



Simulation relations for alternating Büchi automata[☆]

Carsten Fritz*, Thomas Wilke

*Institut für Informatik und Praktische Mathematik, Christian-Albrechts-Universität zu Kiel,
24098 Kiel, Germany*

Received 11 November 2003; received in revised form 23 December 2004; accepted 11 January 2005

Communicated by T. Henzinger

Abstract

Quotienting by simulation equivalences is a well-established technique for reducing the size of nondeterministic Büchi automata. We adapt this technique to *alternating* Büchi automata. To this end we suggest two new quotients, namely minimax and semi-elective quotients, prove that they preserve the recognized languages, and show that computing them is not more difficult than computing quotients for nondeterministic Büchi automata. 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. We explain the merits of our quotienting procedures with respect to converting alternating Büchi automata into nondeterministic ones.

© 2005 Elsevier B.V. All rights reserved.

Keywords: Alternating automata; Büchi automata; Büchi games; Simulation relations; Quotienting; Automata simplification

1. Introduction

An obvious task of theory is to provide reasonable and practically useful notions for comparing automata. For this purpose, simulation relations [23], which capture the

[☆] An extended abstract of this paper appeared as [12]. Supported by Deutsche Forschungsgemeinschaft under project no. 223228.

* Corresponding author.

E-mail addresses: fritz@ti.informatik.uni-kiel.de (C. Fritz), wilke@ti.informatik.uni-kiel.de (Th. Wilke).

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., [5]. 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., [8,28].

In previous work, simulation relations have been introduced for ordinary and alternating transition systems, see, e.g., [24,19,1], and used for checking trace containment. In addition, there is a series of papers studying simulation relations for (nondeterministic) Büchi automata, see, e.g., [17,8,9], 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 [24,19,1]). 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., [27,30,22], so that simulation relations for alternating Büchi automata are of practical interest—a variant of the concepts described in this paper is used in the LTL-to-Büchi automata implementation $LTL \rightarrow NBA$ [11,10].

Our definitions of the various simulation relations for alternating Büchi automata are game-based and follow closely the approach of [9]. 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, and, as we prove, does not increase the complexity of the algorithms.

Treating existential and universal states at the same time makes the situation complicated. The naive quotient construction, which was also used in [9] 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 that they 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 nondeterministic

Büchi automata, the minimax quotient corresponds to direct simulation with edge deletion, cf. [28], while the semi-elective quotient w.r.t. delayed simulation is the same as the quotient construction of [9]. (Note that a quotient with respect to fair simulation usually cannot preserve the recognized language [9], even in the absence of alternation, but see [16] for a different minimization technique using fair simulation.) 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. Section 4 is the technical core of the paper and lays the ground for proving that direct and delayed quotients preserve the language recognized. In Sections 5 and 6 the definitions of minimax and semi-elective quotient are presented and it is shown that these quotients preserve the language recognized. In Section 7, we show that our simulation relations are compatible with the standard translation of alternating Büchi automata to nondeterministic Büchi automata. Section 8 presents efficient algorithms for computing the simulation relations introduced.

Related work. Henzinger et al. [17] and Henzinger and Rajamani [18] introduce fair bisimulation and simulation relations and describe how they can be computed efficiently. Somenzi and Bloem [28] and Etessami and Holzmann [8] use direct simulation to reduce the size of nondeterministic Büchi automata in the context of checking linear-time temporal properties and present efficient algorithms for computing direct simulation. Etessami et al. [9] improve on [17,18] and introduce delayed simulation to obtain better reductions; they make use of Jurdziński's algorithm [20], which solves parity games. Gurumurthy et al. [16] build on this; Etessami [7] follows a different direction by exploring multi-pebble simulation. Fast algorithms for computing fair simulation using games were also presented by Bustan and Grumberg [3]. Alur et al. [1] study ordinary simulation for alternating transition systems. Gastin and Oddoux [13] use very weak alternating Büchi automata for translating linear-time temporal formulas into Büchi automata; they suggest some simplification rules for alternating Büchi automata, see above. Generating Büchi automata from linear-time temporal formulas is also dealt with in [14,4,10].

There is more work on simulation in general, see, e.g., [23,19], and on using simulation for testing language inclusion in verification, see, e.g., [5].

2. Notation and basic definitions

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 used in this article.

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 i th 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, $\text{pr}_i(t)$ is the i th component of t (for $1 \leq i \leq n$).

2.1. Games

For our purposes, a *game* is a tuple

$$G = (P, P_0, P_1, p_I, Z, W), \quad (1)$$

where P is the set of all *positions* of G , $\{P_0, P_1\}$ is a partition of P into the positions of Players 0 and 1, respectively, where $P_0 = \emptyset$ or $P_1 = \emptyset$ 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 $\pi = p_0 p_1 p_2 \dots$ is winning for Player 1 if π is infinite and $\pi \in W$, or if π is finite and the last position of π belongs to Player 0 (it is her turn, but she cannot move). In all other cases, Player 0 wins the play.

A *strategy for Player 0* is a partial function $\sigma: P^* P_0 \rightarrow P$ 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 $\pi(i + 1) = \sigma(\pi[0..i])$. 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, not all partial plays are σ -conform, which means strategies need not be total functions. In fact, it is usually enough to require that a strategy for Player 0 is defined for all σ -conform partial plays $\pi \in P^* P_0$.

2.2. Alternating Büchi automata

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

$$\mathbf{Q} = (Q, \Sigma, q_I, A^q, E^q, U^q, F^q), \quad (2)$$

where Q is a finite *set of states*, Σ is a finite *alphabet*, $q_I \in Q$ is the *initial state*, $\{E^q, U^q\}$ is a partition of Q in *existential* and *universal states*, where $E^q = \emptyset$ or $U^q = \emptyset$ are allowed, $A^q \subseteq Q \times \Sigma \times Q$ is the *transition relation*, and $F^q \subseteq Q$ is the set of *accepting states*.

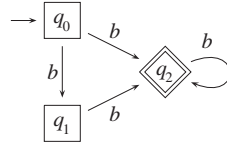


Fig. 1. Alternating Büchi automaton.

Automata will also be named **R** and **S**, with their components named accordingly; we will always assume a common alphabet Σ . We will omit the superscripts when no confusion can arise.

Acceptance of alternating Büchi automata is best defined via games. For an alternating Büchi automaton **Q** as above and an ω -word $w \in \Sigma^\omega$, the *word game* $G(\mathbf{Q}, w)$ is a game where $P = Q \times \omega$ is the set of positions with $P_0 = U^q \times \omega$, $P_1 = E^q \times \omega$, $p_I = (q_I, 0)$, $Z = \{((q, i), (q', i + 1)) \mid (q, w(i), q') \in \Delta\}$, and $W = (P^*(F^q \times \omega))^\omega$.

Following [15], 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 **Q** if Automaton wins the game $G(\mathbf{Q}, w)$. The language *recognized* by **Q** is

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

A *nondeterministic*, i.e., nonalternating automaton, is an automaton as in (2) with $U^q = \emptyset$.

For $q \in Q$, we will write $\mathbf{Q}(q)$ for the automaton which is obtained from **Q** by setting q as the new initial state, i.e., $\mathbf{Q}(q) = (Q, \Sigma, q, \Delta^q, E^q, U^q, F^q)$.

In figures, existential states are shown as diamonds and universal states as squares; accepting states have double lines, see, e.g., Fig. 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 $\mathbf{Q} = (Q, \Sigma, q_I, \Delta^q, E^q, U^q, F^q)$ and $\mathbf{S} = (S, \Sigma, s_I, \Delta^s, E^s, U^s, F^s)$ be alternating Büchi automata. The *basic simulation game* $G(\mathbf{Q}, \mathbf{S})$ is played by two players, *Spoiler* and *Duplicator*, who play the game in rounds. At the beginning of each round, a pair (q, s) of states $q \in Q$ and $s \in S$ is given, and the players play as follows:

1. Spoiler chooses a letter $a \in \Sigma$.
2. The next step depends on the modes (existential or universal) of q and s .

- If $(q, s) \in E^q \times E^s$, then Spoiler chooses a transition $(q, a, q') \in \Delta^q$ and after that Duplicator chooses a transition $(s, a, s') \in \Delta^s$.
 - If $(q, s) \in U^q \times U^s$, then Spoiler chooses a transition $(s, a, s') \in \Delta^s$ and after that Duplicator chooses a transition $(q, a, q') \in \Delta^q$.
 - If $(q, s) \in E^q \times U^s$, then Spoiler chooses transitions $(q, a, q') \in \Delta^q$ and $(s, a, s') \in \Delta^s$.
 - If $(q, s) \in U^q \times E^s$, then Duplicator chooses transitions $(q, a, q') \in \Delta^q$ and $(s, a, s') \in \Delta^s$.
3. The starting pair for the next round is (q', s') .

Intuitively, Spoiler produces, letter by letter, an ω -word as simultaneous input for the automata **Q** and **S**. Spoiler controls the nondeterministic choices of **Q** while Duplicator controls the nondeterministic choices of **S**. This is reversed at universal states: A player loses control of “his” automaton, and the adversary gets to choose a successor state.

The first round begins with the pair (q_I, s_I) . If, at any point during the course of the game, a player cannot proceed any more, he or she loses (early). When the players proceed as above and no player loses early, they construct an infinite sequence $(q_0, s_0)(q_1, s_1) \dots$ of pairs of states (with $q_0 = q_I$ and $s_0 = s_I$), 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 $q_i \in F^q$ we have $s_i \in F^s$.

Delayed simulation (de): Duplicator wins if for every i with $q_i \in F^q$ there exists $j \geq i$ such that $s_j \in F^s$.

Fair simulation (f): Duplicator wins if there are only finitely many i with $q_i \in F^q$ or infinitely many j with $s_j \in F^s$.

In all other cases, Spoiler wins. This completes the description of the games.

The games above can formally be described in the following way, using the game notion of 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 (q, s) for the starting point of a round, and positions of the form (q, s, a, A, b, A', b') , which represent the fact that the round started out in (q, s) , Spoiler chose the letter a , player A (Spoiler or Duplicator) first has to pick a transition in **Q** if $b = 0$ or in **S** if $b = 1$, and after that player A' has to pick a position in **Q** or **S** (depending on b').¹ Finally, we have positions of the form (q, s, a, A', b') which represent the fact that Spoiler chose the letter a , and player A' still has to pick a transition in **Q** ($b' = 0$) or **S** ($b' = 1$). That is, in the formal definition of the game, we use

$$U_{sp} = Q \times S \times \Sigma \times \{sp\} \times \{0, 1\} \times \{sp, du\} \times \{0, 1\}, \quad (4)$$

$$U_{du} = Q \times S \times \Sigma \times \{du\} \times \{0, 1\} \times \{sp, du\} \times \{0, 1\}, \quad (5)$$

$$V_{sp} = Q \times S \times \Sigma \times \{sp\} \times \{0, 1\}, \quad (6)$$

$$V_{du} = Q \times S \times \Sigma \times \{du\} \times \{0, 1\}. \quad (7)$$

¹ The reader may have observed that in a position of the form (q, s, a, A, b, A', b') , the last four components, A, b, A' and b' , are redundant, as they can be inferred from q and s . But our definition will facilitate reading the proofs later.

Given a game type $x \in \{\text{di}, \text{de}, \text{f}\}$, the game $G^x(\mathbf{Q}, \mathbf{S})$ is defined by

$$G^x(\mathbf{Q}, \mathbf{S}) = (P, P_0, P_1, (q_I, s_I), Z, W^x), \quad (8)$$

where

$$P = (Q \times S) \cup U_{sp} \cup U_{du} \cup V_{sp} \cup V_{du}, \quad (9)$$

$$P_0 = (Q \times S) \cup U_{sp} \cup V_{sp}, \quad (10)$$

$$P_1 = U_{du} \cup V_{du}, \quad (11)$$

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

$$((q, s), (q, s, a, sp, 0, du, 1)), \text{ for } q \in E^q, s \in E^s, a \in \Sigma, \quad (12)$$

$$((q, s), (q, s, a, sp, 0, sp, 1)), \text{ for } q \in E^q, s \in U^s, a \in \Sigma, \quad (13)$$

$$((q, s), (q, s, a, du, 0, du, 1)), \text{ for } q \in U^q, s \in E^s, a \in \Sigma, \quad (14)$$

$$((q, s), (q, s, a, sp, 1, du, 0)), \text{ for } q \in U^q, s \in U^s, a \in \Sigma, \quad (15)$$

$$((q, s, a, x, 0, y, 1), (q', s, a, y, 1)), \text{ for } (q, a, q') \in \Delta^q, x, y \in \{sp, du\}, \quad (16)$$

$$((q, s, a, sp, 1, du, 0), (q, s', a, du, 0)), \text{ for } (s, a, s') \in \Delta^s, \quad (17)$$

$$((q, s, a, du, 0), (q', s)), \text{ for } (q, a, q') \in \Delta^q, \quad (18)$$

$$((q, s, a, x, 1), (q, s')), \text{ for } (s, a, s') \in \Delta^s, x \in \{sp, du\}. \quad (19)$$

Note that not all positions are reachable from the initial position of the game or from any position in $\mathbf{Q} \times \mathbf{S}$. These unreachable positions can be removed (cf. Section 8), but if we did this here, 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}^q for the set of all positions with an element from F^q in the first component and \hat{F}^s for the set of all positions with an element from F^s in the second component. Also, we will write \bar{F}^q and \bar{F}^s for $P \setminus \hat{F}^q$ and $P \setminus \hat{F}^s$, respectively. Now we can state the winning conditions formally:

The *direct winning condition* is

$$W^{\text{di}} = ((\bar{F}^q \cup \hat{F}^s) \cap (Q \times S))^\omega. \quad (20)$$

Always at the beginning of a round, it must be the case that the first component is not accepting or the second component is accepting.

The *delayed winning condition* is

$$W^{\text{de}} = P^\omega \setminus P^*(\hat{F}^q \cap \bar{F}^s)(\bar{F}^s)^\omega. \quad (21)$$

It must not be the case that eventually the first component is accepting while the second component is not accepting and remains not accepting forever.

The *fair winning condition* is

$$W^{\text{f}} = P^\omega \setminus P^*((\hat{F}^q \cap \bar{F}^s)(\bar{F}^s)^*)^\omega. \quad (22)$$

It must not be the case that eventually the second component is never accepting while the first component is accepting infinitely often.

For $x \in \{\text{di}, \text{de}, \text{f}\}$, we define a relation \leq_x on alternating Büchi automata. We write

$$\mathbf{Q} \leq_x \mathbf{S} \text{ when Duplicator has a winning strategy in } G^x(\mathbf{Q}, \mathbf{S}) \quad (23)$$

and say that \mathbf{S} x -simulates \mathbf{Q} . For states q of \mathbf{Q} , s of \mathbf{S} , we write $q \leq_x s$ to indicate that $\mathbf{S}(s)$ x -simulates $\mathbf{Q}(q)$. We write $G^x(q, s)$ instead of $G^x(\mathbf{Q}(q), \mathbf{S}(s))$ if \mathbf{Q} and \mathbf{S} are obvious from the context.

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

We argue that the games $G^{\text{de}}(q_0, q_1)$, $G^{\text{de}}(q_1, q_0)$, $G^{\text{f}}(q_0, q_1)$ and $G^{\text{f}}(q_1, q_0)$ are a win for Duplicator. To see this, consider the strategy σ defined by

$$\sigma(P^*(q_0, q_2, b, \text{du}, 0)) = (q_2, q_2), \quad (24)$$

$$\sigma(P^*(q_1, y, b, \text{du}, 0)) = (q_2, y), \quad \text{for } y = q_1, q_2, \quad (25)$$

$$\sigma(P^*(q_2, q_2, b, \text{du}, 1)) = (q_2, q_2). \quad (26)$$

In a play starting in position (q_0, q_1) , Spoiler has to choose the letter b , or he loses early, and he has to choose transition (q_1, b, q_2) , i.e., the play reaches position $(q_0, q_2, b, \text{du}, 0)$ after his move. Playing according to σ , Duplicator now chooses the transition (q_0, b, q_2) , and the next round starts in position (q_2, q_2) . Now Spoiler always has to choose the letter b and the transition (q_2, b, q_2) , but Duplicator (using σ) always chooses the same transition, so the play stays in (q_2, q_2) and thus is a win for Duplicator.

