Profile Trees for Büchi Word Automata, with Application to Determinization

Seth Fogarty
Computer Science Department
Trinity University

Orna Kupferman
School of Computer Science and Engineering
Hebrew University of Jerusalem

Moshe Y. Vardi
Department of Computer Science
Rice University

Thomas Wilke
Institut für Informatik
Christian-Albrechts-Universität zu Kiel

The determinization of Büchi automata is a celebrated problem, with applications in synthesis, probabilistic verification, and multi-agent systems. Since the 1960s, there has been a steady progress of constructions: by McNaughton, Safra, Piterman, Schewe, and others. Despite the proliferation of constructions, they are all essentially ad-hoc constructions, with little theory behind them other than proofs of correctness. Since Safra, all optimal constructions employ trees as states of the deterministic automaton, and transitions between states are defined operationally over these trees. The operational nature of these constructions complicates understanding, implementing, and reasoning about them, and should be contrasted with complementation, where a solid theory in terms of automata run DAGs underlies modern constructions.

In 2010, we described a *profile*-based approach to Büchi complementation, where a profile is simply the history of visits to accepting states. We developed a structural theory of profiles and used it to describe a complementation construction that is deterministic in the limit. Here we extend the theory of profiles to prove that every run DAG contains a *profile tree* with at most a finite number of infinite branches. We then show that this property provides a theoretical grounding for a new determinization construction where macrostates are doubly preordered sets of states. In contrast to extant determinization constructions, transitions in the new construction are described declaratively rather than operationally.

1 Introduction

Büchi automata were introduced in the context of decision problems for second-order arithmetic [3]. These automata constitute a natural generalization of automata over finite words to languages of infinite words. Whereas a run of an automaton on finite words is accepting if the run ends in an accepting state, a run of a Büchi automaton is accepting if it visits an accepting state infinitely often.

Determinization of nondeterministic automata is a fundamental problem in automata theory, going back to [19]. Determinization of Büchi automata is employed in many applications, including synthesis of reactive systems [18], verification of probabilistic systems [4, 26], and reasoning about multi-agent systems [2]. Nondeterministic automata over finite words can be determinized with a simple, although exponential, *subset construction* [19], where a state in the determinized automaton is a set of states of the input automaton. Nondeterministic Büchi automata, on the other hand, are not closed under determinization, as deterministic Büchi automata are strictly less expressive than their nondeterministic counterparts [13]. Thus, a determinization construction for Büchi automata must result in automata with a more powerful acceptance condition, such as Muller [15], Rabin [20], or parity conditions [9, 17].

The first determinization construction for Büchi automata was presented by McNaughton, with a doubly-exponential blowup [15]. In 1988, Safra introduced a singly exponential construction [20],

Submitted to:
GANDALF 2013

matching the lower bound of $n^{O(n)}$ [14]. Safra's construction encodes a state of the determinized automaton as a labeled tree, now called a *Safra tree*, of sets of states of the input Büchi automaton. Subsequently, Safra's construction was improved by Piterman, who simplified the use of tree-node labels [17], and by Schewe, who moved the acceptance conditions from states to edges [22]. In a separate line of work, Muller and Schupp proposed in 1995 a different singly exponential determinization construction, based on *Muller-Schupp trees* [16], which was subsequently simplified by Kähler and Wilke [9].

Despite the proliferation of Büchi determinization constructions, even in their improved and simplified forms all constructions are essentially ad-hoc, with little theory behind them other than correctness proofs. These constructions rely on the encoding of determinized-automaton states as finite trees. They are operational in nature, with transitions between determinized-automaton states defined "horticulturally," as a sequence of operations that grow trees and then prune them in various ways. The operational nature of these constructions complicates understanding, implementing, and reasoning about them [1, 23], and should be contrasted with complementation, where an elegant theory in terms of automata run DAGs underlies modern constructions [8, 11, 21]. In fact, the difficulty of determinization has motivated attempts to find determinization-free decision procedures [12] and works on determinization of fragments of LTL [10].

In a recent work [6], we introduced the notion of *profiles* for nodes in the run DAG. We began by labeling accepting nodes of the DAG by 1 and non-accepting nodes by 0, essentially recording visits to accepting states. The profile of a node is the lexicographically *maximal* sequence of labels along paths of the run DAG that lead to that node. Once profiles of nodes and a lexicographic order over profiles were defined, we removed from the run DAG edges that do not contribute to profiles. In the pruned run DAG, we focused on lexicographically maximal runs. This enabled us to define a novel, profile-based Büchi complementation construction that yields *deterministic-in-the-limit* automata: one in which every accepting run of the complementing automaton is eventually deterministic [6] A state in the complementary automaton is a set of states of the input nondeterministic automaton, augmented with the preorder induced by profiles. Thus, this construction can be viewed as an augmented subset construction.

