *game graph* of G and also denoted by G (with no danger of confusion).

A *play* in G is a maximal path through G starting in p_I; a *partial play* is any path through G starting in p_I. A play $\pi = p_0 p_1 p_2 \ldots$ is winning for Player 1 if π is finite and the last position of π belongs to Player 0 (it is her turn, but she cannot move) or if $\pi \in W$. In all other cases, Player 0 wins the play.

A *strategy for Player 0* is a partial function $\sigma\colon P^*P_0 \to P$ satisfying the following condition for every $\pi \in P^*$ and $p \in P_0$. If $pZ \neq \emptyset$, then $\sigma(\pi p) \in pZ$, else $\sigma(\pi p)$ is undefined. A partial play π is *conform with* σ *(σ-conform)* if for every i such that $i+1 < |\pi|$ and $\pi(i) \in P_0$, we have $\pi(i+1) = \sigma(\pi(0)\ldots\pi(i))$. The strategy σ is a *winning strategy* for Player 0 if every σ-conform play is winning for Player 0. Player 0 *wins* G if he has a winning strategy.— For Player 1, the same notions are defined by exchanging 0 with 1.

Note that if Player 0 plays according to a strategy τ and Player 1 plays according to a strategy σ, the resulting play is completely determined. This play is called the (τ,σ)*-conform* play.

In general, when σ is a strategy, then not all partial plays are σ-conform, which means strategies need not be total functions. In fact, it is enough to require that a strategy for Player 0 need only be defined for all partial plays $\pi \in P^*P_0$ which are σ-conform.

2.2 Alternating Büchi automata

For the purpose of this paper, an alternating Büchi automaton is a tuple

$$A = (Q, \Sigma, q_I, \Delta, E, U, F) \tag{2}$$

where

- Q is a finite *set of states,*
- Σ is a finite *alphabet,*
- $q_I \in Q$ is the *initial state,*
- $\{E, U\}$ is a partition of Q in *existential* and *universal states,* where $E = \emptyset$ and $U = \emptyset$ are allowed, and
- $F \subseteq Q$ is the set of *accepting states*.

Acceptance of alternating Büchi automata is best defined via games. For an alternating Büchi automaton A as above and an ω-word $w \in \Sigma^\omega$, the *word game* $G(A, w)$ is defined as in (1) where

- $P = Q \times \omega$,
- $P_0 = U \times \omega$,
- $P_1 = E \times \omega$,
- $p_I = (q_I, 0)$,
- $Z = \{((s, i), (s', i+1)) \mid (s, w(i), s') \in \Delta\}$, and
- $W = (P^*(F \times \omega))^\omega$.

Following [GH82], in the above game, Player 1 is called *Automaton* while Player 0 is called *Pathfinder*. Acceptance is now defined as follows. The word w is *accepted* by the automaton A if Automaton wins the game $G(A, w)$. The language *recognized* by A is

$$L(A) = \{w \in \Sigma^\omega \mid \text{Automaton wins } G(A, w)\} \ . \tag{3}$$

For $q \in Q$, we will write $A(q)$ for the *translation* of A to q, which is defined to be the same automaton, but with initial state q, i.e., $A(q) = (Q, \Sigma, q, \Delta, E, U, F)$.

In figures, existential states are shown as diamonds and universal states as squares; accepting states have double lines, see, e.g., Figure 1.

3 Simulation Relations for Alternating Büchi Automata

In this section, we define three types of simulation relations for alternating Büchi automata, namely direct, delayed, and fair simulation, which are all based on the same simple game, only the winning condition varies. We show that all these simulations have the property that if an automaton simulates another automaton the language recognized by the latter is contained in the language recognized by the former—we say simulation implies language containment.

3.1 Direct, delayed, and fair simulation

Let $A^0 = (Q^0, \Sigma, p_I, \Delta^0, E^0, U^0, F^0)$ and $A^1 = (Q^1, \Sigma, q_I, \Delta^1, E^1, U^1, F^1)$ be alternating Büchi automata. The *basic simulation game* $G(A^0, A^1)$ goes as follows.

The game is played by two players, *Spoiler* and *Duplicator,* who play the game in rounds. There can be an infinite number of rounds, and each individual round is played as follows. At the beginning of a round, a pair (p,q) of states $p \in Q^0$ and $q \in Q^1$ is given, and the players play as follows.

1. Spoiler chooses a letter $a \in \Sigma$.
2. Spoiler and Duplicator play as follows, depending on the modes of p and q.
   - If $(p,q) \in E^0 \times E^1$, then Spoiler chooses a transition $(p,a,p') \in \Delta^0$ and after that Duplicator chooses a transition $(q,a,q') \in \Delta^1$.
   - If $(p,q) \in U^0 \times U^1$, then Spoiler chooses a transition $(q,a,q') \in \Delta^1$ and after that Duplicator chooses a transition $(p,a,p') \in \Delta^0$.
   - If $(p,q) \in E^0 \times U^1$, then Spoiler chooses transitions $(p,a,p') \in \Delta^0$ and $(q,a,q') \in \Delta^1$.
   - If $(p,q) \in U^0 \times E^1$, then Duplicator chooses transitions $(p,a,p') \in \Delta^0$ and $(q,a,q') \in \Delta^1$.
3. The starting pair for the next round is (p', q').

The first round begins with the pair (p_I, q_I). If, at any point during the course of the game, a player cannot proceed any more, he or she looses (early). When the players proceed as above and no player looses early, they construct an infinite sequence $(p_0,q_0),(p_1,q_1),\ldots$ of pairs of states (with $p_0 = p_I$ and $q_0 = q_I$), and this sequence determines the winner depending on the type of simulation relation we are interested in:

Direct simulation (di): Duplicator wins if for every i with $p_i \in F^0$ we have $q_i \in F^1$.

Delayed simulation (de): Duplicator wins if for every i with $p_i \in F^0$ there exists $j \geq i$ such that $q_j \in F^1$.

Fair simulation (f): Duplicator wins if there are infinitely many j with $q_j \in F^1$ whenever there are infinitely many i with $p_i \in F^0$.

In all other cases, Spoiler wins.

The games above can formally be described as follows, using the notion of game from the previous section. Spoiler takes over the role of Player 0, while Duplicator takes over the role of Player 1. The positions in the game reflect the status of a round. We have positions of the form (p,q) for the starting point of a round. We have positions of the form (p,q,a,S,b,S',b') which represent the fact that the round started out in (p,q), Spoiler chose the letter a, player S (Spoiler or Duplicator) first has to pick a transition in A^b and after that player S' has to pick a position in $A^{b'}$. Finally, we have positions of the form (p,q,a,S',b') which represent the fact that Spoiler chose the letter a, and player S' still has to pick a transition in $A^{b'}$. That is, in the formal definition of the game, we use

$$U_s = Q^0 \times Q^1 \times \Sigma \times \{s\} \times \{0,1\} \times \{s,d\} \times \{0,1\} \ , \tag{4}$$
$$U_d = Q^0 \times Q^1 \times \Sigma \times \{d\} \times \{0,1\} \times \{s,d\} \times \{0,1\} \ , \tag{5}$$
$$V_s = Q^0 \times Q^1 \times \Sigma \times \{s\} \times \{0,1\} \ , \tag{6}$$
$$V_d = Q^0 \times Q^1 \times \Sigma \times \{d\} \times \{0,1\} \ . \tag{7}$$

Given a type $x \in \{di, de, f\}$, the game $G^x(A^0,A^1)$ is defined by

$$G^x(A^0,A^1) = (P, P_0, P_1, (p_I, q_I), Z, W^x) \tag{8}$$

where

$$P = (Q^0 \times Q^1) \cup U_s \cup U_d \cup V_s \cup V_d \ , \tag{9}$$
$$P_0 = (Q^0 \times Q^1) \cup U_s \cup V_s \ , \tag{10}$$
$$P_1 = U_d \cup V_d \ , \tag{11}$$

the set $Z \subseteq P \times P$ contains all moves of the form

$$
\begin{aligned}
&((p,q),(p,q,a,s,0,d,1)) \ , && \text{for } p \in E^0, q \in E^1, a \in \Sigma \ , && (12)\\
&((p,q),(p,q,a,s,0,s,1)) \ , && \text{for } p \in E^0, q \in U^1, a \in \Sigma \ , && (13)\\
&((p,q),(p,q,a,d,0,d,1)) \ , && \text{for } p \in U^0, q \in E^1, a \in \Sigma \ , && (14)\\
&((p,q),(p,q,a,s,1,d,0)) \ , && \text{for } p \in U^0, q \in U^1, a \in \Sigma \ , && (15)\\
&((p,q,a,x,0,y,1),(p',q,a,y,1)) \ , && \text{for } (p,a,p') \in \Delta^0, x,y \in \{s,d\} \ , && (16)\\
&((p,q,a,s,1,d,0),(p,q',a,d,0)) \ , && \text{for } (q,a,q') \in \Delta^1 \ , && (17)\\
&((p,q,a,d,0),(p',q)) \ , && \text{for } (p,a,p') \in \Delta^0 \ , && (18)\\
&((p,q,a,x,1),(p,q')) \ , && \text{for } (q,a,q') \in \Delta^1, x \in \{s,d\} \ . && (19)
\end{aligned}
$$

Note that not all positions are reachable from the initial position of the game or from any position in $(Q^0 \times Q^1)$. These unreachable positions can be removed (cf. Section 9), but this would make the proofs somewhat more complicated, so we keep them.

The winning condition depends on the type of simulation relation (see above). To phrase it concisely, we will use the following notation. We will write $\hat{F}^0$ for the set of all positions with an element from F^0 in the first component and $\hat{F}^1$ for the set of all positions with an element from F^1 in the second component. Also, we will write $\bar{F}^0$ and $\bar{F}^1$ for $P \setminus \hat{F}^0$ and $P \setminus \hat{F}^1$. Now we can state the winning conditions formally:

The *direct winning condition* is $W^{di} = (\bar{F}^0 \cup \hat{F}^1)^\omega$.

The *delayed winning condition* is $W^{de} = P^\omega \setminus P^*(\hat{F}^0 \cap \bar{F}^1)(\bar{F}^1)^\omega$.

The *fair winning condition* is $W^f = P^\omega \setminus P^*((\hat{F}^0 \cap \bar{F}^1)(\bar{F}^1)^*)^\omega$, respectively.

For $x \in \{di, de, f\}$, we define a relation $\leq_x$ on alternating Büchi automata. We write $A \leq_x B$ when Duplicator has a winning strategy in $G^x(A,B)$ and say that B *x-simulates* A. For states p of A, q of B, we write $p \leq_x q$ to indicate that $B(q)$ x-simulates $A(p)$. We write $G^x(p,q)$ instead of $G^x(A(p),B(q))$.

As an example for a simulation game, consider the automaton A given in Figure 1, which we view as an automaton over the alphabet $\{a,b\}$.

We argue that the games $G^{de}(0,1), G^{de}(1,0), G^f(0,1)$ and $G^f(1,0)$ are a win for Duplicator. To see this, consider the strategy σ defined by

$$
\begin{aligned}
\sigma(P^*(0,2,b,d,0)) &= (2,2) \ , && && (20)\\
\sigma(P^*(1,y,b,d,0)) &= (2,y) \ , && \text{for } y = 1,2 \ , && (21)\\
\sigma(P^*(2,2,b,d,1)) &= (2,2) \ . && && (22)
\end{aligned}
$$

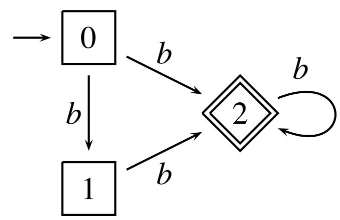

Figure 1: alternating Büchi automaton

In a play starting in position $(0,1)$, Spoiler has to choose the letter b, or he loses early, and he has to choose the transition $(1,b,2)$, i.e., the play reaches position $(0,2,b,d,0)$. Using σ, Duplicator now chooses the transition $(0,b,2)$, and the next round starts in position $(2,2)$. Spoiler now always has to choose the letter b and the transition $(2,b,2)$ (or he loses early), but Duplicator (using σ) always chooses the same transition, so the play stays in $(2,2)$ and is thus a win for Duplicator.

If the play starts in $(1,0)$, the strategy σ also ensures that a play is either an early lose for Spoiler or eventually stays in $(2,2)$.