If the play starts in (q_1, q_0) , the strategy σ also ensures that a play is either an early defeat for Spoiler or eventually stays in (q_2, q_2) . That is, states q_0 and q_1 are equivalent w.r.t. delayed and fair simulation. Note that $q_2 \leq_x q_0$ and $q_2 \leq_x q_1$ for $x \in \{\text{de}, \text{f}\}$; the converse is false.

Lemma 1 (Cf. Etessami et al. [9]). *For every alternating Büchi automaton, the following relations hold between the three types of simulation relations:*

$$\leq_{\text{di}} \subseteq \leq_{\text{de}} \subseteq \leq_{\text{f}}, \quad (27)$$

and these inclusions are strict for certain automata.

Proof. Since $W^{\text{di}} \subseteq W^{\text{de}} \subseteq W^{\text{f}}$, the inclusions follow immediately. It is easy to see that these inclusions are strict for the automata \mathbf{Q} and \mathbf{S} defined by

$$\mathbf{Q} = (\{q_0, q_1\}, \{a\}, q_0, \{(q_i, a, q_1) \mid i \in \{0, 1\}\}, \{q_0, q_1\}, \emptyset, \{q_1\}), \quad (28)$$

$$\mathbf{S} = (\{s_0, s_1\}, \{a\}, s_0, \{(s_i, a, s_1) \mid i \in \{0, 1\}\}, \{s_0, s_1\}, \emptyset, \{s_0\}). \quad (29)$$

In fact, we have $q_1 \leq_{\text{de}} q_0$, but $q_1 \not\leq_{\text{di}} q_0$, and $s_0 \leq_{\text{f}} s_1$, but $s_0 \not\leq_{\text{de}} s_1$. \square

We say that an alternating Büchi automaton as in (2) is *complete* if for every $q \in Q$, $a \in \Sigma$, there is a state $q' \in Q$ such that $(q, a, q') \in \Delta^q$. Clearly, if we are given two alternating Büchi automata \mathbf{Q} and \mathbf{S} such that $\mathbf{Q} \leq_x \mathbf{S}$ for some $x \in \{\text{di}, \text{de}, \text{f}\}$, then, by adding at most two new states and at most $|\Sigma| \cdot (|Q| + 2)$ transitions, we can turn \mathbf{Q} and \mathbf{S} into equivalent complete automata \mathbf{Q}' and \mathbf{S}' such that $\mathbf{Q}' \leq_x \mathbf{S}'$ still holds. Therefore, we

henceforth assume that all automata are complete; we allow incomplete automata only in Section 8, 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 $x \in \{\text{di}, \text{de}, \text{f}\}$ and let \mathbf{Q} and \mathbf{S} be alternating Büchi automata. If $\mathbf{Q} \leq_x \mathbf{S}$, then $L(\mathbf{Q}) \subseteq L(\mathbf{S})$.*

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 $T = t_0 t_0^U t_0^V t_1 t_1^U t_1^V \dots$ where $t_i \in Q \times S$, $t_i^U \in U_{sp} \cup U_{du}$ and $t_i^V \in V_{sp} \cup V_{du}$. But the play T is obviously completely determined by the infinite sequence $t_0 t_1 \dots$ and the sequence of letters $w \in \Sigma^\omega$ in the third component of the elements of the sequence $t_0^U t_1^U t_2^U \dots$ (recall that each t_i^U is of the form (q, s, a, A, b, A', b') where a is a letter from Σ). A similar statement holds true for a partial play ending in a position in $Q \times S$. That is, there is a natural partial mapping

$$\xi: (Q \times S)^\infty \times \Sigma^\infty \rightarrow \text{set of partial or complete } G^x(\mathbf{Q}, \mathbf{S})\text{-plays,} \quad (30)$$

which maps $((q_i, s_i)_{i < n}, w)$ (where $n \in \omega \cup \{\omega\}$) to the corresponding partial play, provided there is such a play. This is the case if $|w| + 1 = n$ and, for all i with $i + 1 < n$, $(q_i, w(i), q_{i+1}) \in \Delta^q$ and $(s_i, w(i), s_{i+1}) \in \Delta^s$.

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

Proof of Theorem 1. Let σ be a winning strategy for Duplicator in $G^x(\mathbf{Q}, \mathbf{S})$. Let $w \in L(\mathbf{Q})$, and let σ^q be a winning strategy for Automaton in $G(\mathbf{Q}, w)$. We have to show that Automaton has a winning strategy σ^s in $G(\mathbf{S}, w)$.

We first give an informal description of the way Automaton plays. While playing $G(\mathbf{S}, w)$, Automaton (in $G(\mathbf{S}, w)$) simultaneously plays the game $G^x(\mathbf{Q}, \mathbf{S})$ and the game $G(\mathbf{Q}, w)$. In the two plays he makes the moves for all players, Spoiler and Duplicator as well as Automaton and Pathfinder, and uses σ and σ^q 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 $G(\mathbf{Q}, w)$ and Spoiler and Duplicator in $G^x(\mathbf{Q}, \mathbf{S})$ will be called the automaton puppet, the pathfinder puppet, the spoiler puppet, and the duplicator puppet, respectively.

Automaton plays in such a way that after each round the state components in $G(\mathbf{Q}, w)$ and $G(\mathbf{S}, w)$ agree with the two state components of $G^x(\mathbf{Q}, \mathbf{S})$, and the partial games in $G^x(\mathbf{Q}, \mathbf{S})$ and $G(\mathbf{Q}, w)$ are conform with σ and σ^q . Then, clearly, since σ and σ^q are winning, in the emerging plays in $G(\mathbf{S}, w)$ infinitely many states will be in F^s , that is, Automaton will win $G(\mathbf{S}, w)$.

The above can be achieved when

- in $G^x(\mathbf{Q}, \mathbf{S})$, Automaton uses σ to determine the moves of the duplicator puppet, and,

- in $G(\mathbf{Q}, w)$, Automaton uses σ^q to determine the moves of the automaton puppet. This is explained in more detail now.

Suppose that a play of $G(\mathbf{S}, w)$ is in a position (s, i) while $G(\mathbf{Q}, w)$ is in position (q, i) . Consequently, $G^x(\mathbf{Q}, \mathbf{S})$ is in position (q, s) . Automaton makes the spoiler puppet in the game $G^x(\mathbf{Q}, \mathbf{S})$ choose the letter $w(i)$. Automaton then proceeds as follows:

- If s is an existential state of \mathbf{S} , then Automaton has to move in $G(\mathbf{S}, w)$. Automaton proceeds according to the mode of q .
 - If q is an existential state of \mathbf{Q} , then Automaton makes the automaton puppet in the game $G(\mathbf{Q}, w)$ move according to the strategy σ^q . Automaton makes the spoiler puppet in $G^x(\mathbf{Q}, \mathbf{S})$ mimic this move and then makes the duplicator puppet in $G^x(\mathbf{Q}, \mathbf{S})$ react to this move by choosing a successor state s' of s according to the strategy σ . This state s' is the successor state Automaton chooses as his move in $G(\mathbf{S}, w)$.
 - If q is a universal state of \mathbf{Q} , then Automaton makes the duplicator puppet in game $G^x(\mathbf{Q}, \mathbf{S})$ choose successor states q' of q and s' of s , according to σ . The pathfinder puppet mimics the choice of q' in $G(\mathbf{Q}, w)$ while Automaton moves to s' in $G(\mathbf{S}, w)$.
- If s is a universal state of \mathbf{S} , then Pathfinder has to move in $G(\mathbf{S}, w)$. Again, Automaton proceeds according to the mode of q :
 - If q is an existential state of \mathbf{Q} , then Automaton makes the automaton puppet in the game $G(\mathbf{Q}, w)$ move according to the strategy σ^q . He makes the spoiler puppet in $G^x(\mathbf{Q}, \mathbf{S})$ mimic the automaton puppet's move in $G(\mathbf{Q}, w)$ and the Pathfinder's move in $G(\mathbf{S}, w)$.
 - If q is a universal state of \mathbf{Q} , then Automaton makes the spoiler puppet mimic the move of Pathfinder in $G^x(\mathbf{Q}, \mathbf{S})$. Automaton then makes the duplicator puppet in game $G^x(\mathbf{Q}, \mathbf{S})$ choose a successor state q' of q , according to σ . The pathfinder puppet in $G(\mathbf{Q}, w)$ mimics this choice.

We now proceed with a formal treatment. In order to define the winning strategy σ^s of Automaton in $G(\mathbf{S}, w)$, we first need a partial function pr^0 mapping partial $G^x(\mathbf{Q}, \mathbf{S})$ -protoplays to prefixes of $G(\mathbf{Q}, w)$ -plays. For any partial $G^x(\mathbf{Q}, \mathbf{S})$ -protoplay $\pi = ((q_i, s_i)_{i \leq n}, w[0..n-1])$, we set

$$\text{pr}^0(\pi) = (q_0, 0) \dots (q_n, n). \quad (31)$$

As another auxiliary function, we define the partial function T by

$$T: (Q \times S \times \omega)^* \rightarrow (Q \times S)^* \times \Sigma^*, \quad (32)$$

$$(q_i, s_i, i)_{i \leq n} \mapsto ((q_i, s_i)_{i \leq n}, w[0..n-1]). \quad (33)$$

Simultaneously, we define a partial function

$$\hat{\sigma}: (Q \times S \times \omega)^* \rightarrow Q \times S \times \omega, \quad (34)$$

describing the interplay of σ^q and σ for a given partial $G^x(\mathbf{Q}, \mathbf{S})$ -play, and a partial function

$$h: (S \times \omega)^* \rightarrow (Q \times S \times \omega)^*. \quad (35)$$

The function $\hat{\sigma}$ will be defined only for sequences $\rho = (q_i, s_i, i)_{i \leq n}$ where $s_n \in E^s$ and $T(\rho)$ is a partial $G^x(\mathbf{Q}, \mathbf{S})$ -protoplay; the function h assigns such sequences ρ to prefixes of $G(\mathbf{Q}, w)$ -plays.

To define the value of the partial function $\hat{\sigma}$ for a sequence $\rho = (q_i, s_i, i)_{i \leq n}$ where $s_n \in E^s$, we first assume that $q_n \in E^q$.

We then define

$$\hat{\sigma}: \rho \mapsto (\sigma(\pi p p'), n + 1), \quad (36)$$

where π is the partial $G^x(\mathbf{Q}, \mathbf{S})$ -play $\xi(T(\rho))$, $p = (q_n, s_n, w(n), sp, 0, du, 1)$ is the next position of the play, and $p' = (\text{pr}_1(\sigma^q(\text{pr}^0(T(\rho))))$, $s_n, w(n), du, 1)$ is the successor position chosen via σ^q .

For the case that $q_n \in U^q$, we define

$$\hat{\sigma}: \rho \mapsto (\sigma(\pi p p'), n + 1), \quad (37)$$

where again π is the partial $G^x(\mathbf{Q}, \mathbf{S})$ -play $\xi(T(\rho))$. The following position now is $p = (q_n, s_n, w(n), du, 0, du, 1)$, and $p' = \sigma(\pi p)$ is the successor position chosen via σ .

The partial function h is inductively defined as follows. For the initial case, we set

$$h: (s_I, 0) \mapsto (q_I, s_I, 0). \quad (38)$$

Now let $s_0 = s_I$ and $\rho = h((s_i, i)_{i \leq n})$, and assume that the last tuple in this sequence is (q_n, s_n, n) . If $q_n \in E^q$, we define

$$h: (s_i, i)_{i \leq n+1} \mapsto \rho p, \quad (39)$$

where $p = (q_{n+1}, s_{n+1}, n + 1)$ with $q_{n+1} = \text{pr}_1(\sigma^q(\text{pr}^0(T(\rho))))$.

For the case $q_n \in U^q$, we have to look at the sub-cases $s_n \in E^s$ and $s_n \in U^s$. If $s_n \in E^s$, we define

$$h: (s_i, i)_{i \leq n+1} \mapsto \rho p, \quad (40)$$

where $p = (\text{pr}_1(\hat{\sigma}(\rho)), s_{n+1}, n + 1)$, while for the sub-case $s_n \in U^s$, we define

$$h: (s_i, i)_{i \leq n+1} \mapsto \rho(q_{n+1}, s_{n+1}, n + 1), \quad (41)$$

where $(q_{n+1}, s_{n+1}) = \sigma(\xi(T(\rho)) p p')$ with $p = (q_n, s_n, w(n), sp, 1, du, 0)$ and $p' = (q_n, s_{n+1}, w(n), du, 0)$.

With these definitions, we can now define a Duplicator winning strategy σ^s for $G(\mathbf{S}, w)$ by

$$\sigma^s: (s_i, i)_{i \leq n} \mapsto (\text{pr}_2(\hat{\sigma}(h((s_i, i)_{i \leq n+1}))), n + 1), \quad (42)$$

for $s_n \in E^s$.

With these definitions, it is tedious but routine to check the following:

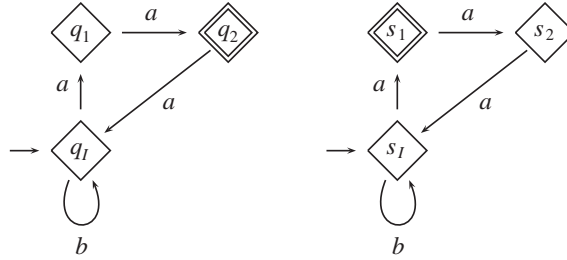
1. The function σ^s is defined for $(s_I, 0)$ if $s_I \in E^s$, and if $(s_i, i)_{i < n}$ is a partial σ^s -conform $G(\mathbf{S}, w)$ -play such that $s_{n-1} \in E^s$, then σ^s is defined for $(s_i, i)_{i < n}$.

That is, σ^s is in fact a Duplicator strategy for $G(\mathbf{S}, w)$.

2. If $(s_i, i)_{i < \omega}$ is a σ^s -conform $G(\mathbf{S}, w)$ -play, then $\text{pr}^0(\xi(T(h((s_i, i)_{i < n}))))$ is a partial σ^q -conform $G(\mathbf{Q}, w)$ -play, for all $n < \omega$.

That is, since σ^q is a winning strategy, in the $G(\mathbf{Q}, w)$ -play connected to $(s_i, i)_{i < \omega}$ via h there are infinitely many occurrences of accepting states.

3. If $(s_i, i)_{i < \omega}$ is a σ^s -conform $G(\mathbf{S}, w)$ -play, then $\xi(T(h((s_i, i)_{i < n})))$ is a partial σ -conform $G^x(\mathbf{Q}, \mathbf{S})$ -play, for all $n < \omega$.

Fig. 2. Spoiler needs memory to win $G^{\text{de}}(\mathbf{Q}, \mathbf{S})$.

From 2 and 3 and since σ also is a winning strategy, we conclude that there must be infinitely many occurrences of accepting states in $(s_i)_{i < \omega}$, that is, σ^s is a winning strategy. \square

3.3. Positional strategies for simulation games

A strategy σ of Player i , $i \in \{0, 1\}$, for a game $G = (P, P_0, P_1, p_I, Z, W)$ as defined in Section 2.1 is called *positional* or *memoryless* if, for every $\pi, \pi' \in P^*$ and $p \in P_i$, either $\sigma(\pi p) = \sigma(\pi' p)$ or both $\sigma(\pi p)$ and $\sigma(\pi' p)$ are undefined. That is, a positional strategy σ only depends on the last position of a partial play and can hence be seen as a partial function $P_i \rightarrow P$.

It is well known that, if G is a so-called reachability game or a parity game and Player i wins G , then Player i has a positional winning strategy [6,26]. Corollary 1 follows immediately.