In this paper, we develop the theory of profiles further, and consider the equivalence classes of nodes induced by profiles, in which two nodes are in the same class if they have the same profile. We show that profiles turn the run DAG into a *profile tree*: a binary tree of *bounded width* over the equivalence classes. The profile tree affords us a novel singly exponential Büchi determinization construction. In this profile-based determinization construction, a state of the determinized automaton is a set of states of the input automaton, augmented with *two* preorders induced by profiles. Note that while a Safra tree is finite and encodes a single level of the run DAG, our profile tree is infinite and encodes the entire run DAG, capturing the accepting or rejecting nature of all paths. Thus, while a state in a traditional determinization construction corresponds to a Safra tree, a state in our deterministic automaton corresponds to a single level in the profile tree.

Unlike previous Büchi determinization constructions, transitions between states of the determinized automaton are defined declaratively rather than operationally. We believe that the declarative character of the new construction will open new lines of research on Büchi determinization. For Büchi complementation, the theory of run DAGs [11] led not only to tighter constructions [8, 21], but also to a rich body of work on heuristics and optimizations [5, 7]. We foresee analogous developments in research on Büchi determinization.

2 Preliminaries

This section introduces the notations and definitions employed in our analysis.

2.1 Relations on Sets

Given a set R, a binary relation $\leq$ over R is a *preorder* if $\leq$ is reflexive and transitive. If for every $r_1, r_2 \in R$ either $r_1 \leq r_2$ or $r_2 \leq r_1$, then $\leq$ is a *linear preorder*. If a preorder $\leq$ is antisymmetric, that is if $r_1 \leq r_2$ and $r_2 \leq r_1$ implies $r_1 = r_2$, then it is a *partial order*. A linear partial order is a *total order*. Consider a partial order $\leq$. If for every $r \in R$, the set $\{r' \mid r' \leq r\}$ of smaller elements is totally ordered by $\leq$, then we say that $\leq$ is a *tree order*. The equivalence class of $r \in R$ under $\leq$, written $[r]$, is $\{r' \mid r' \leq r \text{ and } r \leq r'\}$. The equivalence classes under a linear preorder form a totally ordered partition of R. Given a set R and linear preorder $\leq$ over R, define the minimal elements of R as $\mathtt{min}_{\leq}(R) = \{r_1 \in R \mid \ r_1 \leq r_2 \text{ for all } r_2 \in R\}$. Note that $\mathtt{min}_{\leq}(R)$ is either empty or an equivalence class under $\leq$. Given a non-empty set R and a total order $\leq$, we instead define $\mathtt{min}_{\leq}$ as the function that maps R to its unique minimal element.

Given two finite sets R and R' where $|R| \leq |R'|$, a linear preorder $\leq$ over R, and a total order $<'$ over R', define the $\langle \leq, <' \rangle$*-minjection* from R to R' to be the function $\mathtt{mj}$ that maps all the elements in the k-th equivalence class of R to the k-th element of R'. The number of equivalence classes is at most $|R|$, and thus at most $|R'|$. If $\leq$ is also a total order, than the $\langle \leq, <' \rangle$-minjection is also an injection.

Example 2.1. Let $R = \mathbb{Q}$ and $R' = \mathbb{N}$ be the sets of rational numbers and integers, respectively. Define the linear preorder $\leq_1$ over $\mathbb{Q}$ by $x \leq_1 x'$ iff $\lfloor x \rfloor \leq \lfloor x' \rfloor$, and the total order $<_2$ over $\mathbb{N}$ by $x <_2 x'$ if $x < x'$. Then, the $\langle \leq_1, <_2 \rangle$-minjection from $\mathbb{Q}$ to $\mathbb{N}$ maps a rational number x to $\lfloor x \rfloor$.

2.2 ω-Automata

A *nondeterministic* ω*-automaton* is a tuple $\mathcal{A} = \langle \Sigma, Q, Q^{in}, \rho, \alpha \rangle$, where Σ is a finite alphabet, Q is a finite set of states, $Q^{in} \subseteq Q$ is a set of initial states, $\rho \colon Q \times \Sigma \to 2^Q$ is a nondeterministic transition relation, and α is an acceptance condition defined below. An automaton is *deterministic* if $|Q^{in}| = 1$ and, for every $q \in Q$ and $\sigma \in \Sigma$, we have $|\rho(q, \sigma)| = 1$. For a function $\delta \colon Q \times \Sigma \to 2^Q$, we lift δ to sets R of states in the usual fashion: $\delta(R, \sigma) = \bigcup_{r \in R} \delta(r, \sigma)$. Further, we define the inverse of δ, written δ^{-1}, to be $\delta^{-1}(r, \sigma) = \{q \mid r \in \delta(q, \sigma)\}$.

A *run* of an ω-automaton $\mathcal{A}$ on a word $w = \sigma_0 \sigma_1 \cdots \in \Sigma^\omega$ is an infinite sequence of states $q_0, q_1, \ldots \in Q^\omega$ such that $q_0 \in Q^{in}$ and, for every $i \geq 0$, we have that $q_{i+1} \in \rho(q_i, \sigma_i)$. Correspondingly, a *finite run* of $\mathcal{A}$ to q on $w = \sigma_0 \cdots \sigma_{n-1} \in \Sigma^*$ is a finite sequence of states $p_0, \ldots, p_n$ such that $p_0 \in Q^{in}$, $p_n = q$, and for every $0 \leq i < n$ we have $p_{i+1} \in \rho(p_i, \sigma_i)$.