Lemma 1 *The following relations hold between the three types of simulation relations:*

$$\leq_{di} \subsetneq \leq_{de} \subsetneq \leq_f \quad . \tag{23}$$

Proof. Since $W^{di} \subseteq W^{de} \subseteq W^f$, the inclusions follow immediately. These inclusions are strict, for consider the automata A^0 and A^1 defined by

$$A^0 = (\{q_0,q_1\},\{a\},q_0,\{(q_i,a,q_1) \mid i \in \{0,1\}\},\{q_0,q_1\},\emptyset,\{q_1\}) \ , \tag{24}$$

$$A^1 = (\{q_2,q_3\},\{a\},q_2,\{(q_i,a,q_3) \mid i \in \{2,3\}\},\{q_2,q_3\},\emptyset,\{q_2\}) \ . \tag{25}$$

We have $q_1 \leq_{de} q_0$, but $q_1 \not\leq_{di} q_0$, and $q_2 \leq_f q_3$, but $q_2 \not\leq_{de} q_3$. □

We say that an alternating Büchi automaton as in (2) is *complete* if for every $q \in Q$, $a \in \Sigma$, there is a $q' \in Q$ such that $(q,a,q') \in \Delta$. Clearly, if we are given two alternating Büchi automata A and B such that $A \leq_x B$ for some $x \in \{x,di,de,f\}$, then, by adding at most two new states and at most $(|\Sigma| \cdot (|Q|+2))$ many transitions, we can turn A and B into equivalent complete automata A and B such that $A \leq_x B$ still holds. Therefore, we henceforth assume that all automata are complete; we allow incomplete automata only in Section 9, where we study algorithms for computing simulation relations, and in examples, which we want to keep small.

3.2 Simulation implies language containment

The first theorem states that all types of simulation imply language containment:

Theorem 1 *Let $x \in \{di, de, f\}$ and A^0 and A^1 be alternating Büchi automata. If $A^0 \leq_x A^1$, then $L(A^0) \subseteq L(A^1)$.*

Before we turn to the proof, we introduce useful conventions and notations concerning plays of simulation games.
Formally, a play of a simulation game is an infinite sequence $T = t_0 t_0^0 t_0^1 t_1 t_1^0 t_1^1 \ldots$ where $t_i \in Q^0 \times Q^1$, $t_i^0 \in U_s \cup U_d$ and $t_i^1 \in V_s \cup V_d$. But the play T is obviously completely determined by the infinite sequence $t_0 t_1 \ldots$ and the word $w \in \Sigma^\omega$ that is the sequence of letters in the third components of the t_i^0's—recall that each t_i^0 is of the form (p, q, a, S, b, S', b') where a is a letter from Σ. A similar statement holds true for a partial play ending in a position in $Q^0 \times Q^1$: such a partial play is also determined by its subsequence of $Q^0 \times Q^1$-positions and the respective sequence of letters. That is, there is a natural partial mapping

$$\xi \colon (Q^0 \times Q^1)^\infty \times \Sigma^\infty \to \text{set of } G^x(A^0, A^1)\text{-plays} \ , \tag{26}$$

which maps $((p_i, q_i)_{i<n}, w)$ (where $n \in \omega \cup \{\omega\}$) to the corresponding partial play, provided there is such a play. This is the case if

1. $|w| + 1 = n$ and
2. for all i with $i + 1 < n$, $(p_i, w(i), p_{i+1}) \in \Delta^0$ and $(q_i, w(i), q_{i+1}) \in \Delta^1$.

An element of the domain of ξ will be called a *protoplay*.

Proof of Theorem 1. Let A^0 and A^1 be as above and let σ be a winning strategy for Duplicator in $G^x(A^0, A^1)$. Let $w \in L(A^0)$, and let σ' be a winning strategy for Automaton in $G(A^0, w)$. We have to show that Automaton has a winning strategy σ'' in $G(A^1, w)$.

While playing $G(A^1, w)$, Automaton (in $G(A^1, w)$) simultaneously plays the game $G^x(A^0, A^1)$ and the game $G(A^0, w)$. In both plays he makes the moves for both players, Spoiler and Duplicator as well as Automaton and Pathfinder, and uses σ and σ' to determine their moves. In other words, Automaton works as a puppeteer and moves four puppets at the same time. In this spirit, Automaton and Pathfinder in $G(A^0, w)$ and Spoiler and Duplicator in $G^x(A^0, A^1)$ will be called the automaton puppet, the pathfinder puppet, the spoiler puppet, and the duplicator puppet, respectively.

Automaton plays in a way such that after each round

- $\circ$ the state components of $G(A^0, w)$ and $G(A^1, w)$ agree with the two state components of $G^x(A^0, A^1)$, and
- $\circ$ the partial games in $G^x(A^0, A^1)$ and $G(A^0, w)$ are conform with σ and σ'.

Then, clearly, since σ and σ' are winning, in the emerging plays in $G(A^1, w)$ infinitely many states will be in F, that is, Automaton wins $G(A^1, w)$.

The above can be achieved easily:

- $\circ$ In $G^x(A^0, A^1)$, Automaton uses σ to determine the moves of the duplicator puppet.
- $\circ$ In $G(A^0, w)$, Automaton uses σ' to determine the moves of the automaton puppet.
- $\circ$ In $G^x(A^0, A^1)$, Automaton moves the spoiler puppet as follows.

  If the spoiler puppet needs to choose a letter, Automaton chooses the respective letter from the given word w.

  If the spoiler puppet needs to choose a state in A^1, then this is because the current state from A^1 is universal, which means Pathfinder goes in $G(A^1, w)$. Automaton lets the spoiler puppet mimic this move in $G^x(A^0, A^1)$.

  If the spoiler puppet needs to choose a state in A^1, then this is because the current state from A^1 is existential, which means it is the automaton puppet's turn in $G(A^0, w)$. In $G(A^0, w)$, he lets the automaton puppet move according to σ' (see above) and in $G^x(A^0, A^1)$, he lets the spoiler puppet mimic this move.
- $\circ$ In $G(A^0, w)$, if it is the pathfinder puppet's turn, then this is because the current state in A^1 is universal, which means the duplicator puppet's goes in $G^x(A^0, A^1)$. In $G^x(A^0, A^1)$, he lets the duplicator puppet move according to σ (see above) and in $G(A^0, w)$, he lets the pathfinder puppet mimic this move.

We now proceed with a formal treatment. In order to define the winning strategy σ'' of Automaton in $G(A^1, w)$, we need partial functions pr^0 and pr^1 mapping partial $G^x(A^0, A^1)$-protoplays to prefixes of $G(A^0, w)$-plays and $G(A^1, w)$-plays, respectively. For any partial $G^x(A^0, A^1)$-protoplay $T = ((p_i, q_i)_{i \leq n}, w[0..n-1])$, we set

$$\mathrm{pr}^0(T) = (p_0, 0) \ldots (p_n, n) \qquad \text{and} \qquad \mathrm{pr}^1(T) = (q_0, 0) \ldots (q_n, n) \ . \tag{27}$$

We define σ'' as follows. If $T = ((p_i, q_i)_{i \le n+1}, w[0..n])$ is a partial σ-conform protoplay such that $\mathrm{pr}^0(T)$ is σ'-conform and such that $q_n \in E^1$, we set

$$\sigma''((q_0, 0) \dots (q_n, n)) = (q_{n+1}, n+1). \tag{28}$$

We show that (1) this function is well-defined, (2) this function is a strategy for Automaton, and (3) this strategy is winning.

(1) First, assume $T = ((p_0, q_0) \dots (p_n, q_n), w[0..n-1])$ is some partial σ-conform protoplay such that $(p_0, 0) \dots (p_n, n)$ is σ'-conform. We show that for every $q \in Q^1$, there is at most one $p \in Q^0$ such that $((p_0, q_0) \dots (p_n, q_n)(p, q), w[0..n])$ is a partial σ-conform protoplay and $(p_0, 0) \dots (p_n, n)(p, n+1)$ is σ'-conform. If $p_n \in E^0$, then σ' determines p, and if $p_n \in U^0$, then σ determines p. We can now show σ'' is well-defined.

Assume there are two protoplays $T = ((p_0, q_0) \dots (p_n, q_n)(p, q), w[0..n])$ and $\hat{T} = ((\hat{p}_0, q_0) \dots (\hat{p}_n, q_n)(\hat{p}, \hat{q}), w[0..n])$ such that $q \neq \hat{q}$, $q_n \in E^1$ and such that $\mathrm{pr}^0(T)$ and $\mathrm{pr}^0(\hat{T})$ are σ'-conform. Then, by the above argument, $p_i = \hat{p}_i$ for $i \le n$. But since $q_n \in E^1$, $q = \hat{q}$ is determined by σ.

(2) We have to show that the domain of σ'' contains all σ''-conform plays $T = (q_0, 0) \dots (q_n, n)$ with $q_n \in E^1$. We prove this by induction. If $q_i \in U^1$ for $i < n$, then σ does not impose any restriction on any of the q_i for $i \le n$, so t is in the domain of σ''. If there is some $i < n$ such that $q_i \in E^1$, we argue as follows. Assume i is maximal with this property. Then, by induction hypothesis, $(q_0, 0) \dots (q_i, i)$ is in the domain of σ''. Since σ does not impose any restriction on q_j for $i < j \le n$, T is also in the domain of σ''.

(3) Assume V is some σ''-conform play. Then, by construction, there is a σ-conform protoplay $T = ((p_0, q_0)(p_1, q_1) \dots, w)$ such that $\mathrm{pr}^1(T) = V$ and $\mathrm{pr}^0(T)$ is σ'-conform. Since σ' was assumed to be winning, we know that there exist infinitely many i with $p_i \in F^0$. Since σ was also assumed to be winning, there exist infinitely many i' with $q_{i'} \in F^1$. This shows V is a win for Automaton. □

4 Composing Simulation Strategies

In this section, let $x \in \{di, de, f\}$. We will introduce the join of two Duplicator strategies, a concept fundamental for the proofs of the results in Sections 5 and 7. The idea is that two strategies for simulation games starting in positions (q^0, q^1) and (q^1, q^2), respectively, can be merged to a joint strategy for a game starting in (q^0, q^2); this joint strategy inherits crucial properties of the two original strategies (see Lemmas 3 and 4).

Let $k \in Q^0$, $p \in Q^1$, $q \in Q^2$. Let σ_0 be a Duplicator strategy for the basic game $G(k, p)$, and let σ_1 be a Duplicator strategy for the basic game $G(p, q)$.

To describe the join of the strategies σ_0 and σ_1, denoted $\sigma_0 \bowtie \sigma_1$, informally, we can again use the puppeteering metaphor of the previous section: Duplicator, playing $G(k,q)$ using $\sigma_0 \bowtie \sigma_1$, simultaneously plays $G(k,p)$ and $G(p,q)$. His four puppets are the Spoiler and the Duplicator of these games. We will call the Spoiler and the Duplicator of $G(k,p)$ the left spoiler puppet and the left duplicator puppet, while the Spoiler and Duplicator of $G(p,q)$ are the right spoiler puppet and the right duplicator puppet.

Duplicator (of $G(k,q)$, our puppeteer) plays in such a way that after each round

- the first state component of $G(k,p)$ and the second state component of $G(p,q)$ agree with the first and second state component of $G(k,q)$, respectively, and
- the second state component of $G(k,p)$ agrees with the first state component of $G(p,q)$, and
- the partial plays in $G(k,p)$ and $G(p,q)$ are conform with σ_0 and σ_1, respectively.

This can be achieved in the following way:

- In $G(k,p)$, Duplicator uses σ_0 to determine the moves of the left duplicator puppet, while
- in $G(p,q)$, he uses σ_1 to determine the moves of the right duplicator puppet.
- If one of the spoiler puppets has to choose a letter, it chooses the letter that the Spoiler of $G(k,q)$ has chosen in this round.
- If the left spoiler puppet has to choose a transition of Δ^0, Spoiler has to choose such a transition in this round, too (the identical first state components of $G(k,p)$ and $G(k,q)$ are existential). The left spoiler puppet mimics this choice.

  If the left spoiler puppet has to choose a transition of Δ^1, this is because the second state component is universal. Since this component agrees with the first state component in $G(p,q)$, the right duplicator puppet also has to choose such a transition. Duplicator lets the left spoiler puppet mimic the move of the right duplicator puppet. This move of the right duplicator puppet is done according to σ_1 (see above).
- If the right spoiler puppet has to choose a transition of Δ^2, Spoiler also has to choose such a transition and his move is mimicked by the right spoiler puppet (the second state components of $G(p,q)$ and $G(k,q)$ are identical; in this case, they are universal).

If the right spoiler puppet has to choose a transition of Δ^1, this is because the first state component is existential. Since this component agrees with the second state component in $G(k,p)$, the left duplicator puppet also has to choose such a transition. Duplicator lets the right spoiler puppet mimic the move of the left duplicator puppet, and the move of this puppet is determined by σ_0 (see above).

To define this strategy formally, we also have to keep track of the sequence of second state components of $G(k,p)$, which is the sequence of first state components of $G(p,q)$. We now continue with the formal definitions.

We simultaneously and inductively define the *joint strategy* $\sigma_0 \bowtie \sigma_1$, a Duplicator strategy for $G(k,q)$, and a sequence of Q^1-states (starting with p) for partial $(\sigma_0 \bowtie \sigma_1)$-conform $G(k,q)$-plays, the so-called *intermediate p-sequence*.