Corollary 1. *Let \mathbf{Q} and \mathbf{S} be two ABA. Spoiler (Duplicator) wins $G^{\text{di}}(\mathbf{Q}, \mathbf{S})$ or $G^{\text{f}}(\mathbf{Q}, \mathbf{S})$ if and only if there is a positional winning strategy of Spoiler (Duplicator) for $G^{\text{di}}(\mathbf{Q}, \mathbf{S})$ or $G^{\text{f}}(\mathbf{Q}, \mathbf{S})$, respectively.*

For delayed simulation games, this is only true for Duplicator.

Proposition 1. 1. *For alternating Büchi automata \mathbf{Q} and \mathbf{S} , Duplicator wins $G^{\text{de}}(\mathbf{Q}, \mathbf{S})$ if and only if there is a positional winning strategy of Duplicator for $G^{\text{de}}(\mathbf{Q}, \mathbf{S})$.*

2. *There are Büchi automata \mathbf{Q} and \mathbf{S} such that Spoiler wins $G^{\text{de}}(\mathbf{Q}, \mathbf{S})$, but no positional winning strategy is winning for Spoiler.*

We will only proof the second claim of the proposition here; the proof of the first claim needs some preparation and can be found in Section 8.2.

Proof of Proposition 1. *part 2:* Consider the Büchi automata \mathbf{Q} (on the left) and \mathbf{S} (on the right) of Fig. 2.

We claim that Spoiler wins $G^{\text{de}}(\mathbf{Q}, \mathbf{S})$.

First, note that there are only existential states in the two automata. Therefore, in each round first Spoiler moves in \mathbf{Q} and then Duplicator moves in \mathbf{S} . Next, note that the automata are deterministic. Thus, a play is completely determined by what Spoiler does and, moreover,

the moves of Spoiler are completely determined by the letters he chooses at the beginning of each round. That is, a strategy of Spoiler can be denoted by an ω -word, for instance, $aaab^\omega$. It is easy to see that the set of all winning strategies for Spoiler can be denoted by elements from

$$(aaa + b)^*aaab^\omega, \quad (43)$$

in particular, Spoiler wins $G^{\text{de}}(\mathbf{Q}, \mathbf{S})$. However, the only two positional strategies for Spoiler which do not result in an early loss are a^ω and b^ω , and do not belong to the above set. That is, there is no positional winning strategy for Spoiler in $G^{\text{de}}(\mathbf{Q}, \mathbf{S})$. \square

4. Composing simulation strategies

In this section, let $x \in \{\text{di}, \text{de}, \text{f}\}$. We will introduce the join of two Duplicator strategies, a concept fundamental for the proofs of the results in Sections 4.2 and 6. The idea is that two strategies for simulation games starting in positions (q, r) and (r, s) , respectively, can be merged into a joint strategy for a game starting in (q, s) ; this joint strategy inherits crucial properties of the two original strategies (see Lemma 2 and Corollary 7), and will also be used to show that the relation \leq_x is transitive.

4.1. Definition of the join of strategies

Let $q \in \mathbf{Q}, r \in \mathbf{R}, s \in \mathbf{S}$. Let σ_0 be a Duplicator strategy for the basic game $G(q, r)$, and let σ_1 be a Duplicator strategy for the basic game $G(r, s)$.

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 $G(q, s)$ using $\sigma_0 \bowtie \sigma_1$, simultaneously plays $G(q, r)$ and $G(r, s)$, using σ_0 and σ_1 , respectively. His four puppets are Spoiler and Duplicator of these games. We will call Spoiler and Duplicator of $G(q, r)$ the left spoiler puppet and the left duplicator puppet, while Spoiler and Duplicator of $G(r, s)$ are the right spoiler puppet and the right duplicator puppet.

Duplicator (of $G(q, s)$, our puppeteer) plays in such a way that after each round the first state component of $G(q, r)$ and the second state component of $G(r, s)$ agree with the first and second state components of $G(q, s)$, respectively, and the second state component of $G(q, r)$ agrees with the first state component of $G(r, s)$, and the partial plays in $G(q, r)$ and $G(r, s)$ are conform with σ_0 and σ_1 , respectively.

This can be achieved in the following way. In $G(q, r)$, Duplicator uses σ_0 to determine the moves of the left duplicator puppet, while in $G(r, s)$, he uses σ_1 to determine the moves of the right duplicator puppet. The spoiler puppets just mimic the moves of Spoiler and the duplicator puppets.

We will clarify this interplay by describing the course of two exemplary rounds.

Consider a position (q_i, s_i) of $G(q, s)$ where the simultaneous plays of $G(q, r)$ and $G(r, s)$ are in positions (q_i, r_i) and (r_i, s_i) , respectively, such that $(q_i, r_i, s_i) \in E^q \times U^r \times E^s$. Let Spoiler choose a letter a in the $G(q, s)$ -play.

At first, Duplicator makes the two spoiler puppets choose the same letter a .

Since q_i is existential, Spoiler has to choose an a -successor state q_{i+1} as his next move in $G(q, s)$. Duplicator makes the left spoiler puppet mimic this move in $G(q, r)$.

Since r_i is universal and s_i is existential, Duplicator proceeds as follows. He lets the right duplicator puppet choose a -successors r_{i+1} of r_i and s_{i+1} of s_i according to σ_1 in $G(r, s)$. The left spoiler puppet then mimics this and chooses r_{i+1} as its next move in $G(q, r)$; similarly, Duplicator chooses s_{i+1} in $G(q, s)$.

Now consider a situation where $(q_i, r_i, s_i) \in U^q \times E^r \times E^s$. After mimicking Spoiler's choice of a letter a by the two spoiler puppets, Duplicator makes the left duplicator puppet choose a -successors q_{i+1} of q_i and r_{i+1} of r_i in the $G(q, r)$ -play according to σ_0 . The choice of r_{i+1} is mimicked as its next move by the right spoiler puppet while the choice of q_{i+1} is used by Duplicator as his next move in $G(q, s)$. Duplicator then makes the right duplicator puppet react to the move of the right spoiler puppet by choosing an a -successor s_{i+1} of s_i according to σ_1 in the $G(r, s)$ -play. Duplicator copies this choice of the right duplicator puppet as his next move.

That is, first the spoiler puppets serve to mimic the moves of Spoiler. The moves of the left and right duplicator puppets are then guided by the two strategies σ_0 and σ_1 , respectively. That is, the left duplicator puppet controls the choice of r_{i+1} if r_i is existential, and this choice is mimicked by the right spoiler puppet, which in turn allows the right duplicator puppet to react, if necessary. This situation is reversed if r_i is universal.

To define this strategy formally, we also have to keep track of the sequence of the \mathbf{R} -states in the play of $G(q, r)$, which is identical to the sequence of \mathbf{R} -states in the play of $G(r, s)$. We now continue with the formal definitions.

We simultaneously and inductively define the *joint strategy* $\sigma_0 \bowtie \sigma_1$, which is a Duplicator strategy for $G(q, s)$, and a sequence of \mathbf{R} -states (starting with r) for partial $(\sigma_0 \bowtie \sigma_1)$ -conform $G(q, s)$ -plays, the so-called *intermediate 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 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 sequence for the prolonged prefix.

The joint strategy and the intermediate sequence will have the following property.

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

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

Now assume that for a $(\sigma_0 \bowtie \sigma_1)$ -conform protoplay $T = ((q_i, s_i)_{i < n+1}, w)$, the intermediate sequence is given by $(r_i)_{i < n+1}$ (and $q_0 = q, r_0 = r, s_0 = s$). In particular, T and $(r_i)_{i < n+1}$ have Property 1. Let $T^0 = ((q_i, r_i)_{i < n+1}, w)$ and $T^1 = ((r_i, s_i)_{i < n+1}, w)$. Recall that the last position of $\xi(T)$ is (q_n, s_n) .

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

Case EEE, $(q_n, r_n, s_n) \in E^q \times E^r \times E^s$: Assume Spoiler chooses the $G(q, s)$ -positions $t_n^U = (q_n, s_n, a, sp, 0, du, 1)$ and $t_n^V = (q_{n+1}, s_n, a, du, 1)$. Let

$$\sigma_0(\xi(T^0)(q_n, r_n, a, sp, 0, du, 1)(q_{n+1}, r_n, a, du, 1)) = (q_{n+1}, r_{n+1}), \quad (44)$$

$$\sigma_1(\xi(T^1)(r_n, s_n, a, sp, 0, du, 1)(r_{n+1}, s_n, a, du, 0)) = (r_{n+1}, s_{n+1}). \quad (45)$$

We define

$$\sigma_0 \bowtie \sigma_1(\xi(T)t_n^U t_n^V) = (q_{n+1}, s_{n+1}) \quad (46)$$

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

Case EUE, $(q_n, r_n, s_n) \in E^q \times U^r \times E^s$: Assume Spoiler chooses the $G(q, s)$ -positions $t_n^U = (q_n, s_n, a, sp, 0, du, 1)$ and $t_n^V = (q_{n+1}, s_n, a, du, 1)$. Let

$$\sigma_1(\xi(T^1)(r_n, s_n, a, du, 0, du, 1)) = (r_{n+1}, s_n, a, du, 1), \quad (47)$$

$$\sigma_1(\xi(T^1)(r_n, s_n, a, du, 0, du, 1)(r_{n+1}, s_n, a, du, 1)) = (r_{n+1}, s_{n+1}). \quad (48)$$

We define

$$\sigma_0 \bowtie \sigma_1(\xi(T)t_n^U t_n^V) = (q_{n+1}, s_{n+1}) \quad (49)$$

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

Case UEU, $(q_n, r_n, s_n) \in U^q \times E^r \times U^s$: Assume Spoiler chooses the $G(q, s)$ -positions $t_n^U = (q_n, s_n, a, sp, 1, du, 0)$ and $t_n^V = (q_n, s_{n+1}, a, du, 0)$. Let

$$\sigma_0(\xi(T^0)(q_n, r_n, a, du, 1, du, 0)) = (q_{n+1}, r_n, a, du, 1), \quad (50)$$

$$\sigma_0(\xi(T^0)(q_n, r_n, a, du, 1, du, 0)(q_{n+1}, r_n, a, du, 1)) = (q_{n+1}, r_{n+1}). \quad (51)$$

We define

$$\sigma_0 \bowtie \sigma_1(\xi(T)t_n^U t_n^V) = (q_{n+1}, s_{n+1}) \quad (52)$$

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

Case UUU, $(q_n, r_n, s_n) \in U^q \times U^r \times U^s$: Assume Spoiler chooses the $G(q, s)$ -positions $t_n^U = (q_n, s_n, a, sp, 1, du, 0)$ and $t_n^V = (q_n, s_{n+1}, a, du, 0)$. Let

$$\sigma_1(\xi(T^1)(r_n, s_n, a, sp, 1, du, 0)(r_n, s_{n+1}, a, du, 0)) = (r_{n+1}, s_{n+1}), \quad (53)$$

$$\sigma_0(\xi(T^0)(q_n, r_n, a, sp, 1, du, 0)(q_n, r_{n+1}, a, du, 0)) = (q_{n+1}, r_{n+1}). \quad (54)$$

We define

$$\sigma_0 \bowtie \sigma_1(\xi(T)t_n^U t_n^V) = (q_{n+1}, s_{n+1}) \quad (55)$$

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

Case UEE, $(q_n, r_n, s_n) \in U^q \times E^r \times E^s$: Assume Spoiler chooses the position $t_n^U = (q_n, s_n, a, du, 0, du, 1)$. Let

$$\sigma_0(\xi(T^0)(q_n, r_n, a, du, 0, du, 1)) = (q_{n+1}, r_n, a, du, 1), \quad (56)$$

$$\sigma_0(\xi(T^0)(q_n, r_n, a, du, 0, du, 1)(q_{n+1}, r_n, a, du, 1)) = (q_{n+1}, r_{n+1}), \quad (57)$$

$$\sigma_1(\xi(T^1)(r_n, s_n, a, sp, 0, du, 1)(r_{n+1}, s_n, a, du, 1)) = (r_{n+1}, s_{n+1}). \quad (58)$$

We define

$$\sigma_0 \bowtie \sigma_1(\xi(T)t_n^U) = (q_{n+1}, s_n, a, du, 1), \quad (59)$$

$$\sigma_0 \bowtie \sigma_1(\xi(T)t_n^U(q_{n+1}, s_n, a, du, 1)) = (q_{n+1}, s_{n+1}), \quad (60)$$

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

Case UUE, $(q_n, r_n, s_n) \in U^q \times U^r \times E^s$, and the following Spoiler-chosen $G(q, s)$ -position is $t_n^U = (q_n, s_n, a, du, 0, du, 1)$. Let

$$\sigma_1(\xi(T^1)(r_n, s_n, a, du, 0, du, 1)) = (r_{n+1}, s_n, a, du, 1), \quad (61)$$

$$\sigma_1(\xi(T^1)(r_n, s_n, a, du, 0, du, 1)(r_{n+1}, s_n, a, du, 1)) = (r_{n+1}, s_{n+1}), \quad (62)$$

$$\sigma_0(\xi(T^0)(q_n, r_n, a, sp, 1, du, 0)(q_n, r_{n+1}, a, du, 0)) = (q_{n+1}, r_{n+1}). \quad (63)$$

We define

$$\sigma_0 \bowtie \sigma_1(\xi(T)t_n^U) = (q_{n+1}, s_n, a, du, 1), \quad (64)$$

$$\sigma_0 \bowtie \sigma_1(\xi(T)t_n^U(q_{n+1}, s_n, a, du, 1)) = (q_{n+1}, s_{n+1}), \quad (65)$$

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

Case EEU, $(q_n, r_n, s_n) \in E^q \times E^r \times U^s$: Assume that Spoiler chooses the $G(q, s)$ -positions $t_n^U = (q_n, s_n, a, sp, 0, sp, 1)$ and $t_n^V = (q_{n+1}, s_n, a, sp, 1)$ and $t_{n+1} = (q_{n+1}, s_{n+1})$. Let

$$\sigma_0(\xi(T^0)(q_n, r_n, a, sp, 0, du, 1)(q_{n+1}, r_n, a, du, 1)) = (q_{n+1}, r_{n+1}). \quad (66)$$

We define $(r_i)_{i \leq n+1}$ as the corresponding intermediate 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 $E^q \times U^s$ -state).

Case EEU, $(q_n, r_n, s_n) \in E^q \times U^r \times U^s$, and the following Spoiler-chosen $G(q, s)$ -positions are the three positions defined by $t_n^U = (q_n, s_n, a, sp, 0, sp, 1)$, $t_n^V = (q_{n+1}, s_n, a, sp, 1)$ and $t_{n+1} = (q_{n+1}, s_{n+1})$. Let

$$\sigma_1(\xi(T^1)(r_n, s_n, a, sp, 1, du, 0)(r_n, s_{n+1}, a, du, 0)) = (r_{n+1}, s_{n+1}). \quad (67)$$

We define $(r_i)_{i \leq n+1}$ as the next intermediate 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.

4.2. Fundamental properties of composed strategies and simulation relations

In this section, we will show crucial properties of the simulation relations \leq_{di} , \leq_{de} , \leq_f (summarized as \leq_x) using the concept of a join of two Duplicator strategies, as defined above.

We first 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. Here, we will need the join of two Duplicator strategies, as defined in Section 4.1.

Lemma 2 (Composing winning strategies). *Let $q \in Q$, $r \in R$, and $s \in S$ such that $q \leq_x r$ and $r \leq_x s$. Let σ_0 be a Duplicator strategy for $G^x(q, r)$, and let σ_1 be a Duplicator strategy for $G^x(r, s)$.*

If σ_0 and σ_1 are winning strategies, $\sigma_0 \bowtie \sigma_1$ is a winning strategy (i.e., $q \leq_x r$ and $r \leq_x s$ imply $q \leq_x s$).

Proof. Let σ_0, σ_1 be winning strategies, and let T be a $(\sigma_0 \bowtie \sigma_1)$ -conform play with intermediate sequence $(r_i)_{i < \omega}$. Note that the plays T^0 and T^1 (as defined in Section 4.1) are σ_0 -conform and σ_1 -conform, respectively.