The acceptance condition α determines if a run is *accepting*. If a run is not accepting, we say it is *rejecting*. A word $w \in \Sigma^\omega$ is accepted by $\mathcal{A}$ if there exists an accepting run of $\mathcal{A}$ on w. The words accepted by $\mathcal{A}$ form the *language* of $\mathcal{A}$, denoted by $L(\mathcal{A})$. For a *Büchi automaton*, the acceptance condition is a set of states $F \subseteq Q$, and a run $q_0, q_1, \ldots$ is accepting iff $q_i \in F$ for infinitely many i's. For convenience, we assume $Q^{in} \cap F = \emptyset$. For a *Rabin automaton*, the acceptance condition is a sequence $\langle G_0, B_0 \rangle, \ldots, \langle G_k, B_k \rangle$ of pairs of sets of states. Intuitively, the sets G are "good" conditions, and the sets B are "bad" conditions. A run $q_o, q_1, \ldots$ is accepting iff there exists $0 \leq j \leq k$ so that $q_i \in G_j$ for infinitely many i's, while $q_i \in B_j$ for only finitely many i's. Our focus in this paper is on nondeterministic Büchi automata on words (NBW) and deterministic Rabin automata on words (DRW).

2.3 Safra's Determinization Construction

This section presents Safra's determinization construction, using the exposition in [17]. Safra's construction takes an NBW and constructs an equivalent DRW. Intuitively, a state in this construction is a tree of subsets. Every node in the tree is labeled by the states it follows. The label of a node is a strict superset of the union of labels of its descendants, and the labels of siblings are disjoint. Children of a node are ordered by "age". Let $\mathcal{A} = \langle \Sigma, Q, Q^{in}, \rho, F \rangle$ be an NBW, $n = |Q|$, and $V = \{0, \ldots, n-1\}$.

Definition 2.2. [17] A *Safra tree* over $\mathcal{A}$ is a tuple $t = \langle N, r, p, \psi, l, G, B \rangle$ where:

- $N \subseteq V$ is a set of nodes.
- $r \in N$ is the root node.
- $p : (N \setminus r) \to N$ is the parent function over $N \setminus \{r\}$.
- ψ is a partial order defining 'older than' over siblings.
- $l : N \to 2^Q$ is a labeling function from nodes to non-empty sets of states. The label of every node is a proper superset of the union of the labels of its sons. The labels of two siblings are disjoint.
- $G, B \subseteq V$ are two disjoint subsets of V.

The only way to move from one Safra tree to the next is through a sequence of "horticultural" operations, growing the tree and then pruning it to ensure that the above invariants hold.

Definition 2.3. Define the DRW $D^S(\mathcal{A}) = \langle \Sigma, Q_S, \rho_S, t_0, \alpha \rangle$ where:

- Q_S is the set of Safra trees over $\mathcal{A}$.
- $t_0 = \langle \{0\}, 0, \emptyset, \emptyset, l_0, \emptyset, \{1, \ldots, n-1\} \rangle$ where $l_0(0) = Q^{in}$
- For $t = \langle N, r, p, \psi, l, G, B \rangle \in Q_S$ and $\sigma \in \Sigma$, the tree $t' = \rho_S(t, \sigma)$ is the result of the following sequence of operations. We temporarily use a set V' of names disjoint from V. Initially, let $t' = \langle N', r', p', \psi', l', G', B' \rangle$ where $N' = N$, $r' = r$, $p' = p$, $\psi' = \psi$, l' is undefined, and $G' = B' = \emptyset$.
  (1) For every $v \in N'$, let $l'(v) = \rho(l(v), \sigma)$.
  (2) For every $v \in N'$ such that $l'(v) \cap F \neq \emptyset$, create a new node $v' \in V'$ where: $p(v') = v$; $l'(v') = l'(v) \cap F$; and for every $w' \in V'$ where $p(w') = v$ add (w', v') to ψ.
  (3) For every $v \in N'$ and $q \in l'(v)$, if there is a $w \in N'$ such that $(w, v) \in \psi$ and $q \in l'(w)$, then remove q from $l'(v)$ and, for every descendant v' of v, remove q from $l'(v')$.
  (4) Remove all nodes with empty labels.
  (5) For every $v \in N'$, if $l'(v) = \bigcup \{l'(v') \mid p'(v') = v\}$ remove all children of v, add v to G.
  (6) Add all unused nodes from V to B.
  (7) Change the nodes in V' to unused nodes in V.
- $\alpha = \{\langle G_0, B_0 \rangle, \ldots, \langle G_{n-1}, B_{n-1} \rangle\}$, where:
  - $G_i = \{\langle N, r, p, \psi, l, G, B \rangle \in Q_S \mid i \in G\}$
  - $B_i = \{\langle N, r, p, \psi, l, G, B \rangle \in Q_S \mid i \in B\}$

Theorem 2.4. [20] *For an NBW $\mathcal{A}$ with n states, $L(D^S(\mathcal{A})) = L(\mathcal{A})$ and $D^S(\mathcal{A})$ has $n^{O(n)}$ states.*

3 From Run DAGs to Profile Trees

In this section, we present a framework for simultaneously reasoning about all runs of a Büchi automaton on a word. We use a DAG to encode all possible runs, and give each node in this DAG a profile based on its history. The lexicographic order over profiles induces a preorder $\preceq_i$ over the nodes on level i of the run DAG. Using $\preceq_i$, we prune the edges of the run DAG, and derive a binary tree of bounded width. Throughout this paper we fix an NBW $\mathcal{A} = \langle \Sigma, Q, Q^{in}, \rho, F \rangle$ and an infinite word $w = \sigma_0 \sigma_1 \cdots$.