The definition (construction) of the joint strategy $\sigma_0 \bowtie \sigma_1$ for the prefix of a play that has lasted for n rounds uses the intermediate p-sequence of length $n+1$ for this prefix, and in turn the $(n+1)$th $(\sigma_0 \bowtie \sigma_1)$-conform round defines the $(n+2)$th element of the intermediate p-sequence for the prolonged prefix.

The definition will have the following property.

Property 1 *If* $((k_j,q_j)_{j<n+1},w)$ *is a partial* $(\sigma_0 \bowtie \sigma_1)$*-conform protoplay and* $(p_j)_{j<n+1}$ *is the intermediate p-sequence for this protoplay, then* $((k_j,p_j)_{j<n+1},w)$ *is a partial* σ_0*-conform* $G(k,p)$*-protoplay and* $((p_j,q_j)_{j<n+1},w)$ *is a partial* σ_1*-conform protoplay.*

Initially, for the $G(k,q)$-protoplay $((k,q),\varepsilon)$ (i.e., for the prefix of the play where no moves have been played), the intermediate p-sequence is p. Note that Property 1 is fulfilled.

Now assume that for a $(\sigma_0 \bowtie \sigma_1)$-conform protoplay $T = ((k_i,q_i)_{i<n+1},w)$, the intermediate p-sequence is given by $(p_i)_{i<n+1}$ (and $k_0 = k, p_0 = p, q_0 = q$). In particular, T and $(p_i)_{i<n+1}$ have Property 1. Let $T' = ((k_i,p_i)_{i<n+1},w)$ and $T'' = ((p_i,q_i)_{i<n+1},w)$. Recall that the last position of $\xi(T)$ is (k_n,q_n).

In order to define $\sigma_0 \bowtie \sigma_1$ and p_{n+1} for the round following T, we distinguish eight cases depending on the modes of k_n, p_n, and q_n.

Case EEE, $(k_n,p_n,q_n) \in E^0 \times E^1 \times E^2$. Assume Spoiler chooses the $G(k,q)$-positions $t_n^0 = (k_n,q_n,a,s,0,d,1)$ and $t_n^1 = (k_{n+1},q_n,a,d,1)$. Let

$$\sigma_0(\xi(T')(k_n,p_n,a,s,0,d,1)(k_{n+1},p_n,a,d,1)) = (k_{n+1},p_{n+1}) \ , \tag{29}$$

$$\sigma_1(\xi(T'')(p_n,q_n,a,s,0,d,1)(p_{n+1},q_n,a,d,0)) = (p_{n+1},q_{n+1}) \ . \tag{30}$$

We define

$$\sigma_0 \bowtie \sigma_1(\xi(T)t_n^0 t_n^1) = (k_{n+1},q_{n+1}) \tag{31}$$

and define $(p_i)_{i\leq n+1}$ to be the intermediate p-sequence for the partial protoplay $((k_i,q_i)_{i\leq n+1}, wa)$; note that the two have Property 1.

Case EUE, $(k_n,p_n,q_n)\in E^0\times U^1\times E^2$. Assume Spoiler chooses the $\xi(T)$-positions $t_n^0=(k_n,q_n,a,s,0,d,1)$ and $t_n^1=(k_{n+1},q_n,a,d,1)$. Let

$$\sigma_1(\xi(T'')(p_n,q_n,a,d,0,d,1))=(p_{n+1},q_n,a,d,1)\ , \tag{32}$$
$$\sigma_1(\xi(T'')(p_n,q_n,a,d,0,d,1)(p_{n+1},q_n,a,d,1))=(p_{n+1},q_{n+1}). \tag{33}$$

We define

$$\sigma_0\bowtie\sigma_1(\xi(T_n)t_n^0t_n^1)=(k_{n+1},q_{n+1}) \tag{34}$$

and $(p_i)_{i\leq n+1}$ as the corresponding intermediate p-sequence.

Case UEU, $(k_n,p_n,q_n)\in U^0\times E^1\times U^2$. Assume that the following $G(k,q)$-positions are $t_n^0=(k_n,q_n,a,s,1,d,0)$ and $t_n^1=(k_n,q_{n+1},a,d,0)$. Let

$$\sigma_0(\xi(T')(k_n,p_n,a,d,1,d,0))=(k_{n+1},p_n,a,d,1)\ , \tag{35}$$
$$\sigma_0(\xi(T')(k_n,p_n,a,d,1,d,0)(k_{n+1},p_n,a,d,1))=(k_{n+1},p_{n+1}). \tag{36}$$

We define

$$\sigma_0\bowtie\sigma_1(\xi(T_n)t_n^0t_n^1)=(k_{n+1},q_{n+1}) \tag{37}$$

and $(p_i)_{i\leq n+1}$ as the corresponding intermediate p-sequence.

Case UUU, $(k_n,p_n,q_n)\in U^0\times U^1\times U^2$. Assume that the following $G(k,q)$-positions are $t_n^0=(k_n,q_n,a,s,1,d,0)$ and $t_n^1=(k_n,q_{n+1},a,d,0)$. Let

$$\sigma_1(\xi(T'')(p_n,q_n,a,s,1,d,0)(p_n,q_{n+1},a,d,0))=(p_{n+1},q_{n+1})\ , \tag{38}$$
$$\sigma_0(\xi(T')(k_n,p_n,a,s,1,d,0)(k_n,p_{n+1},a,d,0))=(k_{n+1},p_{n+1})\ . \tag{39}$$

We define

$$\sigma_0\bowtie\sigma_1(\xi(T)t_n^0t_n^1)=(k_{n+1},q_{n+1}) \tag{40}$$

and $(p_i)_{i\leq n+1}$ as the next intermediate p-sequence.

Case UEE, $(k_n,p_n,q_n)\in U^0\times E^1\times E^2$. Assume Spoiler chooses the position $t_n^0=(k_n,q_n,a,d,0,d,1)$. Let

$$\sigma_0(\xi(T')(k_n,p_n,a,d,0,d,1))=(k_{n+1},p_n,a,d,1)\ , \tag{41}$$
$$\sigma_0(\xi(T')(k_n,p_n,a,d,0,d,1)(k_{n+1},p_n,a,d,1))=(k_{n+1},p_{n+1})\ , \tag{42}$$
$$\sigma_1(\xi(T'')(p_n,q_n,a,s,0,d,1)(p_{n+1},q_n,a,d,1))=(p_{n+1},q_{n+1}). \tag{43}$$

We define

$$\sigma_0 \bowtie \sigma_1(T_n t_n^0) = (k_{n+1}, q_n, a, d, 1) \quad , \tag{44}$$

$$\sigma_0 \bowtie \sigma_1(T_n t_n^0(k_{n+1}, q_n, a, d, 1)) = (k_{n+1}, q_{n+1}) \quad , \tag{45}$$

and choose $(p_i)_{i \leq n+1}$ as the corresponding intermediate p-sequence.

Case UUE, $(k_n, p_n, q_n) \in U^0 \times U^1 \times E^2$, and the following Spoiler-chosen $\xi(T_n)$-position is $t_n^0 = (k_n, q_n, a, d, 0, d, 1)$. Let

$$\sigma_1(\xi(T'')(p_n, q_n, a, d, 0, d, 1)) = (p_{n+1}, q_n, a, d, 1) \quad , \tag{46}$$

$$\sigma_1(\xi(T'')(p_n, q_n, a, d, 0, d, 1)(p_{n+1}, q_n, a, d, 1)) = (p_{n+1}, q_{n+1}) \quad , \tag{47}$$

$$\sigma_0(\xi(T')(k_n, p_n, a, s, 1, d, 0)(k_n, p_{n+1} a, d, 0)) = (k_{n+1}, p_{n+1}) \quad . \tag{48}$$

We define

$$\sigma_0 \bowtie \sigma_1(T_n t_n^0) = (k_{n+1}, q_n, a, d, 1) \quad , \tag{49}$$

$$\sigma_0 \bowtie \sigma_1(T_n t_n^0(k_{n+1}, q_n, a, d, 1)) = (k_{n+1}, q_{n+1}) \quad , \tag{50}$$

and choose $(p_i)_{i \leq n+1}$ as the corresponding intermediate p-sequence.

Case EEU, $(k_n, p_n, q_n) \in E^0 \times E^1 \times U^2$. Assume Spoiler chooses the $G(k, q)$-positions $t_n^0 = (k_n, q_n, a, s, 0, s, 1)$ and $t_n^1 = (k_{n+1}, q_n, a, s, 1)$ and $t_{n+1} = (k_{n+1}, q_{n+1})$. Let

$$\sigma_0(\xi(T')(k_n, p_n, a, s, 0, d, 1)(k_{n+1}, p_n, a, d, 1)) = (k_{n+1}, p_{n+1}) \quad . \tag{51}$$

We define $(p_i)_{i \leq n+1}$ as the corresponding intermediate p-sequence (the strategy $\sigma_0 \bowtie \sigma_1$ need not be defined in this case, since Duplicator cannot move in a turn starting with a $E^0 \times U^2$-state).

Case EUU, $(k_n, p_n, q_n) \in E^0 \times U^1 \times U^2$, and the following Spoiler-chosen $\xi(T_n)$-positions are the three positions defined by $t_n^0 = (k_n, q_n, a, s, 0, s, 1)$, $t_n^1 = (k_{n+1}, q_n, a, s, 1)$ and $t_{n+1} = (k_{n+1}, q_{n+1})$. Let

$$\sigma_1(\xi(T'')(p_n, q_n, a, s, 1, d, 0)(p_n, q_{n+1}, a, d, 0)) = (p_{n+1}, q_{n+1}). \tag{52}$$

We define $(p_i)_{i \leq n+1}$ as the next intermediate p-sequence (again, $\sigma_0 \bowtie \sigma_1$ need not be defined).

This completes the description of $\sigma_0 \bowtie \sigma_1$. It will be thoroughly analyzed in the next section.

5 Fundamental Properties of Simulation Relations and Composed Strategies

In this section, let $x \in \{di, de, f\}$. We will show crucial properties of the simulation relation $\leq_x$.

Fundamental for the study of $\leq_x$ is the following lemma, which is similar to [ESW01, Lemma 4.1].

Lemma 2 *Let A^0, A^1 be alternating Büchi automata and let p,q be states of A^0 and A^1, respectively, such that $p \leq_x q$. Let $a \in \Sigma$.*

1. *If $(p,q) \in E^0 \times E^1$, there is, for every $p' \in \Delta^0(p,a)$, a $q' \in \Delta^1(q,a)$ such that $p' \leq_x q'$.*
2. *If $(p,q) \in E^0 \times U^1$, for all $p' \in \Delta^0(p,a)$ and for all $q' \in \Delta^1(q,a)$ we have $p' \leq_x q'$.*
3. *If $(p,q) \in U^0 \times E^1$, there are a $p' \in \Delta^0(p,a)$ and a $q' \in \Delta^1(q,a)$ such that $p' \leq_x q'$.*
4. *If $(p,q) \in U^0 \times U^1$, there is, for every $q' \in \Delta^1(q,a)$, a $p' \in \Delta^0(p,a)$ such that $p' \leq_x q'$.*

Proof. First, let $(p,q) \in E^0 \times E^1$. Since $p \leq_x q$, in a play T of $G^x(p,q)$ starting with $T_0 = (p,q)(p,q,a,s,0,d,1)(p',q,a,d,1)$, i.e., $p' \in \Delta(p,a)$, Duplicator can use a winning strategy σ. Let $(p',q') = \sigma(T_0)$. Since σ is a winning strategy for Duplicator, there is a winning strategy of Duplicator for $G^x(p',q')$, thus $p' \leq_x q'$.

Similar arguments yield the claims for the other three cases, i.e., the case $(p,q) \in U^0 \times U^1$ is symmetric, while the arguments for the other cases are as follows. Case $(p,q) \in E^0 \times U^1$: If Duplicator cannot move in a round but has a winning strategy at the beginning of that round, he also has a winning strategy at the beginning of the next round, no matter what Spoiler does. Case $(p,q) \in U^0 \times E^1$: If Duplicator has a winning strategy and can choose both transitions, he can choose the transitions using his winning strategy. Then he has a winning strategy at the beginning of the next round. $\square$

We now want to show that $\leq_x$ is reflexive and transitive, i.e., a preorder. Reflexivity is obvious: whenever in a play a position $(q,q) \in E \times E$ is reached, Duplicator can move in the second component to the state that Spoiler has chosen in the first component; for $(q,q) \in U \times U$, he does the same in the first component (Duplicator literally duplicates Spoiler's moves). Using this strategy, Duplicator wins the game in all three versions.

Transitivity needs some more care. We will show this by using the join of two Duplicator strategies, as defined in Section 4.

We can easily verify the following.