In the case of direct simulation, since T^0 is σ_0 -conform, for every i such that $q_i \in F^q$, we have $r_i \in F^r$. And since T^1 is σ_1 -conform, this implies $s_i \in F^s$, that is, T is a win for Duplicator.

In the case of delayed simulation, for every i such that $q_i \in F^q$, there is a $j_0 \geq i$ such that $r_{j_0} \in F^r$, since T^0 is σ_0 -conform. In turn, by the σ_1 -conformity of T^1 , there is a $j_1 \geq j_0$ such that $s_{j_1} \in F^s$. Hence, T is a win for Duplicator.

Finally, for fair simulation, if there are infinitely many i such that $q_i \in F^q$, the σ_0 -conformity of T^0 ensures that there are also infinitely many j such that $r_j \in F^r$, and the σ_1 -conformity of T^1 then ensures that there are infinitely many l such that $s_l \in F^s$. So, again, T is a win for Duplicator. \square

Corollary 2. *For $x \in \{\text{di}, \text{de}, \text{f}\}$, \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

$$q \equiv_x s \quad \text{iff } q \leq_x s \text{ and } s \leq_x q. \quad (68)$$

By Theorem 1, $q \equiv_x s$ implies $L(\mathbf{Q}(q)) = L(\mathbf{S}(s))$. The relations $\equiv_{\text{di}}, \equiv_{\text{de}}, \equiv_{\text{f}}$ are called *direct, delayed and fair simulation equivalence*, respectively.

While the join of two Duplicator winning strategies is again a winning strategy, the join of two memoryless Duplicator strategies need not be a memoryless strategy.

Lemma 3. *There are Büchi automata $\mathbf{Q}, \mathbf{R}, \mathbf{S}$ such that $\mathbf{Q} \leq_x \mathbf{R} \leq_x \mathbf{S}$ but, for all Duplicator winning strategies σ_0 for $G^x(\mathbf{Q}, \mathbf{R})$ and σ_1 for $G^x(\mathbf{R}, \mathbf{S})$, $\sigma_0 \bowtie \sigma_1$ is not a positional strategy, but, of course, a winning strategy.*

Proof. We give a simple example of such automata for $x \in \{\text{de}, \text{f}\}$; this example can be modified easily so as to work in the case $x = \text{di}$.

Consider the automata $\mathbf{Q}, \mathbf{R}, \mathbf{S}$ (from left to right) of Fig. 3.

The moves of Spoiler and Duplicator in $G^x(\mathbf{Q}, \mathbf{R})$, and hence Duplicator's positional winning strategy σ_0 , are fixed by the structure of the automata (if Spoiler does not want to lose early), i.e., there is only one infinite play of $G^x(\mathbf{Q}, \mathbf{R})$. A positional winning strategy σ_1 for Duplicator in $G^x(\mathbf{R}, \mathbf{S})$ has to satisfy $\sigma_1(r_1, s_I, a, \text{du}, 1) = (r_1, s_I)$ and $\sigma_1(r_2, s_I, a, \text{du}, 1) = (r_2, s_I)$.

Consequently, $\sigma_0 \bowtie \sigma_1$ maps the partial play $\pi = (q_I, s_I) (q_I, s_I, a, \text{sp}, 0, \text{du}, 1) (q_1, s_I, a, \text{du}, 0)$ to (q_1, s_I) , but the partial play $\pi (q_1, s_I) (q_1, s_I, a, \text{sp}, 0, \text{du}, 1) (q_1, s_I, a, \text{du}, 1)$ is mapped to (q_1, s_I) , i.e., $\sigma_0 \bowtie \sigma_1$ is not positional. \square

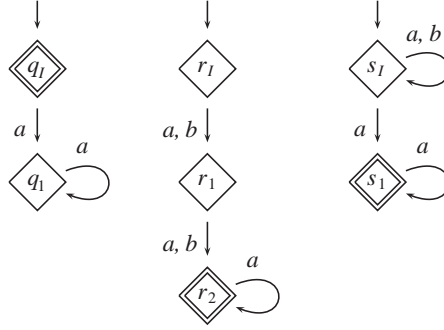


Fig. 3. Joint strategies are not positional.

Fundamental for the further study of \leq_x is the following lemma, which is similar to [9, Lemma 4.1].

Lemma 4. *Let \mathbf{Q}, \mathbf{S} be alternating Büchi automata and let q, s be states of \mathbf{Q} and \mathbf{S} , respectively, such that $q \leq_x s$. Let $a \in \Sigma$.*

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

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

Similar arguments yield the claims for the other three cases, i.e., the case $(q, s) \in U^q \times U^s$ is symmetric, while the arguments for the other cases are as follows. Case $(q, s) \in E^q \times U^s$: 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 $(q, s) \in U^q \times E^s$: 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

In the sequel, we will call a Duplicator strategy σ for a game $G(q_0, s_0) \leq_x$ -respecting if $q \leq_x s$ holds true for every position (q, s) 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 \leq_x -respecting.

The converse is false for $x \in \{\text{de}, \text{f}\}$, as we will see at the beginning of Section 6.

5. Quotienting modulo direct simulation

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

$$(Q/\equiv, \Sigma, [q_I], \Delta', E', U', F/\equiv), \quad (69)$$

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

Furthermore, the following natural constraints must be satisfied:

1. If $([q], a, [q']) \in \Delta'$, then there exist $\hat{q} \equiv q$ and $\bar{q} \equiv q'$ such that $(\hat{q}, a, \bar{q}) \in \Delta$, that is, $\Delta' \subseteq \{([q], a, [q']) \mid (q, a, q') \in \Delta\}$,
2. if $[q] \subseteq E$, then $[q] \in E'$, and
3. if $[q] \subseteq U$, then $[q] \in U'$.

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

In the following, when the considered equivalence relation is direct or delayed simulation equivalence, we will, for instance, write Q_{de} instead of Q/\equiv_{de} and F_{di} instead of F/\equiv_{di} .

A *naive quotient* is a quotient where the converse of the first constraint 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 made neither existential nor universal.

Consider Fig. 4, where an alternating Büchi automaton \mathbf{Q} over $\Sigma = \{a, b\}$ is shown on the left, and the naive x -quotient is shown on the right. For simplicity in notation, we denote the states in the quotients by representatives of the actual equivalence classes, for instance, q_0 on the right stands for $[q_0]$. Note that we have $q_3 \leq_x q_1 \leq_x q_0 \equiv_x q_2$, but $q_3 \not\equiv_x q_1$ and $q_1 \not\equiv_x q_0$ for $x \in \{\text{di}, \text{de}, \text{f}\}$.

The language recognized by the original automaton is $(ba + a)^\omega$, while the naive quotient recognizes Σ^ω . The other possible naive quotient, where the state $[q_0]$ is declared universal, is not equivalent to the original automaton either: that naive quotient only accepts the word a^ω .

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.

5.1. Minimal and maximal successors

To define quotient automata modulo \equiv_x (in fact, for $x = \text{di}$ and de only, since fair quotienting does not preserve the language, see [9]), we will need the notion of maximal and minimal successors of states.

Let $\mathbf{Q} = (Q, \Sigma, q_I, \Delta, E, U, F)$ be an alternating Büchi automaton. Let $q \in Q$, and $a \in \Sigma$. A state $q' \in \Delta(q, a)$ is an *x -maximal a -successor of q* iff $q'' \leq_x q'$ holds for every

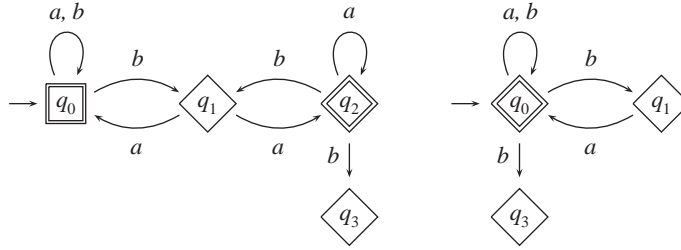


Fig. 4. Naive quotients do not work.

$q'' \in \Delta(q, a)$ with $q' \leq_x q''$. We define

$$\max_a^x(q) = \{q' \in \Delta(q, a) \mid q' \text{ is an } x\text{-maximal } a\text{-successor of } q\}. \quad (70)$$

A state $q' \in \Delta(q, a)$ is an 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

$$\min_a^x(q) = \{q' \in \Delta(q, a) \mid q' \text{ is an } x\text{-minimal } a\text{-successor of } q\}. \quad (71)$$

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

5.2. Minimax quotienting

We can now define a quotient that works for direct simulation, as follows. An x -minimax quotient of \mathbf{Q} is a quotient where the transition relation is given by

$$\begin{aligned} \Delta_x^m = & \{([q], a, [q']) \mid a \in \Sigma, q \in E, q' \in \max_a^x(q)\} \\ & \cup \{([q], a, [q']) \mid a \in \Sigma, q \in U, q' \in \min_a^x(q)\}. \end{aligned} \quad (72)$$

In particular, mixed classes can be declared existential or universal arbitrarily.

We now show that the di-minimax quotient and the original automaton recognize the same language.

We first need some additional insights about maximal successors and the associated strategies.

As a corollary of Lemma 4, we find:

Corollary 3. *Let $q \in Q, s \in S$ be states of alternating Büchi automata \mathbf{Q} and \mathbf{S} such that $q \equiv_x s$. Let $a \in \Sigma$.*

1. *If $(q, s) \in E^q \times E^s$ and $q' \in \max_a^x(q)$, then there is a state $s' \in \max_a^x(s)$ such that $q' \equiv_x s'$.*
2. *If $(q, s) \in U^q \times U^s$ and $q' \in \min_a^x(q)$, then there is a state $s' \in \min_a^x(s)$ such that $q' \equiv_x s'$.*
3. *If $(q, s) \in E^q \times U^s$, then all x -maximal a -successors of q and all x -minimal a -successors of s are x -equivalent.*

Proof. For the first part, let $(q, s) \in E^q \times E^s$ and $q' \in \max_a(q)$. By Lemma 4.1, we find an $s' \in \Delta(s, a)$ such that $q' \leq_x s'$. Let $s'' \in \Delta^s(s, a)$ such that $s' \leq_x s''$. Applying Lemma 4.1 again, there is a $q'' \in \Delta^q(q, a)$ such that $s'' \leq_x q''$, i.e., since q' is an x -maximal a -successor, $q' \leq_x s' \leq_x s'' \leq_x q'' \leq_x q' \leq_x s'$. Hence s' is an x -maximal a -successor of s and satisfies $q' \equiv_x s'$.

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

For the third part, let $(q, s) \in E^q \times U^s$, $q' \in \max_a^x(q)$, $s' \in \min_a^x(s)$. By Lemma 4.2 $q' \leq_x s'$. By Lemma 4.3, there is a state $q'' \in \Delta^q(q, a)$ and a state $s'' \in \Delta^s(s, a)$ such that $s'' \leq_x q''$. Lemma 4.2 shows $q' \leq_x s'' \leq_x q'' \leq_x s'$. But since q' is an x -maximal a -successor, $q'' \leq_x q'$ holds; since s' is an x -minimal a -successor, $s' \leq_x s''$ holds. Hence $q' \equiv_x s'$. So for every $r_0, r_1 \in \min_a^x(s) \cup \max_a^x(q)$, we have $r_0 \equiv_x r_1$, using the transitivity of \equiv_x . \square

This is the reason why mixed classes can be declared existential or universal in the di-minimax quotient: From Corollary 3.3, we can conclude the following.

Remark 2. For a mixed class $M \in Q/\equiv_x$ and $a \in \Sigma$,

$$\begin{aligned} & \{[q'] \mid \exists q(q \in M \cap E \wedge q' \in \max_a(q))\} \\ &= \{[q'] \mid \exists q(q \in M \cap U \wedge q' \in \min_a(q))\}, \end{aligned} \quad (73)$$

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

By Corollary 3, we also have

$$\begin{aligned} \Delta_x^m &= \{([q], a, [q']) \mid a \in \Sigma, q \in E, q' \in \max_a^x(q)\} \\ &\cup \{([q], a, [q']) \mid a \in \Sigma, [q] \subseteq U, q' \in \min_a^x(q)\}. \end{aligned} \quad (74)$$

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

Remark 3. 1. For all $q, q' \in Q$, if $q \leq_{\text{di}} q'$ and $q \in F$, then $q' \in F$.
2. For all $q, q' \in Q$, if $q \equiv_{\text{di}} q'$, then $q \in F$ iff $q' \in F$.

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

Lemma 5. Let $q_0 \leq_{\text{di}} s_0$. In the game $G^{\text{di}}(q_0, s_0)$, every \leq_{di} -respecting strategy for Duplicator is a winning strategy.

Proof. Let $q_0 \leq_{\text{di}} s_0$, and let σ be a \leq_{di} -respecting strategy of Duplicator for $G^{\text{di}}(q_0, s_0)$. Let $T = ((q_i, s_i)_{i < \omega}, w)$ be a σ -conform $G^{\text{di}}(q_0, s_0)$ -protoplay. By assumption, we have $q_i \leq_{\text{di}} s_i$ for every $i \geq 0$; by Remark 3, $s_i \in F^s$ whenever $q_i \in F^q$, 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 $G^x(q_0, s_0)$. We call σ a *minimax strategy* if, for every σ -conform protoplay $T = ((q_i, s_i)_{i < \omega}, w)$ and every $i < \omega$, if $(q_i, s_i) \in U^q \times S$, then $q_{i+1} \in \min_{w(i)}^x(q_i)$, and if $(q_i, s_i) \in Q \times E^s$, then $s_{i+1} \in \max_{w(i)}^x(s_i)$.

We note:

Lemma 6. *Let \mathbf{Q}, \mathbf{S} be alternating Büchi automata. There is a positional strategy σ of Duplicator such that for all $q \in \mathbf{Q}, s \in \mathbf{S}$ where $q \leq_x s$, σ is a \leq_x -respecting minimax strategy for $G^x(q, s)$.*

Proof. Using Lemma 4, such a strategy can easily be defined. \square

Now it is easy to show:

Theorem 2 (Minimax quotients). *Let $\mathbf{Q} = (Q, \Sigma, q_I, \Delta, E, U, F)$ be an alternating Büchi automaton and \mathbf{Q}^m any di-minimax quotient of \mathbf{Q} .*

1. *For all $k_0, q_0 \in Q$ such that $k_0 \leq_{\text{di}} q_0$, $\mathbf{Q}(q_0)$ di-simulates $\mathbf{Q}^m([k_0])$ and $\mathbf{Q}^m([q_0])$ di-simulates $\mathbf{Q}(k_0)$, that is, $[k_0] \leq_{\text{di}} q_0$ and $k_0 \leq_{\text{di}} [q_0]$.*
2. *\mathbf{Q} and \mathbf{Q}^m di-simulate each other, that is, $\mathbf{Q} \equiv_{\text{di}} \mathbf{Q}^m$.*
3. *\mathbf{Q} and \mathbf{Q}^m are equivalent, that is, $L(\mathbf{Q}) = L(\mathbf{Q}^m)$.*

Proof. Mixed classes are deterministic states by Remark 2, so existential choice is the same as universal branching for these states. Hence, it suffices to consider a quotient \mathbf{Q}^m where every mixed class is existential. Also, it is enough to show the first part, the other parts follow immediately from this.

Let $\mathbf{Q}^m = (Q_{\text{di}}, \Sigma, [q_I], \Delta^m, E^m, U^m, F_{\text{di}})$ such that $\Delta^m = \Delta_{\text{di}}^m$, $E^m = \{[q] \in Q_{\text{di}} \mid [q] \cap E \neq \emptyset\}$ and $U^m = Q_{\text{di}} \setminus E^m$. We first show that $\mathbf{Q}(q_0)$ di-simulates $\mathbf{Q}^m([k_0])$. To do so, we define a positional winning strategy σ of Duplicator for $G^{\text{di}}([k_0], q_0)$. First, let σ_{di} be a positional strategy of Duplicator such that σ_{di} is \leq_{di} -respecting and minimax for all games $G^{\text{di}}(q, q')$ where $q, q' \in Q$ and $q \leq_{\text{di}} q'$. Such a strategy exists by Lemma 6.