3.1 Run DAGs and Profiles

The runs of $\mathcal{A}$ on w can be arranged in an infinite DAG $G = \langle V, E \rangle$, where

- $V \subseteq Q \times \mathbb{N}$ is such that $\langle q, i \rangle \in V$ iff there is a finite run of $\mathcal{A}$ to q on $\sigma_0 \cdots \sigma_{i-1}$.
- $E \subseteq \bigcup_{i \geq 0} (Q \times \{i\}) \times (Q \times \{i+1\})$ is such that $E(\langle q, i \rangle, \langle q', i+1 \rangle)$ iff $\langle q, i \rangle \in V$ and $q' \in \rho(q, \sigma_i)$.

The DAG G, called the *run* DAG *of* $\mathcal{A}$ *on* w, embodies all possible runs of $\mathcal{A}$ on w. We are primarily concerned with *initial paths* in G: paths that start in $Q^{in} \times \{0\}$. A node $\langle q, i \rangle$ is an F-node if $q \in F$, and a path in G is *accepting* if it is both initial and contains infinitely many F-nodes. An accepting path in G corresponds to an accepting run of $\mathcal{A}$ on w. If G contains an accepting path, we say that G is *accepting*; otherwise it is *rejecting*. Let G' be a sub-DAG of G. For $i \geq 0$, we refer to the nodes in $Q \times \{i\}$ as *level* i of G'. Note that a node on level $i+1$ has edges only from nodes on level i. We say that G' has *bounded width of degree* c if every level in G' has at most c nodes. By construction, G has bounded width of degree $|Q|$.

Consider the run DAG $G = \langle V, E \rangle$ of $\mathcal{A}$ on w. Let $f : V \to \{0, 1\}$ be such that $f(\langle q, i \rangle) = 1$ if $q \in F$ and $f(\langle q, i \rangle) = 0$ otherwise. Thus, f labels F-nodes by 1 and all other nodes by 0. The *profile* of a path in G is the sequence of labels of nodes in the path. The profile of a node is then the lexicographically maximal profile of all initial paths to that node. The profile of a finite path $b = v_0, v_1, \ldots, v_n$ in G, written h_b, is $f(v_0) f(v_1) \cdots f(v_n)$, and the profile of an infinite path $b = v_0, v_1, \ldots$ is $h_b = f(v_0) f(v_1) \cdots$. Finally, the profile of a node v, written h_v, is the lexicographically maximal element of $\{h_b \mid b \text{ is an initial path to } v\}$.

The lexicographic order of profiles induces a linear preorder over nodes. We define a sequence of linear preorders $\preceq_i$ over the nodes on level i of G as follows. For nodes u and v on level i, let $u \prec_i v$ if $h_u < h_v$, and $u \approx_i v$ if $h_u = h_v$. We group nodes by their equivalence classes under $\preceq_i$. Since the final element of a node's profile is 1 iff the node is an F-node, all nodes in an equivalence class agree on membership in F. Call an equivalence class an F-class when all members are F-nodes, and a non-F-class when none of its members are F-nodes. When a state can be reached by two finite runs, a node will have multiple incoming edges in G. We now remove from G all edges that do not contribute to profiles. Formally, define the pruned run DAG $G' = \langle V, E' \rangle$ where $E' = \{\langle u, v \rangle \in E \mid \text{for every } u' \in V, \text{ if } \langle u', v \rangle \in E \text{ then } u' \preceq_{|u|} u\}$. Note that the set of nodes in G and G' are the same, and that an edge is removed from E' only when there is another edge to its destination.

Lemma 3.1 states that, as we have removed only edges that do not contribute to profiles, nodes derive their profiles from their parents in G'.

Lemma 3.1. [6] *For two nodes u and u' in V, if $\langle u, u' \rangle \in E'$, then $h_{u'} = h_u 0$ or $h_{u'} = h_u 1$.*

While nodes with different profiles can share a child in G, Lemma 3.2 precludes this in G'.