Lemma 3 (composing winning strategies) *Let $k \in Q^0$, $p \in Q^1$, and $q \in Q^2$ such that $k \le_x p$ and $p \le_x q$. Let σ_0 be a Duplicator strategy for $G^x(k,p)$, and let σ_1 be a Duplicator strategy for $G^x(p,q)$.*

If σ_0 and σ_1 are winning strategies, $\sigma_0 \bowtie \sigma_1$ is a winning strategy (i.e., $k \le_x p$ and $p \le_x q$ imply $k \le_x q$).

Proof. Let σ_0, σ_1 be winning strategies, and let T be a $(\sigma_0 \bowtie \sigma_1)$-conform play with intermediate p-sequence $(p_i)_{i<\omega}$. Note that the plays T' and T'' (as defined in Section 4) are σ_0-conform and σ_1-conform, respectively.

In the case of direct simulation, since T' is σ_0-conform, for every i such that $k_i \in F^0$, we have $p_i \in F^1$. And since T'' is σ_1-conform, this implies $q_i \in F^2$, that is, T is a win for Duplicator.

In the case of delayed simulation, for every i such that $k_i \in F^0$, there is a $j_0 \ge i$ such that $p_{j_0} \in F^1$, since T' is σ_0-conform. And in turn, by the σ_1-conformity of T'', there is a $j_1 \ge j_0$ such that $q_{j_1} \in F^2$. Hence T is a win for Duplicator.

Finally, for fair simulation, if there are infinitely many i such that $k_i \in F^0$, the σ_0-conformity of T' ensures that there are also infinitely many j such that $p_j \in F^1$, and the σ_1-conformity of T'' ensures that there are infinitely many l such that $q_l \in F^2$. So again T is a win for Duplicator. □

In the sequel, we will call a Duplicator strategy σ for a game $G(p_0,q_0)$ $\le_x$-*respecting* if $p \le_x q$ holds true for every position reachable in any play where Duplicator follows σ.

The following is easy to see:

Remark 1 *A winning strategy of Duplicator for an x-simulation game is $\le_x$-respecting.*

The converse is false for $x \in \{de, f\}$, as we will see later.

Lemma 4 (composing $\le_x$-respecting strategies) *If σ_0 and σ_1 are $\le_x$-respecting strategies, then $\sigma_0 \bowtie \sigma_1$ is a $\le_x$-respecting strategy.*

Proof. Let τ be a Spoiler strategy for $G^x(k,q)$, and let $T = ((t_j)_{j<\omega}, w)$ be the $(\tau, \sigma_0 \bowtie \sigma_1)$-conform protoplay. We have $k \le_x p \le_x q$, hence $k \le_x q$, by Lemma 3.

Now let $i \in \omega$, and $T_i = ((t_j)_{j \le i}, w[0..i-1])$ be the prefix of T of length $i+1$. Let $t_i = (k_i, q_i)$, and let $(p_j)_{j \le i}$ be the intermediate p-sequence of T_i. Assume $k_i \le_x p_i \le_x q_i$.

We show that $k_{i+1} \le_x p_{i+1} \le_x q_{i+1}$ holds for the next $(Q^0 \times Q^2)$-position $t_{i+1} = (k_{i+1}, q_{i+1})$ of T and the next state of the intermediate p-sequence, distinguishing four cases:

In the first case, let $(k_i,q_i) \in U^0 \times U^2$. Let $t_i^0 := \tau(\xi(T_i)) = (k_i,q_i,a,s,1,d,0)$ and $t_i^1 := \tau(\xi(T_i)t_i^0) = (k_i,q_{i+1},a,d,0)$. Let $\sigma_0 \bowtie \sigma_1(\xi(T_i)t_i^0t_i^1) = (k_{i+1},q_{i+1})$, and p_{i+1} be the next state of the intermediate p-sequence according to Section 4.

If $p_i \in E^1$, the definition of $\sigma_0 \bowtie \sigma_1$ implies $k_{i+1} \leq_x p_{i+1}$, since $\xi(T')$ is σ_0-conform (both k_{i+1} and p_{i+1} are chosen according to σ_0). And $p_{i+1} \leq_x q_{i+1}$ by Lemma 2, since $p_i \leq_x q_i$ and $(p_i,q_i) \in E^1 \times U^2$. Hence $k_{i+1} \leq_x q_{i+1}$.

If $p_i \in U^1$, the definition of $\sigma_0 \bowtie \sigma_1$ also implies $k_{i+1} \leq_x q_{i+1}$, since $\xi(T'')$ is σ_1-conform (p_{i+1} is chosen according to σ_1, hence $p_{i+1} \leq_x q_{i+1}$). Because $\xi(T'_{i+1})$ is σ_0-conform (i.e., k_{i+1} is chosen according to σ_0), we have $k_{i+1} \leq_x p_{i+1} \leq_x q_{i+1}$.

The other cases are shown analogously, i.e., the case $(k_i,q_i) \in E^0 \times E^2$ is symmetric to $U^0 \times U^2$, and in the cases $E^0 \times U^2$ and $U^0 \times E^2$, the definition of $\sigma_0 \bowtie \sigma_1$ together with Lemma 2 also straightly shows the desired property. $\square$

Corollary 1 *For $x \in \{di,de,f\}$, $\leq_x$ is a preorder, that is, $\leq_x$ is reflexive and transitive.*

Being a preorder, $\leq_x$ induces an equivalence relation $\equiv_x$ by virtue of

$$p \equiv_x q \qquad \text{iff} \qquad p \leq_x q \text{ and } q \leq_x p. \tag{53}$$

By Theorem 1, $p \equiv_x q$ implies $L(A(p)) = L(A(q))$. The relations $\equiv_{di}$, $\equiv_{de}$, $\equiv_f$ are called *direct*, *delayed* and *fair simulation equivalence*, respectively.

In the next section, we will study quotient automata modulo $\equiv_x$ (in fact, for $x = di$ and $x = de$ only, since fair quotienting does not preserve the language, see [ESW01]). We will need one more notion, namely the notion of maximal and minimal successors of states.

Let q be a state of an alternating Büchi automaton, and $a \in \Sigma$. A state $q' \in \Delta(q,a)$ is a *x-maximal a-successor of q* iff $q'' \leq_x q'$ holds for every $q'' \in \Delta(q,a)$ with $q' \leq_x q''$. We define

$$\max_a^x(q) = \{q' \in \Delta(q,a) \mid q' \text{ is a } x\text{-maximal } a\text{-successor of } q\} \ . \tag{54}$$

A state $q' \in \Delta(q,a)$ is a *x-minimal a-successor of q* iff $q' \leq_x q''$ for every $q'' \in \Delta(q,a)$ with $q'' \leq_x q'$. We define

$$\min_a^x(q) = \{q' \in \Delta(q,a) \mid q' \text{ is a } x\text{-minimal } a\text{-successor of } q\} \ . \tag{55}$$

We will also write $\min_a$ and $\max_a$ instead of $\min_a^x$ and $\max_a^x$ if the context determines the intended winning mode.

As a corollary of Lemma 2, we find:

Corollary 2 *Let $p \in Q^0, q \in Q^1$ be states of alternating Büchi automata A^0 and A^1 such that $p \equiv_x q$. Let $a \in \Sigma$.*

1. *If* $(p,q) \in E^0 \times E^1$ *and* $p' \in \max_a^x(p)$*, then there is a* $q' \in \max_a^x(q)$ *such that* $p' \equiv_x q'$*.*

2. *If* $(p,q) \in U^0 \times U^1$ *and* $p' \in \min_a^x(p)$*, then there is a* $q' \in \min_a^x(q)$ *such that* $p' \equiv_x q'$*.*

3. *If* $(p,q) \in E^0 \times U^1$*, then all x-maximal a-successors of p and all x-minimal a-successors of q are x-equivalent.*

Proof. For the first part, let $(p,q) \in E^0 \times E^1$ and $p' \in \max_a(p)$. By Lemma 2.1, we find a $q' \in \Delta(q,a)$ such that $p' \leq_x q'$. Let $q'' \in \Delta(q,a)$ such that $q' \leq_x q''$. Applying Lemma 2.1 again, there is a $p'' \in \Delta(p,a)$ such that $q'' \leq_x p''$, i.e., since p' is a x-maximal a-successor, $p' \leq_x q' \leq_x q'' \leq_x p'' \leq_x p' \leq_x q'$. Hence q' is a x-maximal a-successor of q and satisfies $p' \equiv_x q'$.

The second part is dual to the case $(p,q) \in E^0 \times E^1$.

For the third part, let $(p,q) \in E^0 \times U^1$, $p' \in \max_a^x(p)$, $q' \in \min_a^x(q)$. Hence $p' \leq_x q'$ by Lemma 2.2. By Lemma 2.3, there is a $p'' \in \Delta(p,a)$ and a $q'' \in \Delta(q,a)$ such that $q'' \leq_x p''$. Lemma 2.2 shows $p' \leq_x q'' \leq_x p'' \leq_x q'$. But since p' is a x-maximal a-successor, $p'' \leq_x p'$ holds; since q' is a x-minimal a-successor, $q' \leq_x q''$ holds. Hence $p' \equiv_x q'$. Using the transitivity of $\equiv_x$, we have $q_0 \equiv_x q_1$ for every $q_0, q_1 \in \min_a^x(q) \cup \max_a^x(p)$. □

6 Quotienting Modulo Direct Simulation

In general, when $\equiv$ is an equivalence relation on the state space of an alternating Büchi automaton A, we call every alternating Büchi automaton a *quotient of A* with respect to $\equiv$ if it is of the form

$$(Q/{\equiv}, \Sigma, [q_I]_\equiv, \Delta', E', U', F/{\equiv}) \tag{56}$$

where $M/{\equiv} = \{[q]_\equiv \mid q \in M\}$ for every $M \subseteq Q$ and $[q]_\equiv = \{q' \in Q \mid q \equiv q'\}$.

Furthermore, the following natural constraints must be satisfied:

1. If $([p]_\equiv, a, [q]_\equiv) \in \Delta'$, then there exist $p' \equiv p$ and $q' \equiv q$ such that $(p', a, q') \in \Delta$,

2. if $[q]_\equiv \subseteq E$, then $[q]_\equiv \in E'$, and

3. if $[q]_\equiv \subseteq U$, then $[q]_\equiv \in U'$.

Note that 1–3 are minimal requirements so that the quotient really reflects the structure of A and is not just any automaton on the equivalence classes of $\equiv$.

A *naive quotient* is a quotient where the converse of 1 is true, that is, where transitions are representative-wise.

Direct simulation is particularly easy (compared to delayed or fair simulation), so one might expect that a naive definition of the quotient automaton modulo direct simulation should be equivalent to the original automaton. Problems arise for mixed equivalence classes, i. e., classes containing both existential and universal states. In the naive quotienting, these states can be either existential or universal.

Consider Figure 2, showing an alternating Büchi automaton A over $\Sigma = \{a, b\}$ on the left, where $0 \equiv_x 2$, and the naive x-quotient on the right.

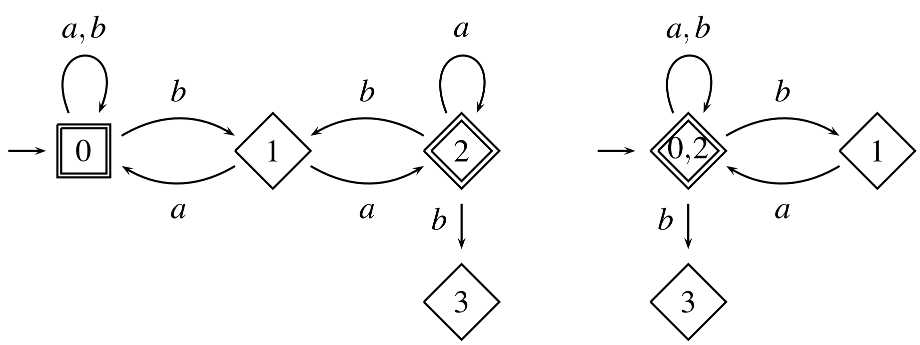

Figure 2: naive quotients don't work

While the language recognized by the naive quotient is Σ^ω, the original automaton does not accept any word containing two or more consecutive b's. The other possible naive quotient, where the state $[0]_x$ is declared universal is not equivalent to the original automaton either: the original automaton on the left-hand side accepts $(ba)^\omega$, but the quotient does not accept this word.

We overcome these problems for direct simulation quotienting by using a more sophisticated transition relation for the quotient automaton exploiting the simple structure of direct simulation games.

6.1 Minimax strategies

Given an alternating Büchi automaton $A = (Q, \Sigma, q_I, \Delta, E, U, F)$, the two relations $\leq_{di} \subseteq Q \times Q$ and $\equiv_{di} \subseteq Q \times Q$ obviously have the following property.