Further, for every class $[k] \in Q_{\text{di}}$, let $\text{rep}([k])$ be a fixed representative of that class, i.e., $\text{rep}([k]) \in [k]$. We also require that $\text{rep}([k]) \in E$ if $[k] \in E^m$.

We now define σ as follows. For all $k, q \in Q, a \in \Sigma$, let

$$\sigma([k], q, a, \text{du}, 1) = ([k], \text{pr}_2(\sigma_{\text{di}}(\text{rep}([k]), q, a, \text{du}, 1))), \quad (75)$$

$$\begin{aligned} \sigma([k], q, a, \text{du}, 0, \text{du}, 1) \\ = ([\text{pr}_1(\sigma_{\text{di}}(\text{rep}([k]), q, a, \text{du}, 0, \text{du}, 1))], q, a, \text{du}, 1), \end{aligned} \quad (76)$$

$$\sigma([k], q, a, \text{du}, 0) = ([\text{pr}_1(\sigma_{\text{di}}(\text{rep}([k]), q, a, \text{du}, 0))], q). \quad (77)$$

This function is well-defined because σ_{di} is minimax, i.e., the result of σ really is a successor position in $G^{\text{di}}(\mathbf{Q}^m, \mathbf{Q})$.

We now show that σ is, in fact, a winning strategy. Consider a round starting in a position $([k], q) \notin E^m \times U$ such that $k \leq_{\text{di}} q$. Since σ_{di} is a \leq_{di} -respecting minimax strategy, if

Duplicator uses σ in this round, then the next round starts in a position $([k'], q')$ such that $k' \leq_{\text{di}} q'$.

We also consider the case of a round in which Spoiler acts alone, i.e., the round starts in a position of the form $([k], q) \in E^m \times U$ such that $k \leq_{\text{di}} q$. The round continues with the positions $([k], q, a, sp, 0, sp, 1)([k'], q, a, sp, 1)([k'], q')$, that is, there are $\hat{k} \in [k] \cap E$ and $\bar{k} \in [k']$ such that $(\hat{k}, a, \bar{k}) \in \Delta$. Now $k' \leq_{\text{di}} q'$ follows directly by Lemma 4.2.

This shows that σ is a winning strategy of Duplicator for $G^{\text{di}}([k_0], q_0)$, since $k \leq_{\text{di}} q$ holds for every position $([k], q)$ that occurs in a σ -conform play. Note that if we had a position $([k], q)$ occurring in such a play with $([k], q) \in F_{\text{di}} \times (Q \setminus F)$, then we would have $k \not\leq_{\text{de}} q$.

That $\mathbf{Q}^m([q])$ di-simulates $\mathbf{Q}(k)$ can be shown using a symmetrical construction and reasoning. To prove this, we now define a Duplicator winning strategy σ for $G^{\text{di}}(k_0, [q_0])$ as follows. For all $k, q \in Q, a \in \Sigma$, let

$$\sigma(k, [q], a, du, 1) = (k, [\text{pr}_2(\sigma_{\text{di}}(k, \text{rep}([q]), a, du, 1))]), \quad (78)$$

$$\begin{aligned} \sigma(k, [q], a, du, 0, du, 1) \\ = (\text{pr}_1(\sigma_{\text{di}}(k, \text{rep}([q]), a, du, 0, du, 1)), [q], a, du, 1), \end{aligned} \quad (79)$$

$$\sigma(k, [q], a, du, 0) = (\text{pr}_1(\sigma_{\text{di}}(k, \text{rep}([q]), a, du, 0)), [q]). \quad (80)$$

Again, if a round starts in a position $(k, [q]) \notin E \times U^m$ such that $k \leq_{\text{di}} q$ and if Duplicator uses σ in this round, then the next round starts in a position $(k', [q'])$ such that $k' \leq_{\text{di}} q'$. This again follows since σ_{di} is a \leq_{di} -respecting minimax strategy.

Again, we finally consider the case of a round in which Spoiler acts alone, i.e., the last position is of the form $(k, [q]) \in E \times U^m$ such that $k \leq_{\text{di}} q$. The round continues with the positions $(k, [q], a, sp, 0, sp, 1)(k', [q], a, sp, 1)(k', [q'])$, that is, there are $\hat{q} \in [q] \subseteq U$ and $\bar{q} \in [q']$ such that $(\hat{q}, a, \bar{q}) \in \Delta$. Now $k' \leq_{\text{di}} q'$ follows directly by Lemma 4.2.

By an analogous argument as above, it follows that this σ is a Duplicator winning strategy for $G^{\text{di}}(k_0, [q_0])$. \square

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

Corollary 4. *Let $\mathbf{Q} = (Q, \Sigma, q_I, \Delta, E, U, F)$ be an alternating Büchi automaton. Let $\mathbf{Q}' = (Q_{\text{di}}, \Sigma, [q_I], \Delta', E', U', F_{\text{di}})$ be a quotient w.r.t. direct simulation of \mathbf{Q} such that*

- $\Delta_{\text{di}}^m \subseteq \Delta'$,
- $[q] \cap E \neq \emptyset$ implies $[q] \in E'$, and,
- for every $q \in U$ such that $[q] \cap E \neq \emptyset$, if $([q], a, [q']) \in \Delta'$ then there are $\hat{q} \in [q] \cap E$, $\bar{q} \in [q']$ such that $(\hat{q}, a, \bar{q}) \in \Delta$.

Then, \mathbf{Q} and \mathbf{Q}' simulate each other.

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

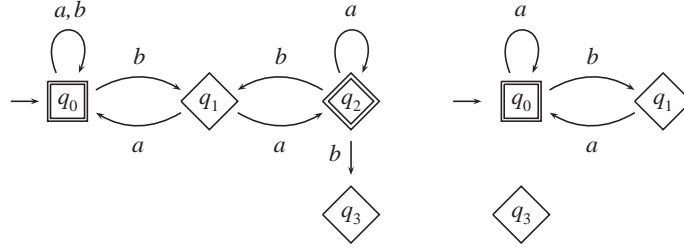


Fig. 5. The automaton of Fig. 4 (left) and its di-minimax quotient (right).

5.3. Example: minimax quotient

As an example, we reconsider the Automaton of Fig. 4. Remember that $q_3 \leq_{\text{di}} q_1 \leq_{\text{di}} q_0 \equiv_{\text{di}} q_2$, but $q_3 \not\equiv_{\text{di}} q_1$ and $q_1 \not\equiv_{\text{di}} q_0$ for this automaton. That is, $\min_b(q_0) = \{q_1\} = \max_b(q_2)$. Fig. 5 shows the resulting di-minimax quotient where the state $[q_0] = [q_2]$ is declared universal.

6. Quotienting modulo delayed simulation

If there is a winning strategy for Duplicator in a game $G^{\text{de}}(q, s)$, there is also a \leq_{de} -respecting minimax strategy (cf. Lemma 6), but this may not necessarily be a winning strategy; it is possible that no minimax strategy is winning. Consider the automaton in Fig. 6.

For $x \in \{\text{de}, \text{f}\}$, we have $q_0 \geq_x q_1$ but not $q_0 \equiv_x q_1$, i.e., $\max_a(q_0) = \{q_0\}$. That is, for a minimax strategy σ of Duplicator, $\sigma(P^*(q_1, q_0, a, \text{du}, 1)) = (q_1, q_0)$ holds. Hence $((q_1, q_0)(q_1, q_0, a, \text{sp}, 0, \text{du}, 1)(q_1, q_0, a, \text{du}, 1))^\omega$ is a σ -conform $G^x(q_1, q_0)$ -play, but not a win for Duplicator. Consequently, the language of any minimax quotient is empty since $\Delta_{\text{de}}^{\text{m}}$ does not contain a transition from $[q_0]_{\text{de}}$ to $[q_1]_{\text{de}}$.

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

6.1. Semi-elective quotienting

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

$$\begin{aligned} \Delta_x^{\text{se}} = & \{([q], a, [q']) \mid (q, a, q') \in \Delta, q \in E\} \\ & \cup \{([q], a, [q']) \mid a \in \Sigma, [q] \subseteq U, q' \in \min_a(q)\}, \end{aligned} \quad (81)$$

and every mixed class is declared existential, i.e., $E_x^{\text{se}} = \{[q] \in Q_x \mid [q] \cap E \neq \emptyset\}$.

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.

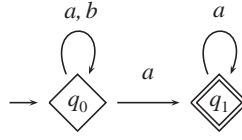


Fig. 6. De-minimax quotients do not work.

By Corollary 3.3, we have

$$\begin{aligned} \Delta_x^{\text{se}} = \{([q], a, [q']) \mid (q, a, q') \in \Delta, q \in E\} \\ \cup \{([q], a, [q']) \mid a \in \Sigma, q \in U, q' \in \min_a(q)\}, \end{aligned} \quad (82)$$

where again mixed classes are existential states.

We will show that \mathbf{Q} and \mathbf{Q}_x^{se} simulate each other. For $x = \text{di}$, this follows immediately from Corollary 4, i.e.:

Corollary 5. *For every alternating Büchi automaton \mathbf{Q} , the automata \mathbf{Q} and $\mathbf{Q}_{\text{di}}^{\text{se}}$ simulate each other, in particular, $L(\mathbf{Q}) = L(\mathbf{Q}_{\text{di}}^{\text{se}})$.*

Note that usually the minimax quotient has less transitions than the semi-elective quotient. That is, for direct simulation, the minimax quotient is the better choice, because it is advantageous to also minimize the number of transitions. This is especially important because states can become unreachable and can thus be deleted as a result of such a minimization.

The more complicated case, where $x = \text{de}$, is treated in the following sections.

6.2. \mathbf{Q} simulates $\mathbf{Q}_{\text{de}}^{\text{se}}$

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 $G(q, s)$; the winning condition is assumed to be trivial in the sense that if no early loss occurs, Duplicator wins. That is, the basic simulation game is a simulation game in the sense of Section 3.1 with winning condition P^ω .

We may extend this observation to a basic simulation game $G(K_0, q_0)$ where K_0 is a state of the quotient automaton $\mathbf{Q}_{\text{de}}^{\text{se}}$ such that $k_0 \leq_{\text{de}} q_0$ holds for some $k_0 \in K_0$, which we write as $K_0 \sqsubseteq_{\text{de}} q_0$:

Corollary 6. *For all $K_0 \in \mathbf{Q}_{\text{de}}$ and for all $q_0 \in \mathbf{Q}$ such that $K_0 \sqsubseteq_{\text{de}} q_0$, there is a minimax strategy σ of Duplicator for $G(K_0, q_0)$ such that, for all Spoiler strategies τ for $G(K_0, q_0)$, the (τ, σ) -conform protoplay $((K_i, q_i)_{i < \omega}, w)$ satisfies $K_i \sqsubseteq_{\text{de}} q_i$ for every $i < \omega$.*

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

Proof. Let $K_0 \in \mathbf{Q}_{\text{de}}$, $q_0 \in \mathbf{Q}$. Let T_i be a prefix of a $G(K_0, q_0)$ -play such that the last position of T_i is a P_1 -position such that $K_i \sqsubseteq_{\text{de}} q_i$. Again, we make a case distinction.

In the first case, if $(K_i, q_i)(K_i, q_i, a, sp, 0, du, 1)(K_{i+1}, q_i, a, du, 1)$ is a suffix of T (hence $K_i \in E^{\text{se}}$), we find $k_i \in K_i \cap E$ and $k_{i+1} \in \Delta(k_i, a) \cap K_{i+1}$.

By Lemma 4.1, the set $\{q' \in \Delta(q_i, a) \mid k_{i+1} \leq_{\text{de}} q'\}$ is not empty. We choose a \leq_{de} -maximal element q_{i+1} of this set (which is an element of $\max_a^{\text{de}}(q_i)$) and define $\sigma(T) = (K_{i+1}, q_{i+1})$. Hence $K_{i+1} \sqsubseteq_{\text{de}} q_{i+1}$.

In the other cases, the suffixes are of the form $(K_i, q_i, a, du, 0, du1)$, of the form $(K_i, q_{i+1}, a, du, 0)$, or of the form $(K_i, q_i)(K_i, q_i, a, du, 0, du, 1)(K_{i+1}, q_i, a, du, 1)$ where K_{i+1} is chosen such that there is a $q' \in \Delta(q_i, a)$ satisfying $K_{i+1} \sqsubseteq_{\text{de}} q'$. These cases are also treated using Lemma 4, i.e., by Lemmas 4.3 and 4.4, 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. Note that the case $(K_i, q_i) \in E^{\text{se}} \times U$, where Duplicator does not move in the following round, can again be treated by Lemma 4.2. \square

Moreover, we can show that the join of such a \sqsubseteq_{de} -respecting minimax strategy and a Duplicator winning strategy is again \sqsubseteq_{de} -respecting.

Corollary 7. *Let $K_0 \in Q_{\text{de}}$, $q_0 \in Q$ such that $K_0 \sqsubseteq_{\text{de}} q_0$, and $s_0 \in S$ such that $q_0 \leq_{\text{de}} s_0$. Let σ be a \sqsubseteq_{de} -respecting minimax strategy for Duplicator in $G(K_0, q_0)$ and let σ^{de} be a Duplicator winning strategy for $G^{\text{de}}(q_0, s_0)$.*

Then $\sigma \bowtie \sigma^{\text{de}}$ is a \sqsubseteq_{de} -respecting strategy for $G(K_0, s_0)$.

Proof. Let τ be some Spoiler strategy for $G^{\text{de}}(K_0, s_0)$, and let $T = ((t_j)_{j < \omega}, w)$ be the $(\tau, \sigma \bowtie \sigma^{\text{de}})$ -conform protoplay. Initially, we have $K_0 \sqsubseteq_{\text{de}} q_0 \leq_{\text{de}} s_0$, hence $K_0 \sqsubseteq_{\text{de}} s_0$.

Now let $i \in \omega$, and $T_i = ((t_j)_{j \leq i}, w[0..i - 1])$ be the prefix of T of length $i + 1$. Let $t_i = (K_i, s_i)$, and let $(q_j)_{j \leq i}$ be the intermediate sequence of T_i . Assume $K_i \sqsubseteq_{\text{de}} q_i \leq_{\text{de}} s_i$.

We show that $K_{i+1} \sqsubseteq_{\text{de}} q_{i+1} \leq_{\text{de}} s_{i+1}$ holds for the next $(Q_{\text{de}} \times S)$ -position $t_{i+1} = (K_{i+1}, s_{i+1})$ of T and the next state of the intermediate sequence, distinguishing four cases.

In the first case, let $K_i \subseteq U^q$, $s_i \in U^s$. Let $t_i^U = \tau(\xi(T_i)) = (K_i, s_i, a, sp, 1, du, 0)$ and $t_i^V = \tau(\xi(T_i)t_i^U) = (K_i, s_{i+1}, a, du, 0)$. Let $\sigma \bowtie \sigma^{\text{de}}(\xi(T_i)t_i^U t_i^V) = (K_{i+1}, s_{i+1})$, and let q_{i+1} be the next state of the intermediate sequence according to Section 4.

If $q_i \in E^q$, the definition of $\sigma \bowtie \sigma^{\text{de}}$ implies $K_{i+1} \leq_{\text{de}} q_{i+1}$, since $\xi(T^0)$ is σ -conform (both K_{i+1} and q_{i+1} are chosen according to σ). And $q_{i+1} \leq_{\text{de}} s_{i+1}$ by Lemma 4, since $q_i \leq_{\text{de}} s_i$ and $(q_i, s_i) \in E^q \times U^s$. Hence $K_{i+1} \leq_{\text{de}} s_{i+1}$.