Lemma 3.2. *Consider nodes u and v on level i of G' and nodes u' and v' on level $i+1$ of G'. If $\langle u, u' \rangle \in E'$, $\langle v, v' \rangle \in E'$, and $u' \approx_{i+1} v'$, then $u \approx_i v$.*

Proof: Since $u' \approx_{i+1} v'$, we have $h_{u'} = h_{v'}$. If u' is an F-node, then v' is an F-node and the last letter in both $h_{u'}$ and $h_{v'}$ is 1. By Lemma 3.1 we have $h_u 1 = h_{u'} = h_{v'} = h_v 1$. If u' and v' are non-F-nodes, then we have $h_u 0 = h_{u'} = h_{v'} = h_v 0$. In either case, $h_u = h_v$ and $u \approx_i v$. □

Finally, we have that G' captures the accepting or rejecting nature of G. This result was employed to provide deterministic-in-the-limit complementation in [6]

Theorem 3.3. [6] *The pruned run* DAG *G' of an NBW $\mathcal{A}$ on a word w is accepting iff $\mathcal{A}$ accepts w.*

3.2 The Profile Tree

Using profiles, we define the *profile tree* T, a binary tree of bounded width that captures the accepting or rejecting nature of the pruned run DAG G'. The nodes of T are the equivalence classes $\{[u] \mid u \in G'\}$ of G'. To remove confusion, we refer to the nodes of T as *classes* and use and U and V for classes in T, while reserving u and v for nodes in G or G'. The edges in T are induced by these in G' as expected: for an edge $\langle u,v\rangle \in E'$, the class $[v]$ is the child of $[u]$ in T. A class V is a *descendant* of a class U if there is a, possibly empty, path from U to V.

Theorem 3.4. *The profile tree T of an n-state NBW $\mathcal{A}$ on an infinite word w is a binary tree whose width is bounded by n.*

Proof: That T has bounded width follows from the fact that a class on level i contains at least one node on level i of G, and G is of bounded width of degree n. To prove every class has one parent, for a class V let $U = \{u \mid \text{there is } v \in V \text{ such that } \langle u,v\rangle \in E'\}$. Lemma 3.2 implies that U is an equivalence class, and is the sole parent of V. To show that T has a root, note that as $Q^{in} \cap F = \emptyset$, all nodes on the first level of G have profile 0, and every class descends from this class of nodes with profile 0. Finally, as noted Lemma 3.1 entails that a class U can have at most two children: the class with profile $h_U 1$, and the class with profile $h_U 0$. Thus T is binary. □

A *branch* of T is a finite or infinite initial path in T. Since T is a tree, two branches share a prefix until they *split*. An infinite branch is *accepting* if it contains infinitely many F-classes, and *rejecting* otherwise. An infinite rejecting branch must reach a suffix consisting only of non-F-classes. A class U is called *finite* if it has finitely many descendants, and a finite class U *dies out* on level k if it has a descendant on level $k-1$, but none on level k. Say T is *accepting* if it contains an accepting branch, and *rejecting* if all branches are rejecting.

As all members of a class share a profile, we define the profile h_U of a class U to be h_u for some node $u \in U$. We extend the function f to classes, so that $f(U) = 1$ if U is an F-class, and $f(U) = 0$ otherwise. We can then define the profile of an infinite branch $b = U_0, U_1, \ldots$ to be $h_b = f(U_0)f(U_1)\cdots$. For two classes U and V on level i, we say that $U \prec_i V$ if $h_U < h_V$. For two infinite branches b and b', we say that $b \prec b'$ if $h_b < h_{b'}$. Note that $\prec_i$ is a total order over the classes on level i, and that $\prec$ is a total order over the set of infinite branches.

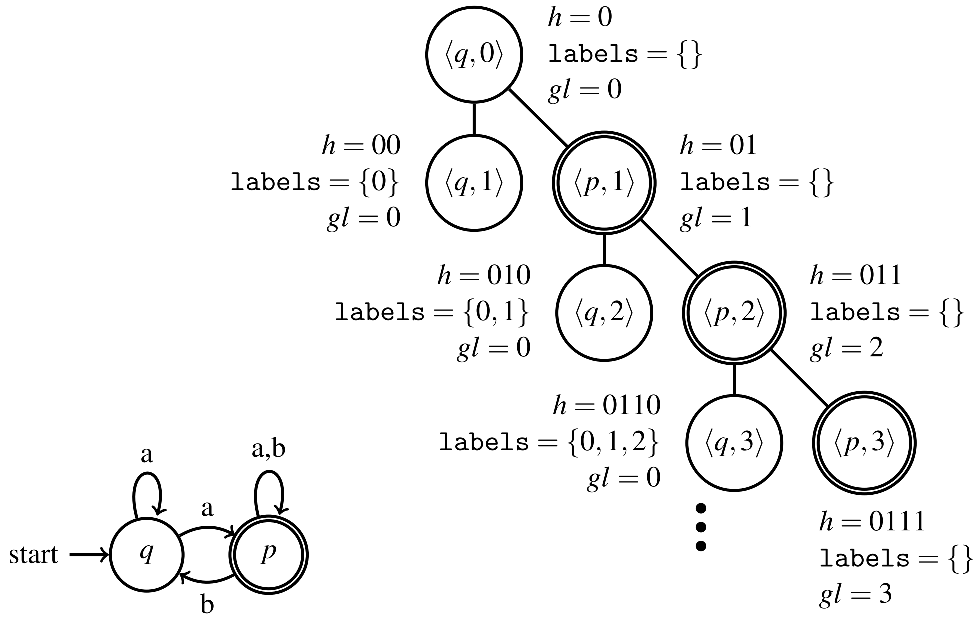
As proven above, a class U has at most two children: the class of F-nodes with profile $h_U 1$, and the class of non-F-nodes with profile $h_U 0$. We call the first class the F-child of U, and the second class the non-F-child of U. While G' can have infinitely many infinite branches, bounding the width of the profile tree also bounds the number of infinite branches it may have.