Remark 2 *1. For all $p, q \in Q$, if $p \leq_{di} q$ and $p \in F$, then $q \in F$.*

2. For all $p, q \in Q$, if $p \equiv_{di} q$, then $p \in F$ iff $q \in F$.

Clearly, if $((p_i, q_i), w)$ is a protoplay in an x-game which is conform with a winning strategy for Duplicator, then $p_i \leq_x q_i$ holds for every $i \geq 0$. In the case of direct simulation, the converse is true as well:

Lemma 5 *Let $p_0 \leq_{di} q_0$. A $\leq_{di}$-respecting strategy for Duplicator in $G^{di}(p_0, q_0)$ is a winning strategy.*

Proof. Let $p_0 \leq_{di} q_0$, and let σ be a $\leq_{di}$-respecting strategy of Duplicator for $G^{di}(p_0, q_0)$. Let $T = ((p_i, q_i)_{i<\omega}, w)$ be a σ-conform $G^{di}(p_0, q_0)$-protoplay. By assumption, we have $p_i \leq_{di} q_i$ for every $i \geq 0$. Remark 2 tells us that $q_i \in F$ whenever $p_i \in F$, for every $i \geq 0$. Hence T is a win for Duplicator and σ is a winning strategy for Duplicator. $\square$

The $\leq_{di}$-respecting strategies are exactly the winning strategies. Of these winning strategies, some are optimal in the sense that they choose moves to maximal successors in the second component and to minimal successors in the first component.

Let σ be a Duplicator strategy for a game $G^x(p_0, q_0)$. We call σ a *minimax strategy* if, for every σ-conform protoplay $T = ((p_i, q_i)_{i<\omega}, w)$,

- ∘ if $(p_i, q_i) \in U^0 \times Q^1$, then $p_{i+1} \in \min^x_{w(i)}(p_i)$, and
- ∘ if $(p_i, q_i) \in Q^0 \times E^1$, then $q_{i+1} \in \max^x_{w(i)}(q_i)$.

We note:

Lemma 6 *Let $p_0 \leq_{di} q_0$. Then there exists a $\leq_x$-respecting minimax strategy for Duplicator in $G^x(p_0, q_0)$.*

Proof. Using Lemma 2, a $\leq_x$-respecting minimax strategy for Duplicator can easily be defined inductively. $\square$

6.2 Minimax quotienting

Let $A = (Q, \Sigma, q_0, \Delta, E, U, F)$ be an alternating Büchi automaton. An *x-minimax quotient* of A is a quotient where the transition relation is given by

$$\begin{aligned}\Delta^m_x = \{([p]_x, a, [q]_x) \mid a \in \Sigma, p \in E, q \in \max^x_a(p)\} \\ \cup \{([p]_x, a, [q]_x) \mid a \in \Sigma, p \in U, q \in \min^x_a(p)\} \ . \end{aligned} \tag{57}$$

In particular, mixed classes can be declared existential or universal. This is not surprising, since from Corollary 2 and Remark 2, we can conclude:

Remark 3 *1. For a mixed class $M \in Q/{\equiv_x}$ and $a \in \Sigma$,*

$$\{[q]_x \mid \exists p(p \in M \cap E \wedge q \in \max_a(p))\} \\ = \{[q]_x \mid \exists p(p \in M \cap U \wedge q \in \min_a(p))\} \ , \tag{58}$$

and the size of these sets is 1, i.e., mixed classes are deterministic states of minimax quotients.

2. For every $q \in Q$,

$$[q]_{di} \cap F \neq \emptyset \qquad \text{iff} \qquad [q]_{di} \subseteq F \ . \tag{59}$$

Now it is easy to show:

Theorem 2 (minimax quotients) *Let A be an alternating Büchi automaton as in (2) and B^m any di-minimax quotient of A.*

1. *For all $p_0, q_0 \in Q$ such that $p_0 \leq_{di} q_0$, $A(q_0)$ di-simulates $B^m([p_0]_{di})$ and $B^m([q_0]_{di})$ di-simulates $A(p_0)$, that is, $[p_0] \leq_{di} q_0$ and $p_0 \leq_{di} [q_0]_{di}$.*
2. *A and B^m di-simulate each other, that is, $A \equiv_{di} B^m$.*
3. *A and B^m are equivalent, that is, $L(A) = L(B^m)$.*

Proof. Since mixed classes are deterministic states by Remark 3, it suffices to consider a quotient B^m where every mixed class is existential. Also, it is enough to show the first part, the other parts follow immediately.

We first show that $A(q_0)$ di-simulates $B^m([p_0]_{di})$. To do so, we define a winning strategy σ of Duplicator for $G^{di}([p_0]_{di}, q_0)$. First, for every (q, q') such that $q \leq_{di} q'$, let $\sigma_{q,q'}$ be a $\leq_{di}$-respecting minimax strategy of Duplicator, which exists by Lemma 6.

Now let T be the prefix of a $G^{di}([p_0]_{di}, q_0)$-play such that the last position t of T is in P_1 (that is, $\mathrm{pr}_4(t) = d$) and the last $(Q_{di} \times Q)$-position $([p]_{di}, q)$ of T satisfies $p \leq_{di} q$. We distinguish several cases depending on the suffixes of T.

Case 1, the suffix of the partial protoplay T is of the form $([p]_{di}, q)([p]_{di}, q, a, s, 0, d, 1)([p']_{di}, q, a, d, 1)$, *i.e.,* $p, q \in E$. Then there are a $\hat{p} \in [p]_{di} \cap E$ and a $\hat{p}' \in [p']_{di}$ such that $(\hat{p}, a, \hat{p}') \in \Delta$.

We define $\sigma(T) = ([p']_{di}, q')$ where

$$q' = \mathrm{pr}_2(\sigma_{\hat{p},q}((\hat{p}, q)(\hat{p}, q, a, s, 0, d, 1)(\hat{p}', q, a, d, 1))) \ . \tag{60}$$

Note that $q' \in max_a(q)$ since $\sigma_{\hat{p},q}$ is minimax. By choice of $\sigma_{\hat{p},q}$, we have $p' \equiv_{di} \hat{p}' \leq_{di} q'$.

Case 2, the suffix of T is of the form $([p]_{di},q)([p]_{di},q,a,d,0,d,1)$, i.e., $p \in U$, $q \in E$. We define $\sigma(T) = ([p']_{di},q,a,d,1)$ and $\sigma(\sigma(T)) = ([p']_{di},q')$ where

$$p' = \mathrm{pr}_1(\sigma_{p,q}((p,q)(p,q,a,d,0,d,1))) \ , \tag{61}$$
$$q' = \mathrm{pr}_2(\sigma_{p,q}(\sigma_{p,q}((p,q)(p,q,a,d,0,d,1))) \ . \tag{62}$$

Note that $p' \in \min_a(p)$ since $\sigma_{p,q}$ is minimax. Again, $p' \leq_{di} q'$ holds.

Case 3, the suffix of the partial protoplay T is of the form $([p]_{di},q)([p]_{di},q,a,s,1,d,0)([p]_{di},q',a,d,0)$, i.e., $p,q \in U$. We define $\sigma(T) = ([p']_{di},q')$ where

$$p' = \mathrm{pr}_1(\sigma_{p,q}((p,q)(p,q,a,s,1,d,0)(p,q',a,d,0))) \ . \tag{63}$$

By choice of $\sigma_{p,q}$, we have $p' \in \min_a(p)$ and $p' \leq_{di} q'$.

Now σ is a winning strategy of Duplicator for $G^{di}([p_0]_{di},q_0)$, since any σ-conform play T satisfies, for every position $([p_i]_{di},q_i) \in Q_{di} \times Q$ it contains, $p_i \leq_{di} q_i$. So there cannot be a position $([p_j]_{di},q_j) \in F_{di} \times (Q \setminus F)$ in T, since this would imply $p_j \not\leq_{di} q_j$.

That $B^m([q]_{di})$ di-simulates $A(p)$ can be shown using a symmetrical construction and argumentation. □

The above proof does not use that the set of transitions is minimal–we may allow more transitions, provided that mixed classes are existential in the quotient and no transitions induced by universal states to non-minimal successors are considered for mixed classes. That is, as a corollary of the proof of Theorem 2, we have:

Corollary 3 *Let $A = (Q,\Sigma,q_I,\Delta,E,U,F)$ be an alternating Büchi automaton. Let $B_{di} = (Q/\equiv_{di},\Sigma,[q_I]_{di},\Delta',E',U',F/\equiv_{di})$ be a quotient w.r.t. direct simulation of A such that $\Delta^m_{di} \subseteq \Delta'$, $[q]_{di} \cap E \neq \emptyset$ implies $[q]_{di} \in E'$, and, for every $q_u \in [q]_{di} \cap U$ such that $[q]_{di} \cap E \neq \emptyset$, if $([q_u]_{di},a,[q']_{di}) \in \Delta'$ then there are $\bar{q} \in [q_u]_{di} \cap E$, $\bar{q}' \in [q']_{di}$ such that $(\bar{q},a,\bar{q}') \in \Delta$.*

Then, A and B_{di} simulate each other.

Theorem 2 is false for delayed simulation, as we will see in the next section.

7 Quotienting Modulo Delayed Simulation

If there is a winning strategy for Duplicator in a game $G^{de}(p,q)$, there also is a $\leq_{de}$-respecting minimax strategy, but this may not necessarily be a winning strategy; it is possible that no minimax strategy is winning. Consider the automaton in Figure 3.

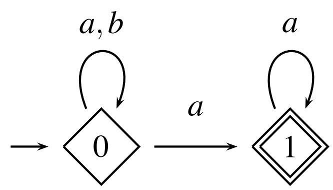

Figure 3: de-minimax quotients don't work

For $x \in \{de, f\}$, we have $0 \geq_x 1$ but not $0 \equiv_x 1$, i.e., $\max_a(0) = \{0\}$. That is, for a minimax strategy σ of Duplicator,

$$\sigma(P^*(1,0,a,d,1)) = (1,0) \tag{64}$$

holds. Hence $((1,0)(1,0,a,s,0,d,1)(1,0,a,d,1))^\omega$ is a σ-conform $G^x(1,0)$-play, but not a win for Duplicator.

Consequently, the language of any minimax quotient is empty since Δ^m_{de} does not contain a transition from $[0]_{de}$ to $[1]_{de}$.

To circumvent this problem, we define semi-elective quotients.

7.1 Semi-elective quotienting

Let $A = (Q, \Sigma, q_I, \Delta, E, U, F)$ be an alternating Büchi automaton. In the *semi-elective quotient* of A, denoted A^s_x, the transition relation is given by

$$\begin{aligned}\Delta^s_x = \{([p]_x, a, [q]_x) \mid (p,a,q) \in \Delta, p \in E\} \\ \cup \{([p]_x, a, [q]_x) \mid a \in \Sigma, [p]_x \subseteq U, q \in \min_a(p)\},\end{aligned} \tag{65}$$

and every mixed class is declared existential.

That is, purely universal classes are treated like in the case of minimax quotienting while purely existential and mixed classes are existential states having all transitions induced by their existential states.

We will show that A and A^s_x simulate each other. For $x = di$, this proposition follows immediately by Corollary 3, i. e.:

Corollary 4 *For every alternating Büchi automaton A, the automata A and A^s_{di} simulate each other. In particular, $L(A) = L(A^s_{di})$.*

The more complicated case, where $x = de$, is treated in the following subsections.

7.2 A simulates A^s_{de}

Although a $\leq_{de}$-respecting minimax strategy σ of Duplicator is not necessarily a winning strategy, it is a $\leq_{de}$-respecting winning strategy for Duplicator in the basic simulation game $G(p_0, q_0)$; the winning condition is assumed to be trivial in the sense that if no early loss occurs, Duplicator wins.

We may extend this observation to a basic simulation game $G(K_0, p_0)$ where K_0 is a state of the quotient automaton A^s_{de} such that $k_0 \leq_{de} p_0$ holds for some $k_0 \in K_0$, which we write as $K_0 \sqsubseteq_{de} p_0$:

Corollary 5 *For all $K_0 \in Q/{\equiv_{de}}$ and for all $p_0 \in Q$ such that $K_0 \sqsubseteq_{de} p_0$, there is a minimax strategy σ of Duplicator for $G(K_0, p_0)$ such that, for all Spoiler strategies τ for $G(K_0, p_0)$, the (τ, σ)-conform protoplay $((K_i, p_i)_{i<\omega}, w)$ satisfies $K_i \sqsubseteq_{de} p_i$ for every $i < \omega$.*

We then say that σ is a $\sqsubseteq_{de}$-respecting minimax strategy.

Proof. Let $K_0 \in Q_{de}$, $p_0 \in Q$. Let T_i be a prefix of a $G(K_0, p_0)$-play such that the last position of T_i is a P_1-position. Again, we make a case distinction.