If $q_i \in U^q$, the definition of $\sigma \bowtie \sigma^{\text{de}}$ also implies $K_{i+1} \leq_{\text{de}} s_{i+1}$, since $\xi(T^1)$ is σ^{de} -conform (q_{i+1} is chosen according to σ^{de} , hence $q_{i+1} \leq_{\text{de}} s_{i+1}$). Because $\xi(T_{i+1}^1)$ is σ -conform (i.e., K_{i+1} is chosen according to σ), we have $K_{i+1} \leq_{\text{de}} q_{i+1} \leq_{\text{de}} s_{i+1}$.

The other cases are shown analogously, i.e., the case $K_i \cap E^q \neq \emptyset$, $s_i \in E^s$ is symmetric to $K_i \subseteq U^q$, $s_i \in U^s$, and in the cases $K_i \cap E^q \neq \emptyset$, $s_i \in U^s$ and $K_i \subseteq U^q$, $s_i \in E^s$, the desired property also results from the definition of $\sigma \bowtie \sigma^{\text{de}}$ together with Lemma 4. \square

And we can easily verify the following.

Lemma 7. *Let $K_0, q_0, s_0, \sigma, \sigma^{\text{de}}$ be chosen like in Corollary 7.*

For every Spoiler strategy τ in $G^{\text{de}}(K_0, s_0)$, $q_0 \in F^q$ implies that the $(\tau, \sigma \bowtie \sigma^{\text{de}})$ -conform play contains a position $(K_j, s_j) \in Q_{\text{de}} \times F^s$, i.e., $\sigma \bowtie \sigma^{\text{de}}$ is a winning strategy for Duplicator in $G(K_0, s_0)$ with winning set $\{u \in P^\omega \mid \exists i (u_i \in Q_{\text{de}} \times F^s)\}$.

Proof. Let τ be a Spoiler strategy for $G^{\text{de}}(K_0, s_0)$, and let $q_0 \in F^q$. Let $T = ((t_i)_{i < \omega}, w)$ be the $(\tau, \sigma \bowtie \sigma^{\text{de}})$ -conform protoplay, and assume that there is no $i \in \omega$ such that $t_i = (K_i, s_i) \in Q_{\text{de}} \times F^s$. Since T is $\sigma \bowtie \sigma^{\text{de}}$ -conform, the play T^1 (as defined in Section 4) is σ^{de} -conform. But T^1 is not a win for Duplicator, in contradiction to σ^{de} being a winning strategy for Duplicator. Hence there must be a position $t_i = (K_i, s_i)$ in T such that $s_i \in F^s$. \square

We are now ready to show Theorem 3, stating that in fact an alternating Büchi automaton simulates its semi-elective quotient w.r.t. delayed simulation. The idea of the proof is that, in order to win the respective simulation game, Duplicator uses the join of a \sqsubseteq_{de} -respecting strategy and a winning strategy. But this joint strategy is only \sqsubseteq_{de} -respecting and not necessarily a winning strategy: The intermediate sequence may miss the accepting representatives of the states of the quotient automaton, so that Duplicator may stick to a merely \sqsubseteq_{de} -respecting strategy.

As a remedy, we define the Duplicator strategy as a modified join of the two strategies such that Duplicator is forced to reach for accepting states when necessary.

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

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

1. for every $K \in Q_{\text{de}}$, a representative $\text{rep}(K) \in K$ such that if $K \cap F \neq \emptyset$ then $\text{rep}(K) \in F$,
2. for every $(K, q) \in Q_{\text{de}} \times Q$ such that $K \sqsubseteq_{\text{de}} q$, a \sqsubseteq_{de} -respecting minimax strategy σ_{Kq}^o of Duplicator for $G(K, q)$ (by Corollary 6, there is such a strategy), and
3. for every $(k, q) \in Q \times Q$ such that $k \leq_{\text{de}} q$, a winning strategy σ_{kq}^{de} of Duplicator for $G^{\text{de}}(k, q)$.

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

$$j = \min\{i \leq n \mid (K_i, q_i) \in F_{\text{de}} \times (Q \setminus F) \wedge \forall i' (i \leq i' \leq n \rightarrow q_{i'} \notin F)\}, \quad (83)$$

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

$$\sigma(T_i) := \sigma_{K_j \text{ rep}(K_j)}^o \bowtie \sigma_{\text{rep}(K_j) q_j}^{\text{de}}(T_{[j,i]}). \quad (84)$$

By Corollary 7, σ is \sqsubseteq_{de} -respecting. Now if $t_i = (K_i, q_i)$ is the first $(F_{\text{de}} \times (Q \setminus F))$ -position after the last $(Q_{\text{de}} \times F)$ -position (or the first $(F_{\text{de}} \times (Q \setminus F))$ -position at all), we have $K_i \sqsubseteq_{\text{de}} q_i$. The strategy σ is updated to $\sigma_{K_i \text{ rep}(K_i)}^o \bowtie \sigma_{\text{rep}(K_i) q_i}^{\text{de}}$ where $\text{rep}(K_i) \in K_i \cap F$, and only the suffix starting with (K_i, q_i) of the play is taken into account for the following moves of Duplicator. (Remember that a joint strategy cannot be assumed to be positional.)

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

6.3. $Q_{\text{de}}^{\text{se}}$ simulates Q

Theorem 3 states that $\mathbf{Q}(q)$ de-simulates $\mathbf{Q}_{\text{de}}^{\text{se}}([k])$. We also want to show that $\mathbf{Q}_{\text{de}}^{\text{se}}([q])$ de-simulates $\mathbf{Q}(k)$. 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, ensuring that the intermediate sequence is a path in the sequence of second state components in the plays of $G^{\text{de}}(k, [q])$.

We start with the following corollary, a direct consequence of the construction of $\mathbf{Q}_{\text{de}}^{\text{se}}$ together with Corollary 3.

Corollary 8. *Let $q'_0 \in [q_0]$. There is a Duplicator strategy σ^{\equiv} for $G^{\text{de}}(q'_0, [q_0])$ such that, for every $Q \times Q_{\text{de}}$ -position $(q'_i, [q_i])$ of a σ^{\equiv} -conform play, $q'_i \in [q_i]$ 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 $G^{\text{de}}(k_0, q_0)$ and a \equiv_{de} -respecting strategy for $G^{\text{de}}(q_0, [q_0])$ is a winning strategy for $G^{\text{de}}(k_0, [q_0])$.

Theorem 4. *Let \mathbf{Q} be a Büchi automaton with states k_0, q_0 such that $k_0 \leq_{\text{de}} q_0$. The automaton $\mathbf{Q}_{\text{de}}^{\text{se}}([q_0])$ de-simulates $\mathbf{Q}(k_0)$, i.e., there is a winning strategy for Duplicator in $G^{\text{de}}(k_0, [q_0])$.*

Proof. Let σ^{de} be a winning strategy of Duplicator for $G^{\text{de}}(k_0, q_0)$, and let σ^{\equiv} be a \equiv_{de} -respecting Duplicator strategy for $G^{\text{de}}(q_0, [q_0])$. We show that $\sigma^{\text{de}} \bowtie \sigma^{\equiv}$ is a Duplicator winning strategy for $G^{\text{de}}(k_0, [q_0])$.

Let τ be a Spoiler strategy for $G^{\text{de}}(k_0, [q_0])$. Let $T = (t_i)_{i < \omega}$ be the $(\tau, \sigma^{\text{de}} \bowtie \sigma^{\equiv})$ -conform protoplay with the intermediate sequence $(q'_i)_{i < \omega}$.

Since $\xi(T^0)$ is σ^{de} -conform, there is, for every $i < \omega$ such that $\text{pr}_1(t_i) \in F$, a $j \geq i$ such that $q'_j \in F$. Since $\xi(T^1)$ is σ^{\equiv} -conform, we have $q'_j \in \text{pr}_2(t_j)$, hence $t_j \in Q \times F_{\text{de}}$. Consequently, $\sigma^{\text{de}} \bowtie \sigma^{\equiv}$ is a winning strategy. \square

Theorems 3 and 4 yield:

Theorem 5 (Semi-elective quotients). *For every alternating Büchi automaton \mathbf{Q} , the automata \mathbf{Q} and $\mathbf{Q}_{\text{de}}^{\text{se}}$ de-simulate each other, in particular, $L(\mathbf{Q}) = L(\mathbf{Q}_{\text{de}}^{\text{se}})$.*

6.4. Remarks and possible optimizations

In the construction of the quotient automaton, a transition $(q_u, a, q') \in \Delta$ with $q_u \in U$ only results in a transition $([q_u]_{\text{de}}, a, [q']_{\text{de}}) \in \Delta_{\text{de}}^{\text{se}}$ if $q' \in \min_a(q_u)$, even if $[q_u]_{\text{de}}$ 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 of Fig. 1 again, and remember that the alphabet is $\{a, b\}$. We have $q_0 \equiv_{\text{de}} q_1 >_{\text{de}} q_2$. So a quotient construction preserving nonminimal successors of universal

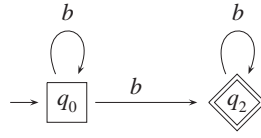


Fig. 7. A quotient of the automaton in Fig. 1.

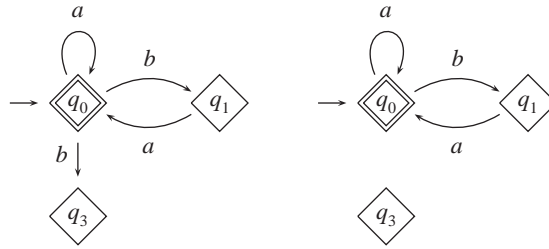


Fig. 8. The semi-elective and optimized semi-elective quotients of the automaton of Figs. 4 and 5.

states would result in the automaton given in Fig. 7. But the original Automaton accepts b^ω whereas the quotient does not; in the semi-elective quotient w.r.t. delayed simulation, there is no edge from the state $[q_0]$ to itself.

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

Remark 4. Let $\mathbf{Q}_{\text{de}}^{\text{se}'}$ be the quotient which is defined just as $\mathbf{Q}_{\text{de}}^{\text{se}}$ but with the transition relation given by

$$\begin{aligned} \Delta_{\text{de}}^{\text{se}'} = \{ & ([q], a, [q']) \mid (q, a, q') \in \Delta \wedge q \in E \wedge ([q] \cap F \neq \emptyset \rightarrow q' \in \max_a(q)) \} \\ & \cup \{ ([q], a, [q']) \mid a \in \Sigma, [q] \subseteq U, q' \in \min_a(q) \}. \end{aligned} \quad (85)$$

Then $\mathbf{Q}_{\text{de}}^{\text{se}'}$ de-simulates \mathbf{Q} .

Proof. In a delayed simulation game, Duplicator can stick to a \leq_{de} -respecting minimax strategy until the play reaches an $(F \times (Q \setminus F))$ -position, in which case he may be forced, in order to win, to switch to another strategy until the play reaches a $(Q \times F)$ -position (cf. the proof of Theorem 3). That is, we may assume that a Duplicator winning strategy behaves like a \leq_{de} -respecting minimax strategy at all $(Q \times F)$ -positions. Hence, only de-maximal successors are necessary at accepting existential states. \square

In other words, if an existential state is de-equivalent to an accepting state, its transitions to non-de-maximal successor states are superfluous.

As a simple example, consider the automaton of Figs. 4 and 5 once again. The semi-elective quotient of this automaton is shown on the left-hand side of Fig. 8, while the quotient defined according to Remark 4 is shown on the right-hand side.

State $[q_3]$ is disconnected from state $[q_0]$ on the right-hand side because state $[q_0]$ is an accepting state.

Thus, a valid strategy for reducing the number of transitions is extending the set of accepting states without changing the simulation relation. One way how this can be carried out is explained in what follows. A necessary condition for the de-equivalence of a state to an accepting state is its de-equivalence to an accepting copy of itself, as defined below (without proof). By checking this equivalence to an accepting copy, we can also identify states which are not equivalent to an actual accepting state in the original automaton, but which can be declared accepting without changing their status w.r.t. \leq_{de} .

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

We define

$$PF_{\text{de}} = \{q \in Q \mid q' \leq_{\text{de}} q\}. \quad (86)$$

The elements of PF_{de} are called *pseudo-accepting states*. Note that $F \subseteq PF_{\text{de}}$. We define

$$\mathbf{Q}_{PF} = (Q, \Sigma, q_I, \Delta, E, U, PF_{\text{de}}). \quad (87)$$

Lemma 8. *For every alternating Büchi automaton $\mathbf{Q} = (Q, \Sigma, q_I, \Delta, E, U, F)$, we have $\mathbf{Q} \equiv_{\text{de}} \mathbf{Q}_{PF}$.*

Proof. Obviously, $\mathbf{Q} \leq_{\text{de}} \mathbf{Q}_{PF}$. Conversely, let σ_q be a Duplicator winning strategy for $G^{\text{de}}(q', q)$ for every $q \in PF_{\text{de}} \setminus F$, where $q' \in PF_{\text{de}}$ is the copy of q in \mathbf{Q}_{PF} . Let $\sigma_{q\bar{q}}$ be a winning strategy for $G^{\text{de}}(q, \bar{q})$ for every pair of states $(q, \bar{q}) \in Q \times Q$ such that $q \leq_{\text{de}} \bar{q}$.

To win the game $G^{\text{de}}(\mathbf{Q}_{PF}, \mathbf{Q})$, Duplicator starts with the strategy $\sigma = \sigma_{q_I q_I}$, but whenever the play reaches a position (k, q) such that $k \in PF_{\text{de}} \setminus F$, $q \in Q \setminus F$, Duplicator switches to the strategy $\sigma_k \bowtie \sigma$. He then uses this strategy until the play reaches a position $(q, \bar{q}) \in Q \times F$; this is guaranteed to happen since both σ_k and σ are winning strategies, and it is also guaranteed that $q \leq_{\text{de}} \bar{q}$ holds. At this point, Duplicator changes his strategy to $\sigma = \sigma_{q\bar{q}}$ and continues with that strategy until the play reaches a position in $(PF_{\text{de}} \setminus F) \times (Q \setminus F)$ once again, which again forces him to switch his strategy as explained above. By Lemma 2, this strategy is winning. \square

In summary, when computing the semi-elective quotient, we may treat existential pseudo-accepting states like accepting states, i.e., remove their nonmaximal successors. This and similar simple modifications can greatly enhance the performance of an implementation, see [10].

6.5. Example: semi-elective quotient

As an example of the construction of the semi-elective quotient automaton modulo delayed simulation, consider Fig. 9.

For the automaton \mathbf{Q} on the left, we have $q_2 <_{\text{de}} q_1 \equiv_{\text{de}} q_5 <_{\text{de}} q_0 \equiv_{\text{de}} q_3 <_{\text{de}} q_4$. Thus there are four states in the quotient automaton $\mathbf{Q}_{\text{de}}^{\text{se}}$ on the right. Since $\min_b(q_1) = \{q_2\}$, the

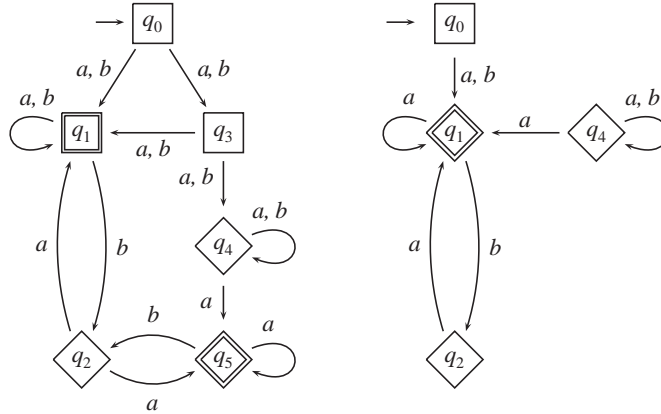


Fig. 9. Automaton and de-semi-elective quotient.

edge $([q_1], b, [q_1])$ is not in $\Delta_{\text{de}}^{\text{se}}$ (cf. (82)); since $\min_a(q_0) = \min_b(q_0) = \{q_1\}$, there is no edge $([q_0], c, [q_3])$ in $\Delta_{\text{de}}^{\text{se}}$ with $c \in \{a, b\}$. And since $\min_a(q_3) = \min_b(q_3) = \{q_1\}$, there is no edge $([q_3], c, [q_4])$ in $\Delta_{\text{de}}^{\text{se}}$ with $c \in \{a, b\}$. Consequently, the state $[q_4]$ is not reachable in $\mathbf{Q}_{\text{de}}^{\text{se}}$ and should be removed in a successive optimization of the quotient automaton.