Corollary 3.5. *The profile tree T of an NBW $\mathcal{A}$ on an infinite word w has a finite number of infinite branches.*

Example 3.6. Consider for example the NBW in Figure 1(a), and the tree of equivalence classes that corresponds to a run of it in the word ab^ω in Figure 1(b). The only infinite branch, $\{\langle q,0\rangle\},\{\langle p,1\rangle\},\{\langle p,2\rangle\},\ldots$, is accepting. The set of labels and the global labeling gl are explained below, in Section 4.1.

We conclude this section with Theorem 3.7, which enables us to reduce the search for an accepting path in G' to a search for an accepting branch in T.

Theorem 3.7. *The profile tree T of an NBW $\mathcal{A}$ on an infinite word w is accepting iff $\mathcal{A}$ accepts w.*



(a) An automaton **(b)** T for automaton (a) on ab^ω.

Figure 1: An automaton and tree of classes. Each class is a singleton set, brackets are omitted for brevity. F-classes are circled twice. Each class is labeled with its profile h, as well as the set `labels` and the global label gl as defined in Section 4.1.

Proof: If $w \in L(\mathcal{A})$, then by Theorem 3.3 we have that G' contains an accepting path $u_0, u_1, \ldots$. This path gives rise to an accepting branch $[u_0], [u_1], \ldots$ in T. In the other direction, if T has an accepting branch $U_0, U_1, \ldots$, consider the infinite subgraph of G' consisting only of the nodes in U_i, for $i > 0$. For every $i > 0$ there exists $u_i \in U_i$ and $u_{i+1} \in U_{i+1}$ so that $E'(u_i, u_{i+1})$. Because no node is orphaned in G', Lemma 3.2 implies that every node in U_{i+1} has a parent in U_i, thus this subgraph is connected. As each node has degree of as most n, König's Lemma implies that there is an initial path $u_0, u_1, \ldots$ through this subgraph. Further, at every level i where U_i is an F-class, we have that $u_i \in F$, and thus this path is accepting and $w \in L(\mathcal{A})$. □

4 Labeling

In this section we present a method of deterministically labeling the classes in T with integers, so we can determine if T is accepting by examining the labels. Each label m represents the proposition that the lexicographically minimal infinite branch through the first class labeled with m is accepting. On each level we give the label m to the lexicographically minimal descendant, on any branch, of this first class labeled with m. We initially allow the use of global information about T and an unbounded number of labels. We then show how to determine the labeling using bounded information about each level of T, and how to use a fixed set of labels.

4.1 Labeling T

We first present a labeling that uses an unbounded number of labels and global information about T. We call this labeling the *global labeling*, and denote it with gl. For a class U on level i of T, and a class V on level j, we say that V is *before* U if $j < i$ or $j = i$ and $V \prec_i U$. For each label m, we refer to the first class labeled m as $\texttt{first}(m)$. That is, $U = \texttt{first}(m)$ if U is labeled m and, for all classes V before U, the label of V is not m. We define the labeling function gl inductively over the nodes of T. For the initial class $U_0 = \{\langle q,0\rangle \mid q \in Q^{in}\}$ with profile 0, let $gl(U_0) = 0$.

Each label m follows the lexicographically minimal child of $\texttt{first}(m)$ on every level. When a class with label m has two children, we are not certain which, if either, is part of an infinite branch. We are thus conservative, and follow the non-F-child. If the non-F-child dies out, we revise our guess and move to a descendant of the F-child. For a label m and level i, let the lexicographically minimal descendant of m on level i, written $\texttt{lmd}(m,i)$, be $\texttt{min}_{\preceq}(\{V \mid V \text{ is a descendant of } \texttt{first}(m) \text{ on level } i\})$. That is, $\texttt{lmd}(m,i)$ is the class with the minimal profile among all the descendants of $\texttt{first}(m)$ on level i. For a class U on level i, define $\texttt{labels}(U) = \{m \mid U = \texttt{lmd}(m,i)\}$ as the set of valid labels for U. If U has more than one valid label, we give it the smallest label, which corresponds to the earliest ancestor. If $\texttt{labels}(U)$ is empty, U is given an unused label one greater than the maximum label occurring earlier in T.

Definition 4.1. $gl(U) = \begin{cases} \texttt{min}(\texttt{labels}(U)) & \text{if } \texttt{labels}(U) \neq \emptyset, \\ \texttt{max}(\{gl(V) \mid V \text{ is before } U\}) + 1 & \text{if } \texttt{labels}(U) = \emptyset. \end{cases}$

Lemma 4.2 demonstrates that every class on a level gets a unique label, and that despite moving between nephews the labeling adheres to branches in the tree. The proof is reserved for Appendix A.