In the first case, if $(K_i, p_i)(K_i, p_i, a, s, 0, d, 1)(K_{i+1}, p_i, a, d, 1)$ is a suffix of T (hence $K_i \in E_{de}$), we find $k_i \in K_i \cap E$ and $k_{i+1} \in \Delta(k_i, a) \cap K_{i+1}$.

By Lemma 2.1, the set $\{p' \in \Delta(p_i, a) \mid k_{i+1} \leq_{de} p'\}$ is not empty. We choose a $\leq_{de}$-maximal element p_{i+1} of this set (which is an element of $\max^{de}_a(p_i)$) and define $\sigma(T) = (K_{i+1}, p_{i+1})$. Hence $K_{i+1} \sqsubseteq_{de} p_{i+1}$.

In the other cases, the suffixes are of the form $(K_i, p_i, a, d, 0, d, 1)$, of the form $(K_i, p_{i+1}, a, d, 0)$, or of the form $(K_i, p_i)(K_i, p_i, a, d, 0, d, 1)(K_{i+1}, p_i, a, d, 1)$ where K_{i+1} is chosen such that there is a $p' \in \Delta(p_i, a)$ satisfying $K_{i+1} \sqsubseteq_{de} p'$. These cases are also treated using Lemma 2, i.e., by Lemma 2.3 and 2.4, we can find a de-minimal a-successor K_{i+1} of K_i and use similar arguments if Duplicator has to move in the first component. □

With a proof completely analogous to the proof of Lemma 4, we can show Corollary 6.

Corollary 6 *Let $K_0 \in Q/{\equiv_{de}}$, $p_0 \in Q$ such that $K_0 \sqsubseteq_{de} p_0$, and $q_0 \in Q$ such that $p_0 \leq_{de} q_0$. Let σ be a $\sqsubseteq_{de}$-respecting minimax strategy for Duplicator in $G(K_0, p_0)$ and let σ^{de} be a Duplicator winning strategy for $G^{de}(p_0, q_0)$.*

Then $\sigma \bowtie \sigma^{de}$ is a $\sqsubseteq_{de}$-respecting strategy.

And we can easily verify the following.

Lemma 7 *Let $K_0, p_0, q_0, \sigma, \sigma^{de}$ be chosen like in Corollary 6.*

For every Spoiler strategy τ in $G^{de}(K_0, q_0)$, $p_0 \in F$ implies that the $(\tau, \sigma \bowtie \sigma^{de})$-conform play contains a position $(K_j, q_j) \in Q_{de} \times F$, i.e., $\sigma \bowtie \sigma^{de}$ is a winning strategy for Duplicator in $G(K_0, q_0)$ with winning set $\{u \in P^{\omega} \mid \exists i(u_i \in Q_{de} \times F)\}$.

Proof. Let τ be a Spoiler strategy for $G^{de}(K_0, q_0)$, and let $p_0 \in F$. Let $T = ((t_i)_{i<\omega}, w)$ be the $(\tau, \sigma \bowtie \sigma^{de})$-conform protoplay, and assume that there is no $i \in \omega$ such that $t_i = (K_i, q_i) \in Q_{de} \times F$. Since T is $\sigma \bowtie \sigma^{de}$-conform, the play T'' (as defined in Section 4) is σ^{de}-conform. But T'' is not a win for Duplicator, in contradiction to σ^{de} being a winning strategy for Duplicator. Hence there must be a position $t_i = (K_i, q_i)$ in T such that $q_i \in F$. $\square$

We are now ready to show:

Theorem 3 *Let A be a Büchi automaton, and let p, q be states such that $p \leq_{de} q$. $A(q)$ de-simulates $A^s_{de}([p]_{de})$, i.e., there is a winning strategy for Duplicator in $G^{de}([p]_{de}, q)$.*

Proof. To show that there is a winning strategy σ for Duplicator in $G^{de}([p]_{de}, q)$, we fix

1. for every $K \in Q_{de}$, a representative $r(K) \in K$ such that if $K \cap F \neq \emptyset$ then $r(K) \in F$,
2. for every $(K, p) \in Q_{de} \times Q$ such that $K \sqsubseteq_{de} p$, a $\sqsubseteq_{de}$-respecting minimax strategy $\sigma^o_{K,p}$ of Duplicator for $G(K, p)$ (by Corollary 5, there is such a strategy), and
3. for every $(p, q) \in Q \times Q$ such that $p \leq_{de} q$, a winning strategy $\sigma^{de}_{p,q}$ of Duplicator for $G^{de}(p, q)$.

For the prefix T_n of a $G^{de}([p]_{de}, q)$-play T, let $(t_i)_{i \leq n} = (K_i, q_i)_{i \leq n}$ be the subsequence of the $(Q_{de} \times Q)$-positions in T_n. Let

$$j = \min\{i \leq n \mid (K_i, q_i) \in F_{de} \times (Q \setminus F) \ \wedge \forall i' (i \leq i' \leq n \rightarrow q_{i'} \notin F)\} \ , \qquad (66)$$

or $j = 0$ if this set is empty. Let $T_{[j,i]}$ be the suffix of T_i starting with t_j, and define

$$\sigma(T_i) := \sigma^o_{K_j, r(K_j)} \bowtie \sigma^{de}_{r(K_j), q_j}(T_{[j,i]}) \ . \qquad (67)$$

By Corollary 6, σ is $\sqsubseteq_{de}$-respecting. Now if $t_i = (K_i, q_i)$ is the first $(F_{de} \times (Q \setminus F))$-position after the last $(Q_{de} \times F)$-position (or the first $(F_{de} \times (Q \setminus F))$-position at all), we have $K_i \sqsubseteq_{de} q_i$. The strategy σ is updated to $\sigma^o_{K_i, r(K_i)} \bowtie \sigma^{de}_{r(K_i), q_i}$ where

$r(K_i) \in K_i \cap F$, and only the suffix starting with (K_i, q_i) of the play is taken into account for the following moves of Duplicator.

By Lemma 7, Duplicator's use of σ forces the play to reach a position $(K_j, q_j) \in Q_{de} \times F$ (and $K_j \sqsubseteq_{de} q_j$). Hence every position in $F_{de} \times (Q \setminus F)$ is followed by a position in $Q_{de} \times F$ in a σ-conform play. Thus σ is a winning strategy of Duplicator for $G^{de}([p]_{de}, q)$. □

7.3 A^s_{de} simulates A

Theorem 3 says that $A(q)$ de-simulates $A^s_{de}([p]_{de})$. We also want to show that $A^s_{de}([q]_{de})$ de-simulates $A(p)$. The main idea is quite similar to the previous proof: we will not join a $\leq_{de}$-respecting strategy with a winning strategy, but a winning strategy with a "$\equiv_{de}$-respecting" strategy, thus ensuring that the intermediate sequence is a path in the sequence of second state components in the plays of $G^{de}(p, [q]_{de})$.

We start with the following corollary which is a direct consequence of the construction of A^s_{de} together with Corollary 2.

Corollary 7 *Let $q'_0 \in [q_0]_{de}$. There is a Duplicator strategy $\sigma^{\equiv}$ for $G^{de}(q'_0, [q_0]_{de})$ such that, for every $Q \times Q_{de}$-position $(q'_i, [q_i]_{de})$ of a $\sigma^{\equiv}$-conform play, $q'_i \in [q_i]_{de}$ holds.*

We call such a strategy $\equiv_{de}$*-respecting*.

A $\equiv_{de}$-respecting strategy will replace the $\leq_{de}$-respecting minimax strategy of the previous proof. We will show that the join of a winning strategy for $G^{de}(p_0, q_0)$ and a $\equiv_{de}$-respecting strategy for $G^{de}(q_0, [q_0]_{de})$ is a winning strategy for $G^{de}(p_0, [q_0]_{de})$.

Theorem 4 *Let A be a Büchi automaton with states p_0, q_0 such that $p_0 \leq_{de} q_0$. The automaton $A^s_{de}([q_0]_{de})$* de*-simulates $A(p_0)$, i.e., there is a winning strategy for Duplicator in $G^{de}(p_0, [q_0]_{de})$.*

Proof. Let σ^{de} be a winning strategy of Duplicator for $G^{de}(p_0, q_0)$, and let $\sigma^{\equiv}$ be a $\equiv_{de}$-respecting Duplicator strategy for $G^{de}(q_0, [q_0]_{de})$. We show that $\sigma^{de} \bowtie_{q_0} \sigma^{\equiv}$ is a Duplicator winning strategy for $G^{de}(p_0, [q_0]_{de})$.

Let τ be a Spoiler strategy for $G^{de}(p_0, [q_0]_{de})$. Let $T = (t_i)_{i<\omega}$ be the $(\tau, \sigma^{de} \bowtie \sigma^{\equiv})$-conform protoplay with the intermediate q_0-sequence $(q'_i)_{i<\omega}$.

Since $\xi(T')$ is σ^{de}-conform, there is, for every $i < \omega$ such that $\mathrm{pr}_1(t_i) \in F$, a $j \geq i$ such that $q'_j \in F$. Since $\xi(T'')$ is $\sigma^{\equiv}$-conform, we have $q'_j \in \mathrm{pr}_2(t_j)$, hence $t_j \in Q \times F_{de}$. Consequently, $\sigma^{de} \bowtie_{q_0} \sigma^{\equiv}$ is a winning strategy. □

Theorems 3 and 4 yield:

Theorem 5 (semi-elective quotients) *For every alternating Büchi automaton A, the automata A and A^s_{de}* de-*simulate each other, in particular, $L(A) = L(A^s_{de})$.*

7.4 Remarks and possible optimizations

In the construction of the quotient automaton, a transition $(q_u, a, q') \in \Delta$ where $q_u \in U$ only results in a transition $([q_u]_{de}, a, [q']_{de}) \in \Delta_{de}$ if $q' \in \min_a(q_u)$, even if $[q_u]_{de}$ is not a mixed but a purely universal class. This is not a technical trick to permit an easier proof, but a necessity, for without this restriction the resulting quotient automaton would not recognize the language of the original automaton.

Consider the automaton given in Figure 1 again, this time as an automaton over the alphabet $\{b\}$. We have $0 \equiv_{de} 1 >_{de} 2$. So a quotient construction preserving non-minimal successors of universal states would result in the automaton given in Figure 4. But the original automaton accepts b^ω whereas the quotient does not.

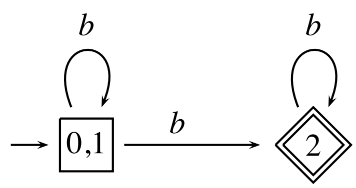

Figure 4: a quotient of the automaton in Figure 1

Above we saw that in some cases existential classes need transitions to non-maximal successors. In certain situations, not all such transitions are really necessary. For example, accepting classes only need maximal transitions:

Remark 4 *Let $A^{s'}_{de}$ be the quotient which is defined just as A^s_{de} but with the transition relation given by*

$$\Delta^{s'}_{de} = \{([p]_{de}, a, [q]_{de}) \mid (p,a,q) \in \Delta \wedge p \in E \wedge ([p]_{de} \cap F = \emptyset \vee q \in \max_a(p))\} \\ \cup \{([p]_{de}, a, [q]_{de}) \mid a \in \Sigma, [p]_{de} \subseteq U, q \in \min_a(p)\} \ . \quad (68)$$

Then $A^{s'}_{de}$ de-simulates A.

In other words, if an existential state is *de*-equivalent to an accepting state, its non-*de*-maximal transitions are superfluous. Conversely, a universal state *de*-equivalent to an accepting state may keep its non-*de*-minimal transitions in the quotient.

Now a state may be equivalent to an accepting state, but this accepting state may not be a state of the original automaton. However, if a state is equivalent to some accepting state, it also is equivalent to an accepting copy of itself:

Definition 1 *Let $A = (Q, \Sigma, q_I, \Delta, E, U, F)$ be an ABA. Let $Q' = \{q' \mid q \in Q\}$ be a disjoint copy of Q (analogously $E', U' \subseteq Q'$), and let $\Delta' = \{(q', a, p) \mid (q, a, p) \in \Delta\}$. Let*

$$A' = (Q \cup Q', \Sigma, q_I, \Delta \cup \Delta', E \cup E', U \cup U', F \cup Q').$$

We define

$$PF_{de} = \{q \in Q \mid q' \leq_{de} q\}.$$

The elements of PF_{de} are called pseudo-accepting states. *Note that $F \subseteq PF_{de}$. We define $A_{PF} = (Q, \Sigma, q_I, \Delta, E, U, PF_{de})$.*

Lemma 8 *For every ABA $A = (Q, \Sigma, q_I, \Delta, E, U, F)$,*

$$A \equiv_{de} A_{PF}.$$

Proof. Obviously, $A \leq_{de} A_{PF}$. Conversely, let σ_q be a Duplicator winning strategy for $G^{de}(q', q)$ for every $q \in PF_{de} \setminus F$, where $q' \in PF_{de}$ is the copy of q in A_{PF}. Let σ be a winning strategy for $G^{de}(A, A)$.