7. From alternating Büchi automata to nondeterministic Büchi automata

Given an alternating Büchi automaton \mathbf{Q} , the standard approach for constructing an equivalent nondeterministic (i.e., nonalternating) Büchi automaton is the construction of Miyano and Hayashi [25].

In this section, we will show that our simulation relations are compatible with the Miyano–Hayashi construction. That is, if an ABA \mathbf{Q} is simulated by an ABA \mathbf{S} , the same holds true for their nondeterministic versions resulting from the Miyano–Hayashi construction. We conclude that our simulation quotienting can be applied to the alternating automaton prior to the Miyano–Hayashi construction without changing its status w.r.t. the simulation relation. This is of practical importance since we can further conclude that our simulation relations can be used for on-the-fly simplifications during the Miyano–Hayashi construction. (However, a subsequent simulation quotienting usually will still improve the result.) Traditionally, simulation quotienting and simulation-based simplifications are only applied to the nondeterministic automaton.

Fig. 10 shows these two possible ways from an ABA to a nondeterministic automaton (NBA).

From a practical point of view, applying simulation quotienting to the alternating automaton is relatively cheap compared to simulation quotienting for the nondeterministic automaton (cf. Section 8), since the MH-construction incurs an exponential growth (see below). For this reason, a state space reduction of the alternating automaton often results in a substantial reduction of the size of the nondeterministic automaton. Aside from these

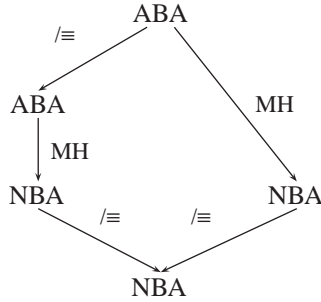


Fig. 10. Two ways from alternating BA to nondeterministic BA (the symbol $/\equiv$ stands for quotienting).

savings in the state space, a smaller intermediate automaton speeds up a subsequent simulation quotienting.

We will now give a short summary of the MH-construction. The construction of Miyano and Hayashi for converting an ABA into a nondeterministic automaton is a subset construction modified for de-universalization instead of determinization. The states are *pairs* (M, N) of subsets of the state set Q . The first component is used in a similar fashion as in the normal subset construction, that is, if there is a universal state q_u in the first component M with a -successors q' and q'' , and (M', N') is an a -successor state of (M, N) then $\{q', q''\} \subseteq M'$. The second component is used to keep track of computation branches with an obligation to reach an accepting state, i.e., $N \subseteq M$ and N is disjoint to the set of accepting states F , because a state is deleted from N as soon as its computation branch reaches an accepting state. Especially, (M, N) is accepting if the second component is the empty set, and if (M', N') is a successor state of (M, \emptyset) then $N' = M' \setminus F$.

We will call the automaton resulting from the Miyano–Hayashi construction the MH-automaton and denote it by \mathbf{Q}_{nd} , i.e., \mathbf{Q}_{nd} is a nondeterministic Büchi automaton such that $L(\mathbf{Q}) = L(\mathbf{Q}_{\text{nd}})$. Note that \mathbf{Q}_{nd} is an exponential size automaton in the number of states of \mathbf{Q} (and that this is necessarily so in the worst case).

Proposition 2. Let $\mathbf{Q} = (Q, \Sigma, q_I, \Delta^q, E^q, U^q, F^q)$ and $\mathbf{S} = (S, \Sigma, s_I, \Delta^s, E^s, U^s, F^s)$ be alternating Büchi automata, and let $x \in \{\text{di}, \text{de}, \text{f}\}$. If $\mathbf{Q} \leq_x \mathbf{S}$, then $\mathbf{Q}_{\text{nd}} \leq_x \mathbf{S}_{\text{nd}}$.

Proof. The proof is somewhat similar to the construction of a joint strategy (Section 4) and the proof of Lemma 2. Let σ be a Duplicator winning strategy for $G^x(\mathbf{Q}, \mathbf{S})$. We will now simultaneously and inductively construct a Duplicator strategy σ' for $G^x(\mathbf{Q}_{\text{nd}}, \mathbf{S}_{\text{nd}})$ and a set of σ -conform $G^x(\mathbf{Q}, \mathbf{S})$ -protoplays L . This set is called the *logbook* of the partial σ' -conform play.

Remember that, in a $G^x(\mathbf{Q}_{\text{nd}}, \mathbf{S}_{\text{nd}})$ -protoplay $((P_i)_{i < n}, w)$, the positions P_i are pairs consisting of a state of \mathbf{Q}_{nd} and a state of \mathbf{S}_{nd} , and these states in turn are pairs of subsets of Q and S , respectively. Hence, every such position P_i is of the form $((M_i^q, N_i^q), (M_i^s, N_i^s))$ where $N_i^q \subseteq M_i^q \subseteq Q$ and $N_i^s \subseteq M_i^s \subseteq S$.

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 $G^x(\mathbf{Q}, \mathbf{S})$ -protoplays over the word w .

2. For every $s \in M_i^s$, there is a $q \in M_i^q$ such that (q, s) is the $(i + 1)$ th position of an element of L_{n-1} , and, conversely,
 3. if (q, s) is the $(i + 1)$ th position of an element of L_{n-1} , then $q \in M_i^q, s \in M_i^s$.
- Initially, for the protoplay $((\{q_I\}, \{q_I\} \setminus F^q), (\{s_I\}, \{s_I\} \setminus F^s), \varepsilon)$ of length 1, $L_0 = \{((q_I, s_I), \varepsilon)\}$ is a valid logbook.

Now let $T_n = ((P_i)_{i < n}, w)$ be a partial σ' -conform $G^x(\mathbf{Q}_{\text{nd}}, \mathbf{S}_{\text{nd}})$ -protoplay with logbook L_{n-1} , and assume Spoiler chooses in $\xi(T_n)$ the position $t'_n = ((M_n^q, N_n^q), (M_{n-1}^s, N_{n-1}^s), a)$. (Since \mathbf{Q}_{nd} and \mathbf{S}_{nd} are nondeterministic automata, we may assume that Spoiler chooses a letter and a state simultaneously, cf. [9].)

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_n^s , since N_n^s is determined by this choice. For every protoplay $K_{n-1} = ((q_i, s_i)_{i < n}, w) \in L_{n-1}$, we distinguish the following four cases. Note that $q_{n-1} \in M_{n-1}^q$ and $s_{n-1} \in M_{n-1}^s$ by the logbook property.

First case: $(q_{n-1}, s_{n-1}) \in E^q \times E^s$. Then, there is a $q_n \in M_n^q$ such that $(q_{n-1}, a, q_n) \in \Delta^q$. Let

$$\sigma(\xi(K_{n-1})(q_{n-1}, s_{n-1}, a, sp, 0, du, 1)(q_n, s_{n-1}, a, du, 1)) = (q_n, s_n). \quad (88)$$

We add s_n to M_n^s and $K_n = ((q_i, s_i)_{i < n+1}, wa)$ to the logbook L_n .

Second case: $(q_{n-1}, s_{n-1}) \in U^q \times E^s$. Then, $\Delta^q(q_{n-1}, a) \subseteq M_n^q$. Let

$$\sigma(\xi(K_{n-1})(q_{n-1}, s_{n-1}, a, du, 0, du, 1)) = (q_n, s_{n-1}, a, du, 1) \quad (89)$$

and

$$\sigma(\xi(K_{n-1})(q_{n-1}, s_{n-1}, a, du, 0, du, 1)(q_n, s_{n-1}, a, du, 1)) = (q_n, s_n). \quad (90)$$

We add s_n to M_n^s and $K_n = ((q_i, s_i)_{i < n+1}, wa)$ to the logbook L_n .

Third case: $(q_{n-1}, s_{n-1}) \in E^q \times U^s$. Then, there is a $q_n \in M_n^q$ such that $(q_{n-1}, a, q_n) \in \Delta^q$, and it must be the case that $\Delta^s(s_{n-1}, a) \subseteq M_n^s$. For every $s_n \in \Delta^s(s_{n-1}, a)$, we add the protoplay $((q_i, s_i)_{i < n+1}, wa)$ to the logbook L_n .

Fourth case: $(q_{n-1}, s_{n-1}) \in U^q \times U^s$. Then, $\Delta^q(q_{n-1}, a) \subseteq M_n^q$, and it must be the case that $\Delta^s(s_{n-1}, a) \subseteq M_n^s$. For every $s_n \in \Delta^s(s_{n-1}, a)$, let

$$\sigma(\xi(K_{n-1})(q_{n-1}, s_{n-1}, a, sp, 1, du, 0)(q_{n-1}, s_n, a, du, 0)) = (q_n, s_n); \quad (91)$$

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

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

It is easy to check that L_n again has the logbook property and that σ' is a Duplicator strategy for $G^x(\mathbf{Q}_{\text{nd}}, \mathbf{S}_{\text{nd}})$. We show that σ' is in fact a winning strategy. In the case $x = \text{de}$, suppose that Spoiler reaches an accepting state (M_m^q, \emptyset) in the m th turn of a $G^{\text{de}}(\mathbf{Q}_{\text{nd}}, \mathbf{S}_{\text{nd}})$ -play π such that Duplicator is in a nonaccepting state (M_m^s, N_m^s) , i.e., $N_m^s \neq \emptyset$. Since $N_m^s \subseteq M_m^s$, by the logbook property there is, for every $s \in N_m^s$, a $q \in M_m^q$ such that (q, s) is the current position of a protoplay in the logbook L_m to π such that, in this protoplay, Duplicator has the obligation to reach an accepting state in the second component in order to win. 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 (q, s) is the m th position of P and Duplicator has to reach an accepting state in the second component in order to win P , then there is an $l \in \{m + 1, \dots, m'\}$ such that the l th position of P is of the form (q', s') and $s' \in F^s$. By the above definition of σ' , this implies that the m' th Duplicator state in π is of the form $(M_{m'}^s, \emptyset)$, i.e., an accepting state.

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

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

The following corollary follows immediately from the proof of Proposition 2.

Corollary 9. *Let $x \in \{\text{di}, \text{de}\}$. Let \mathbf{Q} be an ABA, and let $(M_0, N_0), (M_1, N_1)$ be two states of \mathbf{Q}_{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, we can remove the non- x -minimal elements of the subsets in the states of \mathbf{Q}_{nd} .

Corollary 10. *For every ABA \mathbf{Q} and $x \in \{\text{di}, \text{de}\}$, $((\mathbf{Q}^x)_{\text{nd}})^x \equiv_x (\mathbf{Q}_{\text{nd}})^x$ holds.*

Proof. We have $\mathbf{Q}^x \equiv_x \mathbf{Q}$, so by Proposition 2, $(\mathbf{Q}^x)_{\text{nd}} \equiv_x \mathbf{Q}_{\text{nd}}$ holds, so $((\mathbf{Q}^x)_{\text{nd}})^x \equiv_x (\mathbf{Q}_{\text{nd}})^x$ follows immediately. \square

That is, the original alternating automaton, the intermediate automata of Fig. 10 and the resulting nondeterministic Büchi automaton are all simulation equivalent.

Moreover, optimizations using Corollary 9 can be applied on-the-fly, that is, after the construction of every single state of the MH-automaton [10].

But note that the simulation quotients of simulation equivalent (alternating or nondeterministic) automata need not be isomorphic: Let $\mathbf{Q}^{\text{de}+}$ denote the de-semi-elective quotient of \mathbf{Q} optimized using pseudo-accepting states as described in Section 6.4. Then, $((\mathbf{Q}^{\text{de}+})_{\text{nd}})^{\text{de}+}$, the result of taking the left-hand way in Fig. 10, is, for certain instances, smaller than $(\mathbf{Q}_{\text{nd}})^{\text{de}+}$, the result of the right-hand way. This is because a state (M, N) of \mathbf{Q}_{nd} is pseudo-accepting only if all elements of N are pseudo-accepting. (Without additional optimizations, the left-hand quotient will not be smaller than the right-hand quotient.)

8. Efficient algorithms

Efficient algorithms for computing simulation relations of nondeterministic Büchi automata are given in [9]. We use the same ideas with minor modifications and adjustments for computing simulation relations of alternating Büchi automata. This is explained in the first and third section, while in the second section, we prove part 1 of Proposition 1, which we had postponed earlier. In the fourth section, 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.

8.1. Modifications for the delayed simulation game

For direct and fair simulation, the winning conditions of the corresponding games can be phrased as parity 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 [9]. The crucial information for the players of a delayed simulation game is whether the play has already visited a position in $\hat{F}^q \cap \bar{F}^s$ without having visited a \hat{F}^s -position since or not (cf. Section 3.1). Following [9], we encode this information in the positions of the delayed simulation game. This yields a Büchi game.

For an alternating automaton $\mathbf{Q} = (Q, \Sigma, q_I, \Delta, E, U, F)$ and $k, q \in Q$, let

$$G(k, q) = (P, P_0, P_1, (k, q), Z) \quad (92)$$

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

$$G^{\text{de2}}(k, q) = (P^{\text{de}}, P_0^{\text{de}}, P_1^{\text{de}}, (k, q, b_{kq}), Z^{\text{de}}, W^{\text{de2}}) \quad (93)$$

by

$$P^{\text{de}} = P \times \{0, 1\}, \quad (94)$$

$$P_0^{\text{de}} = P_0 \times \{0, 1\}, \quad (95)$$

$$P_1^{\text{de}} = P_1 \times \{0, 1\}, \quad (96)$$

$$W^{\text{de2}} = (P^{\text{de}*}(P \times \{0\}))^\omega \quad (97)$$

and

$$Z^{\text{de}} = \{((p, b), (p', b)) \in P^{\text{de}} \times P^{\text{de}} \mid (p, p') \in Z, p' \notin Q \times Q\}, \quad (98)$$

$$\cup \{((p, b), (p', b)) \in P^{\text{de}} \times P^{\text{de}} \mid (p, p') \in Z, p' \in (Q \setminus F)(Q \setminus F)\}, \quad (99)$$

$$\cup \{((p, b), (p', 0)) \in P^{\text{de}} \times P^{\text{de}} \mid (p, p') \in Z, p' \in Q \times F\}, \quad (100)$$

$$\cup \{((p, b), (p', 1)) \in P^{\text{de}} \times P^{\text{de}} \mid (p, p') \in Z, p' \in F \times (Q \setminus F)\} \quad (101)$$

with $b_{kq} = 1$ if $k \in F, q \notin F$ and else $b_{kq} = 0$. Observe that the parameters k and q influence the initial position only. The last component of these states will be called the *winning bit*.

Note that the set PF_{de} of pseudo-accepting states (see Section 6.4) 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 $(q, q, 1)$ is a winning position of Duplicator in the above game graph.

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

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

So in the remainder it suffices to consider the games $G^{\text{di}}(k, q)$, $G^{\text{de2}}(k, q)$, and $G^{\text{f}}(k, q)$.

8.2. Proof of Proposition 1, part 1

From [6,26], it follows that the winner of a game $G^{\text{de}2}(q, s)$ always has a positional winning strategy. We can now show that, if Duplicator has a positional winning strategy for $G^{\text{de}2}(q, s)$, then he also has a positional winning strategy for $G^{\text{de}}(q, s)$. Together with Remark 5, this proves the first part of Proposition 1.

The idea of the proof is that Duplicator, using a positional strategy for $G^{\text{de}}(\mathbf{Q}, \mathbf{S})$, plays with a worst-case assumption: He does not “know” the current winning bit, but whenever he can win under the assumption that the winning bit is 1, he does assume that it is 1 indeed.