Lemma 4.2. *For classes U and V on level i of T, it holds that:*
(1) If $U \neq V$ then $gl(U) \neq gl(V)$.
(2) U is a descendant of $\texttt{first}(gl(U))$.
(3) If U is a descendant of $\texttt{first}(gl(V))$, then $V \preceq_i U$. Consequently, if $U \prec_i V$, then U is not a descendant of $\texttt{first}(gl(V))$.
(4) $\texttt{first}(gl(U))$ is the root or an F-class with a sibling.
(5) If $U \neq \texttt{first}(gl(U))$, then there is a class on level $i-1$ that has label $gl(U)$.
(6) If $gl(U) < gl(V)$ then $\texttt{first}(gl(U))$ is before $\texttt{first}(gl(V))$.

As stated above, the label m represents the proposition that the lexicographically minimal *infinite* branch going through $\texttt{first}(m)$ is accepting. Every time we pass through an F-child, this is evidence towards this proposition. When a class with label m has two children, we conservatively follow the non-F-child. If the non-F-child dies out, we revise our guess and move to a descendant of the F-child. Thus revising our guess indicates that at an earlier point the branch did visit an F-child, and also provides evidence towards this proposition. Formally, we say that a label m is *successful on level i* if there is a class U on level $i-1$ and a class U' on level i such that $gl(U) = gl(U') = m$, and either U' is the F-child of U, or U' is not a child of U at all.

Example 4.3. In Figure 1(b), the only infinite branch $\{\langle q,0\rangle\}, \{\langle p,1\rangle\}, \ldots$ is accepting. At level 0 this branch is labeled with 0. At each level $i > 0$, we conservatively assume that the infinite branch beginning $\langle q,0\rangle$ goes through $\{\langle q,i\rangle\}$, and thus label $\{\langle q,i\rangle\}$ by 0. As $\{\langle q,i\rangle\}$ is proven finite on level $i+1$, we revise our assumption and continue to follow the path through $\{\langle p,i\rangle\}$. Since $\{\langle p,i\rangle\}$ is an F-class, the label 0 is successful on every level $i+1$. Although the infinite branch is not labeled 0 after the first level, the label 0 asymptotically approaches the infinite branch, checking along the way that the branch is lexicographically minimal among the infinite branches through the root.

Theorem 4.4 demonstrates that the global labeling captures the accepting or rejecting nature of T. Intuitively, at each level the class U with label m is on the lexicographically minimal branch from $\texttt{first}(m)$. If U is on the lexicographically minimal *infinite* branch from $\texttt{first}(m)$, the label m is waiting for the branch to next reach an F-class. If U is not on the lexicographically minimal infinite branch from $\texttt{first}(m)$, then U is finite and m is waiting for U to die out.

Theorem 4.4. *A profile tree T is accepting iff there is a label m that is successful infinitely often.*

Proof: In one first direction, assume there is a label m that is successful infinitely often. The label m can be successful only when it occurs, and thus m occurs infinitely often, $\texttt{first}(m)$ has infinitely many descendants, and there is at least one infinite branch through $\texttt{first}(m)$. Let $b = U_0, U_1, \ldots$ be the lexicographically minimal infinite branch that goes through $\texttt{first}(m)$. We demonstrate that b cannot have a suffix consisting solely of non-F-classes, and therefore is an accepting branch. By way of contradiction, assume there is an index j so that for every $k > j$, the class U_k is a non-F-class. By Lemma 4.2.(4), $\texttt{first}(m)$ is an F-class or the root and thus occurs before level j.

Let $\mathcal{U} = \{V \mid V \prec_j U_j,\ V \text{ is a descendant of } \texttt{first}(m)\}$ be the set of descendants of $\texttt{first}(m)$, on level j, that are lexicographically smaller than U_j. Since b is the lexicographically minimal infinite branch through $\texttt{first}(m)$, every class in $\mathcal{U}$ must be finite. Let $j' \geq j$ be the level at which the last class in $\mathcal{U}$ dies out. At this point, $U_{j'}$ is the lexicographically minimal descendant of $\texttt{first}(m)$. If $gl(U_{j'}) \neq m$, then there is no class on level j' with label m, and, by Lemma 4.2.(5), m would not occur after level j'. Since m occurs infinitely often, it must be that $gl(U_{j'}) = m$. On every level $k > j'$, the class U_k is a non-F-child, and thus U_k is the lexicographically minimal descendant of $U_{j'}$ on level k and so $gl(U_k) = m$. This entails m cannot be not successful after level j', and we have reached a contradiction. Therefore, there is no such rejecting suffix of b, and b must be an accepting branch.

In the other direction, if there is an infinite accepting branch, then let $b = U_0, U_1, \ldots$ be the lexicographically minimal infinite accepting branch. Let B' be the set of infinite branches that are lexicographically smaller than b. Every branch in B' must be rejecting, or b would not be the minimal infinite accepting branch. Let j be the first index after which the last branch in B' splits from b. Note that either $j = 0$, or U_{j-1} is part of an infinite rejecting branch $U_0, \ldots, U_{j-1}, V_j, V_{j+1}, \ldots$ smaller than b. In both cases, we show that U_j is the first class for a new label m that occurs on every level $k > j$ of T.