To win the game $G^{de}(A_{PF}, A)$, Duplicator uses the strategy σ, but whenever the winning bit switches from 0 to 1 because of the game reaching a position (p, q) such that $p \in PF_{de} \setminus F$, $q \in Q \setminus F$, Duplicator switches to the strategy $\sigma_p \bowtie \sigma$. He then uses this strategy until the winning bit becomes 0; then, he changes back to σ again. By Lemma 3, this strategy ensures that the winning bit eventually will become 0 whenever it has changed to 1 during the course of a $G^{de}(A_{PF}, A)$-play. □

Note that PF_{de} can be computed together with the simulation relation $\leq_{de}$ without changing the automaton: A state q belongs to PF_{de} iff $q \in F$ or $(p, p, 1)$ is a winning position of Duplicator, so only states of the form $(p, p, 1)$ with $p \notin F$ (and their successor states) have to be added to the game graph if they are not already part of it.

When computing the semi-elective quotient, we may treat existential pseudo-accepting states like accepting states, i.e., remove their non-maximal successors.

7.5 An example

As an example of the construction of the semi-elective quotient automaton modulo delayed simulation, consider Figure 5.

For the automaton A on the left, we have $2 <_{de} 1 \equiv_{de} 5 <_{de} 0 \equiv_{de} 3 <_{de} 4$. Thus there are four states in the quotient automaton A^s_{de} on the right. Since $\min_b(1) = \{2\}$, the edge $([1]_{de}, b, [1]_{de})$ is not in Δ^s_{de}; since $\min_a(0) = \min_b(0) = \{1\}$, there

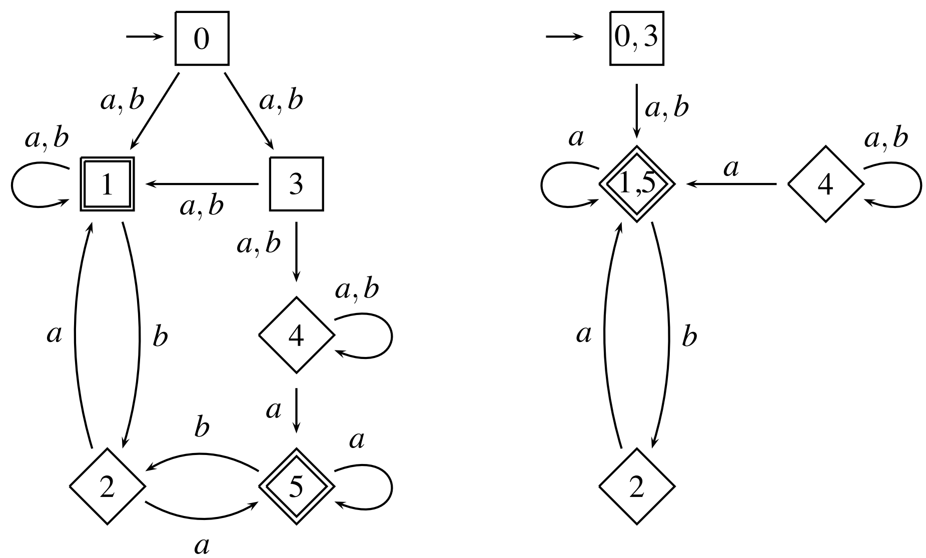

Figure 5: automaton and de-semi-elective quotient

is no edge $([0]_{de}, c, [3]_{de})$ in Δ^s_{de} with $c \in \{a, b\}$. And since $\min_a(3) = \min_b(3) = \{1\}$, there is no edge $([3]_{de}, c, [4]_{de})$ in Δ^s_{de} with $c \in \{a, b\}$. Consequently, the state $[4]_{de}$ is not reachable in A^s_{de} and should be removed in a successive optimization of the quotient automaton.

8 From Alternating Büchi Automata to Nondeterministic Büchi Automata

For every alternating Büchi automaton A, let A_{nd} denote the result of the Miyano-Hayashi construction [MH84], i.e., A_{nd} is a nondeterministic Büchi automaton such that $L(A) = L(A_{nd})$.

Lemma 9 *For $i = 0, 1$, let $A^i = (Q^i, \Sigma, q^i_I, \Delta^i, E^i, U^i, F^i)$ be alternating Büchi automata, and let $x \in \{di, de, f\}$. If $A^0 \leq_x A^1$, then $A^0_{nd} \leq_x A^1_{nd}$.*

Proof. The proof is somewhat similar to the construction of a joint strategy (Section 4) and the proof of Lemma 3. Let σ be a Duplicator winning strategy for $G^x(A^0, A^1)$. We will now simultaneously and inductively construct a Duplicator strategy σ' for $G^x(A^0_{nd}, A^1_{nd})$ and a set of σ-conform $G^x(A^0, A^1)$-protoplays L. This set is called the *logbook* of the partial σ'-conform play.

Remember that, in a $G^x(A^0_{nd}, A^1_{nd})$-protoplay $((P_i)_{i<n}, w)$, the positions P_i are pairs consisting of a state of A^0_{nd} and a state of A^1_{nd}, and these states in turn are pairs of subsets of Q^0 and Q^1, respectively. Hence, every such position P_i is of the form $((M^0_i, N^0_i), (M^1_i, N^1_i))$ where $N^j_i \subseteq M^j_i \subseteq Q^j$ (for every $i < n$, $j = 0, 1$).

For such a protoplay $((P_i)_{i<n}, w)$, the logbook L_{n-1} will have the following properties (for every $i < n$), called the logbook properties.

1. The elements of L_{n-1} are σ-conform $G^x(A^0, A^1)$-protoplays over the word w.
2. For every $q \in M^1_i$, there is a $p \in M^0_i$ such that (p, q) is the $(i+1)$th position of an element of L_{n-1}, and, conversely,
3. if (p, q) is the $(i+1)$th position of an element of L_{n-1}, then $p \in M^0_i$, $q \in M^1_i$.

Initially, for the protoplay $(((\{q^0_I\}, \{q^0_I\} \setminus F^0), (\{q^1_I\}, \{q^1_I\} \setminus F^1)), \varepsilon)$ of length 1, $L_0 = \{((q^0_I, q^1_I), \varepsilon)\}$ is a valid logbook.

Now let $T_n = ((P_i)_{i<n}, w)$ be a partial σ'-conform $G^x(A^0_{nd}, A^1_{nd})$-protoplay with logbook L_{n-1}, and assume Spoiler chooses in $\xi(T_n)$ the position $t'_n = ((M^0_n, N^0_n), (M^1_{n-1}, N^1_{n-1}), a)$. (Since A^0_{nd} and A^1_{nd} are nondeterministic automata, we may assume that Spoiler chooses a letter and a state simultaneously, cf. [ESW01].)

To define $\sigma'(\xi(T_n)t'_n)$, we only have to define the first component of the state Duplicator chooses, i.e., the set M^1_n, since N^1_n is determined by this choice. For every protoplay $K_{n-1} = ((p_i, q_i)_{i<n}, w) \in L_{n-1}$, we distinguish the following four cases. Note that $p_{n-1} \in M^0_{n-1}$ and $q_{n-1} \in M^1_{n-1}$ by the logbook property.

- $(p_{n-1}, q_{n-1}) \in E^0 \times E^1$. Then, there is a $p_n \in M^0_n$ such that $(p_{n-1}, a, p_n) \in \Delta^0$. Let

$$\sigma(\xi(K_{n-1})(p_{n-1}, q_{n-1}, a, s, 0, d, 1)(p_n, q_{n-1}, a, d, 1)) = (p_n, q_n).$$

  We add q_n to M^1_n and $K_n = ((p_i, q_i)_{i<n+1}, wa)$ to the logbook L_n.

- $(p_{n-1}, q_{n-1}) \in U^0 \times E^1$. Then, $\Delta^0(p_{n-1}, a) \subseteq M^0_n$. Let

$$\sigma(\xi(K_{n-1})(p_{n-1}, q_{n-1}, a, d, 0, d, 1)) = (p_n, q_{n-1}, a, d, 1)$$

  and

$$\sigma(\xi(K_{n-1})(p_{n-1}, q_{n-1}, a, d, 0, d, 1)(p_n, q_{n-1}, a, d, 1)) = (p_n, q_n).$$

  We add q_n to M^1_n and $K_n = ((p_i, q_i)_{i<n+1}, wa)$ to the logbook L_n.

- $(p_{n-1}, q_{n-1}) \in E^0 \times U^1$. Then, there is a $p_n \in M_n^0$ such that $(p_{n-1}, a, p_n) \in \Delta^0$, and $\Delta^1(q_{n-1}, a) \subseteq M_n^1$. For every $q_n \in \Delta^1(q_{n-1}, a)$, we add the protoplay $((p_i, q_i)_{i<n+1}, wa)$ to the logbook L_n.

- $(p_{n-1}, q_{n-1}) \in U^0 \times U^1$. Then, $\Delta^0(p_{n-1}, a) \subseteq M_n^0$, and $\Delta^1(q_{n-1}, a) \subseteq M_n^1$. For every $q_n \in \Delta^1(q_{n-1}, a)$, let

$$\sigma(\xi(K_{n-1})(p_{n-1}, q_{n-1}, a, s, 1, d, 0)(p_{n-1}, q_n, a, d, 0)) = (p_n, q_n);$$

  we then add the protoplay $((p_i, q_i)_{i<n+1}, wa)$ to the logbook L_n.

We then define $\sigma'(\xi(T_n)t'_n) = ((M_n^0, N_n^0), (M_n^1, N_n^1))$, where the construction of M_n^1 is determined by t'_n and L_{n-1} as defined above (and N_n^1 in turn is determined by (M_n^1)).

It is easy to check that L_n again has the logbook property and that σ' is a Duplicator strategy for $G^x(A^0_{nd}, A^1_{nd})$. And σ' is in fact a winning strategy: In the case $x = de$, suppose that Spoiler reaches an accepting state $(M_m^0, \emptyset)$ in the mth turn of a $G^{de}(A^0_{nd}, A^1_{nd})$-play π such that Duplicator is in a non-accepting state (M_m^1, N_m^1), i.e., $N_m^1 \neq \emptyset$. Since $N_m^1 \subseteq M_m^1$, by the logbook property there is, for every $q \in N_m^1$, a $p \in M_m^1$ such that $(p, q, 1)$ is the current position of a protoplay in the logbook L_m to π. Since the protoplays in the logbook proceed in a σ-conform way, there is a minimal $m' > m$ with the following property: For every protoplay P in the logbook $L_{m'}$, if $(p, q, 1)$ is the mth position of P, then there is a $k \in \{m+1, \ldots, m'\}$ such that the kth position of P is of the form $(p', q', 0)$ and $q' \in F^1$. By the above definition of σ', this implies that the m'th Duplicator state in π is of the form $(M^1_{m'}, \emptyset)$, i.e., an accepting state.

For $x = di$ and $x = f$, analogous argumentations can be used. $\square$

For a set of states S of an ABA, let $[S]_x = \{[q]_x \mid q \in S\}$ be the set of equivalence classes of the states in S. We say that a set of states S' is a *set of x-minimal representatives* of S if (1) $[S']_x \subseteq [S]_x$ and (2) for every $[q]_x \in [S]_x \setminus [S']_x$, there is a $[q']_x \in [S']_x$ such that $q' \leq_x q$.

The following corollary follows immediately from the proof of Lemma 9.

Corollary 8 *Let $x \in \{di, de\}$. Let A be an ABA, and let (M_0, N_0), (M_1, N_1) be two states of A_{nd} such that M_1 is a set of x-minimal representatives of M_0 and N_1 is a set of x-minimal representatives of N_0.*

Then $(M_0, N_0) \equiv_x (M_1, N_1)$.

That is, only the minimal elements (w.r.t. $\leq_x$) of the subsets in the states of A_{nd} really matter.

Corollary 9 *For every ABA A and $x \in \{di, de\}$, $((A^x)_{nd})^x \equiv_x (A_{nd})^x$ holds.*

Proof. We have $A^x \equiv_x A$, so by Lemma 9, $(A^x)_{nd} \equiv_x A_{nd}$ holds, so $((A^x)_{nd})^x \equiv_x (A_{nd})^x$ follows immediately. $\square$

9 Efficient Algorithms

Efficient algorithms for computing simulation relations of nondeterministic Büchi automata are given in [ESW01]. We can use the same ideas with minor modifications and adjustments. This is explained in the first two subsections. In the third subsection, we focus on weak alternating Büchi automata. We present a specific algorithm for computing simulation relations for weak alternating Büchi automata with a lower time complexity.

9.1 Modifications for the delayed simulation game

For direct and fair simulation, the winning conditions of the corresponding games can be phrased as Büchi or even simpler conditions. This is not true for delayed simulation. But a simple expansion of the game graph will achieve this, as pointed out in [ESW01] already. The crucial information for the players of a delayed simulation game is whether the play has already visited a position in $\hat{F} \cap \bar{F}'$ without having visited a $\hat{F}'$-position since or not. Following [ESW01], we encode this information in the positions of the delayed simulation game. This yields a Büchi game.