Let $\mathbf{Q} = (Q, \Sigma, q_I, A^q, E^q, U^q, F^q)$ and $\mathbf{S} = (S, \Sigma, s_I, A^s, E^s, U^s, F^s)$ be alternating Büchi automata such that Duplicator (Player 1) wins

$$G^{\text{de}}(\mathbf{Q}, \mathbf{S}) = (P^{\text{de}}, P_0^{\text{de}}, P_1^{\text{de}}, (q_I, s_I, b_I), Z^{\text{de}}, W^{\text{de}2}), \quad (102)$$

where $b_I = b_{q_I s_I}$.

Let $D \subseteq Q \times S \times \{0, 1\}$ be the set of positions (q, s, b) in $G^{\text{de}2}(\mathbf{Q}, \mathbf{S})$ such that Duplicator has a (positional) winning strategy for a game starting in (q, s, b) ; in particular, $(q_I, s_I, b_I) \in D$ by Remark 5. For every $(q, s, b) \in D$, let σ_{qsb} be a winning strategy for the game starting in (q, s, b) .

We now define a game

$$G^D(\mathbf{Q}, \mathbf{S}) = (P^{\text{de}}, P_0^{\text{de}}, P_1^{\text{de}}, (q_I, s_I, b_D), Z^D, W^{\text{de}2}) \quad (103)$$

such that the set of moves Z^D equals $Z^{\text{de}2}$, but with the following changes. In Z^D , we replace every move $(p, (q, s, 0))$ by a move $(p, (q, s, 1))$ if $s \notin F^s$ and $(q, s, 1) \in D$ (in this case, the winning bit of p is 0). We set $b_D = 1$ if $b_I = 1$, or if $s_I \notin F^s$ and $(q_I, s_I, 1) \in D$, else $b_D = 0$. Note that $(q_I, s_I, b_D) \in D$.

Now Duplicator has a winning strategy for $G^D(\mathbf{Q}, \mathbf{S})$. In a play of $G^D(\mathbf{Q}, \mathbf{S})$, Duplicator starts with the strategy $\sigma_{q_I s_I b_D}$. Whenever a “new” move $(p, (q, s, 1)) \in Z^D \setminus Z^{\text{de}2}$ is taken in this play, Duplicator switches his strategy to $\sigma_{q s 1}$ (remember that $(q, s, 1) \in D$ by definition of Z^D).

This is a winning strategy, because whenever the winning bit switches from 0 to 1, Duplicator effectively plays in $G^{\text{de}2}(\mathbf{Q}, \mathbf{S})$. Upon taking a move such that the winning bit switches from 0 to 1, no “new” transitions can be taken and Duplicator will not switch his strategy again until the winning bit switches back to 0. And this is guaranteed to happen since Duplicator uses a winning strategy.

That is, Duplicator wins $G^D(\mathbf{Q}, \mathbf{S})$, and since $G^D(\mathbf{Q}, \mathbf{S})$ is a Büchi game, there is a positional winning strategy σ for Duplicator. Now it is easy to see that in a σ -conform play of $G^D(\mathbf{Q}, \mathbf{S})$, for every $(q, s) \in Q \times S$, at most one of the positions $(q, s, 0)$ and $(q, s, 1)$ can be encountered. That is, we can assume that σ is defined for at most one value of the winning bit, for every pair of states, and every σ -conform play π can be mapped (by just deleting the winning bit) to a play π' of $G^{\text{de}}(\mathbf{Q}, \mathbf{S})$ such that Duplicator is the winner of π' . In other words, a positional Duplicator winning strategy $\sigma' : P_1 \rightarrow P$ for $G^{\text{de}}(\mathbf{Q}, \mathbf{S})$ can

be defined by

$$\sigma'(p) = \begin{cases} \sigma(p, 0) & \text{if } \sigma(p, 0) \text{ is defined,} \\ \sigma(p, 1) & \text{if } \sigma(p, 1) \text{ is defined,} \\ \text{undefined} & \text{else,} \end{cases} \quad (104)$$

for all $p \in P$. \square

8.3. Reduction of the game graphs

By definition and by Remark 5 it is clear that in order to determine whether $k \leq_{\text{di}} q$, $k \leq_{\text{de}} q$, or $k \leq_{\text{f}} q$ holds it is sufficient to determine the winner in the game $G^{\text{di}}(k, q)$, $G^{\text{de}2}(k, q)$, or $G^{\text{f}}(k, q)$, 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 can safely ignore the winning bit in the considerations of this section.)

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

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

3. A position (k, q, a, A, b, A', b') or (k, q, a, A, b) is a dead end if $\Delta(k, a) = \emptyset$ 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 $|\Delta| \geq |Q| - 1$. Note that the size of the alphabet is not a factor here. So we conclude:

Remark 7. It can be assumed that the game graphs of $G^{\text{di}}(k, q)$, $G^{\text{de}2}(k, q)$, and $G^{\text{f}}(k, q)$ 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 [9]. This yields:

Theorem 6 (Computing simulation relations). *Given an alternating Büchi automaton \mathbf{Q} 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.

8.4. Computing simulation relations of weak alternating Büchi automata

A weak alternating Büchi automaton $\mathbf{Q} = (Q, \Sigma, q_I, \Delta, E, U, F)$ is an alternating Büchi automaton such that every strongly connected component (SCC) $C \subseteq Q$ 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 [22] 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 an SCC of the game graph of $G^x(\mathbf{Q}, \mathbf{Q})$ for $x \in \{\text{di}, \text{de2}, \text{f}\}$, there are SCCs C_0, C_1 of the transition graph of \mathbf{Q} such that $\{\text{pr}_1(p) \mid p \in C\} \subseteq C_0$ and $\{\text{pr}_2(p) \mid p \in C\} \subseteq C_1$.

As a result, if C is an SCC of the game graph of $G^{\text{de2}}(\mathbf{Q}, \mathbf{Q})$, precisely one of the following statements holds:

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

For a game $G^f(\mathbf{Q}, \mathbf{Q})$ the situation is similar but simpler, because there is no winning bit.

That is, for an SCC of the game graph of $G^{\text{de2}}(\mathbf{Q}, \mathbf{Q})$ or $G^f(\mathbf{Q}, \mathbf{Q})$ from which no other SCC is reachable the winning positions can be determined just as in an ordinary game: Duplicator wins the delayed game starting in any position of C if and only if the winning bit is 0. For the fair simulation game, types 4 and 5 collapse to a single type of SCC which is a win for Duplicator, and Duplicator also wins in SCC types 1 and 2. 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 an SCC C , the winning positions of all topologically smaller SCCs have already been computed, i.e., for all positions $p \in C$ such that $(p, p') \in Z$ for a $p' \notin C$, we already know whether p' is a winning position either for Spoiler or for Duplicator. If $p \in P_0$ and p' is a win for Spoiler, p also is a win for Spoiler; else if p' is a win for Duplicator, we may simply ignore the move (p, p') in the computation of the winning positions of C (symmetrically for $p \in P_1$). That is, the treatment of C reduces to a game of accessibility in a boolean graph, and can be carried out in linear time, see [2].

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

1. Compute the SCCs C_0, \dots, C_{n-1} of the game graph (the time expense is linear in the number of positions and moves [29], that is, the SCCs can be computed in time $O(nm)$).
2. Compute a topological sorting $C_{i_0} \leq_T C_{i_1} \leq_T \dots \leq_T C_{i_{n-1}}$ of the SCCs of the game graph (linear in the number of positions and moves [21]).
3. Compute in the order $C_{i_{n-1}}, C_{i_{n-2}}, \dots, C_{i_0}$ 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 [2].

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.

9. 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. In part, these have been investigated in [10].

References

- [1] R. Alur, Th.A. Henzinger, O. Kupferman, M.Y. Vardi, Alternating refinement relations, in: D. Sangiorgi, R. de Simone (Eds.), CONCUR '98: Concurrency Theory, 9th Internat. Conf., Nice, France, Lecture Notes in Computer Science, Vol. 1466, Springer, Berlin, 1998, pp. 163–178.
- [2] H.R. Andersen, Model checking and boolean graphs, Theoret. Comput. Sci. 126 (1) (1994) 3–30.
- [3] D. Bustan, O. Grumberg, Simulation based minimization, in: D.A. McAllester (Ed.), Automated Deduction—CADE-17, 17th Internat. Conf. on Automated Deduction, Pittsburg, PA, USA, Lecture Notes in Computer Science, Vol. 1831, Springer, Berlin, 2000, pp. 255–270.
- [4] M. Daniele, F. Giunchiglia, M.Y. Vardi, Improved automata generation for linear time temporal logic, in: N. Halbwachs, D. Peled (Eds.), Computer Aided Verification, 11th Internat. Conf., (CAV '99), Trento, Italy, Lecture Notes in Computer Science, Vol. 1633, Springer, Berlin, 1999, pp. 249–260.
- [5] D.L. Dill, A.J. Hu, H. Wong-Toi, Checking for language inclusion using simulation preorders, in: K. Guldstrand Larsen, A. Skou (Eds.), Computer Aided Verification, 3rd Internat. Workshop, CAV '91, Aalborg, Denmark, Lecture Notes in Computer Science, Vol. 575, Springer, Berlin, 1991, pp. 255–265.
- [6] E.A. Emerson, C.S. Jutla, Tree automata, mu-calculus and determinacy (extended abstract), in: Proc. 32nd Ann. Symp. on Foundations of Computer Science (FoCS '91), IEEE Computer Society Press, Silver Spring, MD, 1991, pp. 368–377.
- [7] K. Etessami, A hierarchy of polynomial-time computable simulations for automata, in: L. Brim, P. Janar, M. Ketínský, A. Kuera (Eds.), CONCUR 2002—Concurrency Theory: 13th Internat. Conf., Brno, Czech Republic, Lecture Notes in Computer Science, Vol. 2421, Springer, Berlin, 2002, pp. 131–144.
- [8] K. Etessami, G. Holzmann, Optimizing Büchi automata, in: C. Palamidessi (Ed.), CONCUR 2000 — Concurrency Theory: 11th Internat. Conf., University Park, PA, USA, Lecture Notes in Computer Science, Vol. 1877, Springer, Berlin, 2000, pp. 153–167.
- [9] K. Etessami, R. Schuller, Th. Wilke, Fair simulation relations, parity games, and state space reduction for Büchi automata, in: F. Orejas, P.G. Spirakis, J. van Leeuwen (Eds.), Automata, Languages and Programming, 28th Internat. Coll., ICALP 2001, Crete, Greece, Lecture Notes in Computer Science, Vol. 2076, Springer, Berlin, 2001, pp. 694–707.
- [10] C. Fritz, Constructing Büchi automata from linear temporal logic using simulation relations for alternating Büchi automata, in: O.H. Ibarra, Z. Dang (Eds.), Implementation and Application of Automata, Eighth Internat. Conf. CIAA 2003, Santa Barbara, CA, USA, Lecture Notes in Computer Science, Vol. 2759, Springer Berlin, 2003, pp. 35–48.
- [11] C. Fritz, B. Teegen, LTL \rightarrow NBA, Link at <http://www.ti.informatik.uni-kiel.de/~fritz/>.

- [12] C. Fritz, Th. Wilke, State space reductions for alternating Büchi automata: quotienting by simulation equivalences, in: M. Agrawal, A. Seth (Eds.), 22nd Conf. on Foundations of Software Technology and Theoretical Computer Science, Kanpur, India, Lecture Notes in Computer Science, Vol. 2556, Springer, Berlin, 2002, pp. 157–168.
- [13] P. Gastin, D. Oddoux, Fast LTL to Büchi automata translation, in: G. Berry, H. Comon, A. Finkel (Eds.), Computer Aided Verification, 13th Internat. Conf., CAV 2001, Paris, France, 2001, Lecture Notes in Computer Science, Vol. 2102, Springer, Berlin, 2001, pp. 53–65.
- [14] R. Gerth, D. Peled, M.Y. Vardi, P. Wolper, Simple on-the-fly automatic verification of linear temporal logic, in: Proc. 15th Workshop on Protocol Specification, Testing, and Verification, Warsaw, Poland, June 1995, Chapman & Hall, London, 1995, pp. 3–18.
- [15] Y. Gurevich, L. Harrington, Trees, automata, and games, in: 14th ACM Symp. on the Theory of Computing, San Francisco, 1982, ACM Press, New York, pp. 60–65.
- [16] S. Gurumurthy, R. Bloem, F. Somenzi, Fair simulation minimization, in: E. Brinksma, K. Guldstrand Larsen (Eds.), Computer Aided Verification, 14th Internat. Conf., CAV 2002, Copenhagen, Denmark, Lecture Notes in Computer Science, Vol. 2404, Springer, Berlin, 2002, pp. 610–623.
- [17] Th.A. Henzinger, O. Kupferman, S.K. Rajamani, Fair simulation, in: CONCUR '97: Concurrency Theory, 8th Internat. Conf., Warsaw, Poland, Lecture Notes in Computer Science, Vol. 1243, Springer, Berlin, 1997, pp. 273–287.
- [18] Th.A. Henzinger, S.K. Rajamani, Fair bisimulation, in: S. Graf, M. Schwartzbach (Eds.), Tools and Algorithms for Construction and Analysis of Systems, 6th Internat. Conf., TACAS 2000, Lecture Notes in Computer Science, Vol. 1785, Springer, Berlin, 2000, pp. 299–314.
- [19] M. Henzinger Rauch, Th.A. Henzinger, P.W. Kopke, Computing simulations on finite and infinite graphs, in: 36th Ann. Symp. on Foundations of Computer Science (FOCS '95), 1995, pp. 453–462.
- [20] M. Jurdziński, Small progress measures for solving parity games, in: H. Reichel, S. Tison (Eds.), STACS 2000, 17th Ann. Symp. on Theoretical Aspects of Computer Science, Lille, France, Lecture Notes in Computer Science, Vol. 1770, February 2000, Springer, Berlin, 2000, pp. 290–301.
- [21] D.E. Knuth, Fundamental algorithms, The Art of Computer Programming, Vol. 1, Addison-Wesley, Reading, MA, 1968 (second ed., 1973).
- [22] O. Kupferman, M.Y. Vardi, P. Wolper, An automata-theoretic approach to branching-time model checking, J. ACM 47 (2) (2000) 312–360.
- [23] R. Milner, An algebraic definition of simulation between programs, in: D.C. Cooper (Ed.), Proc. 2nd Internat. Joint Conf. on Artificial Intelligence, pp. 481–489, London, UK, September 1971, William Kaufmann.
- [24] R. Milner, Communication and Concurrency, Prentice-Hall, Englewood Cliffs, NJ, 1989.
- [25] S. Miyano, T. Hayashi, Alternating finite automata on ω -words, Theoret. Comput. Sci. 32 (1984) 321–330.
- [26] A.W. Mostowski, Games with forbidden positions, Tech. Report 78, Instytut Matematyki, Uniwersytet Gdański, Poland, 1991.
- [27] D.E. Muller, A. Saoudi, P.E. Schupp, Weak alternating automata give a simple explanation of why most temporal and dynamic logics are decidable in exponential time, in: 3rd IEEE Ann. Symp. on Logic in Computer Science, Edinburgh, Scotland, UK, 1988, pp. 422–427.
- [28] F. Somenzi, R. Bloem, Efficient Büchi automata from LTL formulae, in: E.A. Emerson, A. Prasad Sistla (Eds.), Computer Aided Verification, 12th Internat. Conf., CAV 2000, Chicago, IL, USA, Lecture Notes in Computer Science, Vol. 1855, Springer, Berlin, July 2000, pp. 248–263.
- [29] R.E. Tarjan, Depth first search and linear graphs, SIAM J. Comput. 1 (2) (1972) 146–160.
- [30] M.Y. Vardi, Nontraditional applications of automata theory, in: M. Hagiya, J.C. Mitchell (Eds.), Theoretical Aspects of Computer Software, Internat. Conf. TACS '94, Sendai, Japan, Lecture Notes in Computer Science, Vol. 789, Springer, Berlin, 1994, pp. 575–597.