For an alternating automaton $A = (Q, \Sigma, q_0, \Delta, E, U, F)$ and $p, q \in Q$, let

$$G(p,q) = (P, P_0, P_1, (p_I, q_I), Z) \tag{69}$$

be the basic simulation game according to Section 3. We define the game

$$G^{de2}(p,q) = (P^{de}, P_0^{de}, P_1^{de}, (p, q, b_{p,q}), Z^{de}, W^{de2}) \tag{70}$$

by

$$P^{de} = P \times \{0,1\} \ , \tag{71}$$
$$P_0^{de} = P_0 \times \{0,1\} \ , \tag{72}$$
$$P_1^{de} = P_1 \times \{0,1\} \ , \tag{73}$$
$$W^{de2} = (P^{de*}(P \times \{0\}))^{\omega} \tag{74}$$

and

$$
\begin{aligned}
Z^{de} = \{((t,b),(t',b)) \in P^{de} \times P^{de} \mid (t,t') \in Z, t' \notin Q^2\} \\
\cup \{((t,b),(t',b)) \in P^{de} \times P^{de} \mid (t,t') \in Z, t' \in (Q \setminus F)^2\} \\
\cup \{((t,b),(t',0)) \in P^{de} \times P^{de} \mid (t,t') \in Z, t' \in Q \times F\} \\
\cup \{((t,b),(t',1)) \in P^{de} \times P^{de} \mid (t,t') \in Z, t' \in F \times (Q \setminus F)\}.
\end{aligned}
$$

with $b_{p,q} = 1$ if $p \in F, q \notin F$ and else $b_{p,q} = 0$. Observe that the parameters p and q influence the initial position only.

We define that $p \leq_{de2} q$ holds if Duplicator has a winning strategy for G^{de2}.

Remark 5 *The game $G^{de}(p,q)$ is a win for Duplicator if and only if the game $G^{de2}(p,q)$ is a win for Duplicator, i.e., $\leq_{de2} = \leq_{de}$.*

So in the remainder it is enough to consider the games $G^{di}(p,q)$, $G^{de2}(p,q)$, and $G^{f}(p,q)$.

9.2 Reduction of the game graphs

By definition and by Remark 5 it is clear that in order to determine whether $p \leq_{di} q$, $p \leq_{de} q$, or $p \leq_{f} q$ holds it is sufficient to determine the winner in the game $G^{di}(p,q)$, $G^{de2}(p,q)$, or $G^{f}(p,q)$, respectively. A priori, the size of these games can be reduced in order to reduce the complexity of determining whether one state simulates another state.

We call a position *productive* if it is reachable in the game graph from a $(Q \times Q)$-position. A position $p \in P$ is a *dead end* if no $(Q \times Q)$-position is reachable from p and $p \notin Q \times Q$. Note that the game graph of a complete automaton does not have dead ends.

Remark 6

1. *A position $(p', q, a, S', 1)$ is productive only if there is a $p \in Q$ such that $(p,a,p') \in \Delta$ and $(p,q) \notin U \times U$.*

2. *A position $(p, q', a, S', 0)$ is productive only if there is a $q \in U$ such that $(q,a,q') \in \Delta$ and $p \in U$.*

3. *A position (p,q,a,S,b,S',b') or (p,q,a,S,b) is a dead end if $\Delta(p,a) = \emptyset$ and $b = 0$, or $\Delta(q,a) = \emptyset$ and $b = 1$.*

That is, in the game graph of an automaton with n states and m transitions, there are $O(n^2 + nm)$ productive states that are not dead ends, and $O(n^2 + nm)$ moves between them. Since we may remove all unproductive positions from the game graph we may assume that there are at most $O(|Q|^2 + |Q| \cdot |\Delta|)$ positions and moves in the game graph. Since we also may assume that every state is reachable from the initial state, we have $|\Delta| \geq |Q| - 1$. So we conclude:

Remark 7 *It can be assumed that the game graphs of $G^{di}(p,q)$, $G^{de2}(p,q)$, and $G^f(p,q)$ have $O(|Q|\cdot|\Delta|)$ positions and moves.*

We may now compute the winning sets and thus the relations $\leq_{di}$, $\leq_{de}$ and $\leq_f$ in the reduced game graph using the algorithms given in [ESW01]. This yields:

Theorem 6 (computing simulation relations) *Given an alternating Büchi automaton A with n states and m transitions, $\leq_{di}$ can be computed in time $O(nm)$. The relations $\leq_{de}$ and $\leq_f$ can be computed in time $O(n^3m)$ and space $O(nm)$.*

The same complexity bounds hold for computing the respective quotients.

9.3 Computing simulation relations of weak alternating Büchi automata

A weak alternating Büchi automaton $A=(Q,\Sigma,q_0,\Delta,E,U,F)$ is an alternating Büchi automaton such that every strongly connected component (SCC for short) $C\subseteq Q$ of the transition graph satisfies $C\subseteq F$ or $C\subseteq Q\setminus F$. This strong requirement lets us design more efficient algorithms for computing simulation relations and quotients, similar to what was done in [KVW00] in the context of emptiness tests for weak alternating automata over one-letter alphabets.

The following is easy to see:

Remark 8 *If C is a SCC of the game graph of $G^x(A,A)$ for $x\in\{di,de2,f\}$, there are SCCs C_0,C_1 of the transition graph of A such that $\{\mathrm{pr}_1(p)\mid p\in C\}\subseteq C_0$ and $\{\mathrm{pr}_2(p)\mid p\in C\}\subseteq C_1$.*

Similarly, if C is a SCC of the game graph of $G^{de2}(A,A)$, precisely one of the following statements holds:

1. For all positions $(p,b)\in C$, $\mathrm{pr}_1(p)\in F$, $\mathrm{pr}_2(p)\in F$ and $b=0$.
2. For all positions $(p,b)\in C$, $\mathrm{pr}_1(p)\notin F$, $\mathrm{pr}_2(p)\in F$ and $b=0$.
3. For all positions $(p,b)\in C$, $\mathrm{pr}_1(p)\in F$, $\mathrm{pr}_2(p)\notin F$ and $b=1$.
4. For all positions $(p,b)\in C$, $\mathrm{pr}_1(p)\notin F$, $\mathrm{pr}_2(p)\notin F$ and $b=0$.
5. For all positions $(p,b)\in C$, $\mathrm{pr}_1(p)\notin F$, $\mathrm{pr}_2(p)\notin F$ and $b=1$.

For a game $G^f(A,A)$ the situation is similar but simpler, for the winning bit (the last component) is missing.

That is, for a SCC of the game graph of $G^{de2}(A,A)$ or $G^f(A,A)$ from which no other SCC is reachable the winning positions can be determined just as in an ordinary game: if the winning bit is 0 Duplicator wins the delayed game starting in any position of C; if $(\mathrm{pr}_1(p), \mathrm{pr}_2(p)) \notin F \times (Q \setminus F)$ for some $p \in C$ Duplicator wins the fair simulation game starting from any position in C; in all other cases Spoiler wins, except for the cases where the SCC consists of a single dead end, but these cases are easy to handle.

Now assume that for a SCC C, the winning positions of all topologically smaller SCCs have already been computed, i. e., for all positions $t \in C$ such that $(t, t') \in Z$ for a $t' \notin C$, we already know whether t' is a winning position either for Spoiler or for Duplicator. If $t \in P_0$ and t' is a win for Spoiler, t also is a win for Spoiler; else if t' is a win for Duplicator, we may simply ignore the move (t, t') in the computation of the winning positions of C (symmetrically for $t \in P_1$). That is, the treatment of C reduces to a game of accessibility in a boolean graph and can be done in linear time, see [And94].

This suggests the following algorithm to compute the winning positions of Duplicator in $G^{de2}(A, A)$ and $G^f(A, A)$:

1. Compute the SCCs $C_0, \ldots, C_{n-1}$ of the game graph (the time expense is linear in the number of positions and moves [Tar72]).

2. Compute a topological sorting $C_{i_0} \leq_T C_{i_1} \leq_T \ldots \leq_T C_{i_{n-1}}$ of the SCCs of the game graph (linear in the number of positions and moves [Knu68]).

3. Compute in the order $C_{i_{n-1}}, C_{i_{n-2}}, \ldots, C_{i_0}$ the winning positions for the separate SCCs. Since these are in fact winning positions of reachability games, this can be done in time linear in the number of positions and moves, see [And94].

Using Remark 7 and Theorem 6, we conclude:

Theorem 7 (weak alternating automata) *Given a weak alternating Büchi automaton with n states and m transitions, $\leq_{di}$, $\leq_{de}$ and $\leq_f$ can be computed in time $O(nm)$.*

The same time bound holds for computing the respective quotients.

Conclusion

We have adapted direct, delayed, and fair simulation relations to alternating Büchi automata, introduced new methods for constructing simulation quotients, and analyzed the complexity of computing these relations and quotients. As a result we can state that even with alternating Büchi automata simulation relations are an appropriate, efficient means for checking language containment and state-space reduction. Since weak alternating Büchi automata are closely related to linear temporal logic formulas, the results also open up new directions for minimizing temporal formulas.

References

[AHKV98] Rajeev Alur, Thomas A. Henzinger, Orna Kupferman, and Moshe Y. Vardi. Alternating refinement relations. In Davide Sangiorgi and Robert de Simone, editors, *9th Int. Conf. on Concurrency Theory (CONCUR '98)*, vol. 1466 of *LNCS*, pp. 163–178, 1998.

[And94] Henrik Andersen. Model checking and boolean graphs. *TCS*, 126(1):3–30, 1994.

[DHW91] David L. Dill, Alan J. Hu, and Howard Wong-Toi. Checking for language inclusion using simulation preorders. In Kim Guldstrand Larsen and Arne Skou, editors, *Computer Aided Verification, 3rd International Workshop, CAV '91*, vol. 575 of *LNCS*, pp. 255–265, 1991.

[EH00] Kousha Etessami and Gerard Holzmann. Optimizing Büchi automata. In Catuscia Palamidessi, editor, *11th Int. Conf. on Concurrency Theory (CONCUR 2000)*, vol. 1877 of *LNCS*, pp. 153–167, 2000.

[ESW01] Kousha Etessami, Rebecca Schuller, and Thomas Wilke. Fair simulation relations, parity games, and state space reduction for Büchi automata. In F. Orejas, P.G. Spirakis, and J. van Leeuwen, editors, *Automata, Languages and Programming (ICALP 2001)*, vol. 2076 of *LNCS*, pp. 694–707, 2001.

[GH82] Yuri Gurevich and Leo Harrington. Trees, automata, and games. In *14th ACM Symposium on the Theory of Computing*, pp. 60–65, 1982.

[HKR97] Thomas A. Henzinger, Orna Kupferman, and Sriram K. Rajamani. Fair simulation. In *Proc. of 8th Int. Conf. on Concurrency Theory (CONCUR '97)*, vol. 1243 of *LNCS*, pp. 273–287, 1997.

[HRHK95] Monika Henzinger Rauch, Thomas A. Henzinger, and Peter W. Kopke. Computing simulations on finite and infinite graphs. In *36th Annual Symposium on Foundations of Computer Science (FOCS '95)*, pp. 453–462, 1995.

[Knu68] Donald E. Knuth. *Fundamental Algorithms*, volume 1 of *The Art of Computer Programming*. Addison-Wesley, 1968.

[KVW00] Orna Kupferman, Moshe Y. Vardi, and Pierre Wolper. An automata-theoretic approach to branching-time model checking. *Journal of the ACM*, 47(2):312–360, 2000.

[MH84] S. Miyano and T. Hayashi. Alternating finite automata on ω-words. *Theoretical Computer Science*, 32:321–330, 1984.

[Mil89] Robin Milner. *Communication and Concurrency*. Prentice Hall, 1989.

[MSS88] David E. Muller, Ahmed Saoudi, and Paul E. Schupp. Weak alternating automata give a simple explanation of why most temporal and dynamic logics are decidable in exponential time. In *3rd IEEE Annual Symposium on Logic in Computer Science*, pp. 422–427, 1988.

[SB00] Fabio Somenzi and Roderick Bloem. Efficient Büchi automata from LTL formulae. In E. Allen Emerson and A. Prasad Sistla, editors, *Computer Aided Verification, 12th International Conference, CAV 2000*, vol. 1855 of *LNCS*, pp. 248–263, 2000.

[Tar72] Robert E. Tarjan. Depth first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2):146–160, 1972.

[Var94] Moshe Y. Vardi. Nontraditional applications of automata theory. In *Theoretical Aspects of Computer Software*, vol. 789 of *LNCS*, pp. 575–597, 1994.