If $j = 0$, then let $m = 0$. As m is the smallest label, and there is a descendant of U_j on every level of T, it holds that m will occur on every level. In the second case, where $j > 0$, then V_j must be the non-F-child of U_{j-1}, and so U_j is the F-child. Thus, U_j is given a new label m where $U_j = \texttt{first}(m)$. For every label $m' < m$ and level $k > j$, since for every descendant U' of U_j it holds that $V_k \preceq_k U'$, it cannot be that $\texttt{lmd}(m', k)$ is a descendant of U_j. Thus, on every level $k > j$, the lexicographically minimal descendant of U_j will be labeled m, and m occurs on every level of T.

We show that m is successful infinitely often by defining an infinite sequence of levels, $j_0, j_1, j_2, \ldots$ so that m is successful on j_i for all $i > 0$. As a base case, let $j_0 = j$. Inductively, at level j_i, let U' be the class on level j_i labeled with m. We have two cases. If $U' \neq U_{j_i}$, then as all infinite branches smaller than b have already split from b, U' must be finite in T. Let j_{i+1} be the level at which U' dies out. At level j_{i+1}, m will return to a descendant of U_{j_0}, and m will be successful. In the second case, $U' = U_{j_i}$. Take the first $k > j_i$ so that U_k is an F-class. As b is an accepting branch, such a k must exist. As every class between U_j and U_k is a non-F-class, $gl(U_{k-1}) = m$. If U_k is the only child of U_{k-1} then let $j_{i+1} = k$: since $gl(U_k) = m$ and U_k is not the non-F-child of U_{k-1}, it holds that m is successful on level k. Otherwise let U'_k be the non-F-child of U_{k-1}, so that $gl(U'_k) = m$. Again, U'_k is finite. Let j_{i+1} be the level at which U'_k dies out. At level j_{i+1}, the label m will return to a descendant of U_k, and m will be successful. □

4.2 Determining Lexicographically Minimal Descendants

Recall that the definition of the labeling gl involves the computation of $\mathtt{lmd}(m,i)$, the class with the minimal profile among all the descendants of $\mathtt{first}(m)$ on level i. Finding $\mathtt{lmd}(m,i)$ requires knowing the descendants of $\mathtt{first}(m)$ on level i. We show how to store this information with a partial order, denoted $\lessdot_i$, over classes that tracks which classes are minimal cousins of other classes. Using this partial order, we can determine the class $\mathtt{lmd}(m,i+1)$ for every label m that occurs on level i, using only information about levels i and $i+1$ of T. Lemma 4.2.(5) implies that we can safely restrict ourselves to labels that occur on level i.

Definition 4.5. For two classes U and V on level i of T, say that U is a *minimal cousin* of V, written $U \lessdot_i V$, iff V is a descendant of $\mathtt{first}(gl(U))$. Say $U \lessdot_i V$ when $U \lessdot_i V$ and $U \neq V$.

For a label m and level i, we can determine $\mathtt{lmd}(m,i+1)$ given only the classes on levels i and $i+1$ and the partial order $\lessdot_i$. Let U be a class U on level i. Because labels can move between branches, the minimal descendant of $\mathtt{first}(gl(U))$ on level $i+1$ may be a nephew of U, not necessarily a direct descendant. Define the $\lessdot_i$-nephew of U as $\mathtt{neph}_i(U) = \mathtt{min}_{\preceq_{i+1}}(\{V' \mid V \text{ is the parent of } V' \text{ and } U \lessdot_i V\})$.

Lemma 4.6. *For a class U on level i of T, it holds that* $\mathtt{lmd}(gl(U),i+1) = \mathtt{neph}_i(U)$.

Proof: We prove that $\{V' \mid V \text{ is the parent of } V' \text{ and } U \lessdot_i V\}$ contains every descendant of $\mathtt{first}(gl(U))$ on level $i+1$, and thus that its minimal element is $\mathtt{lmd}(gl(U),i+1)$. Let V' be a class on level $i+1$, with parent V on level i. If $U \lessdot_i V$, then V is a descendant of $\mathtt{first}(gl(U))$ and V' is likewise a descendant of $\mathtt{first}(gl(U))$. Conversely, as $gl(U)$ exists on level i, if V' is a descendant of $\mathtt{first}(gl(U))$, then its parent V must also be a descendant of $\mathtt{first}(gl(U))$ and $U \lessdot_i V$. □

By using $\mathtt{neph}_i$, we can in turn define the set of valid labels for a class U' on level $i+1$. Formally, define the $\lessdot_i$-uncles of U' as $\mathtt{unc}_i(U') = \{U \mid U' = \mathtt{neph}_i(U)\}$. Lemma 4.7 demonstrates how $\mathtt{unc}_i$ corresponds to $\mathtt{labels}$.

Lemma 4.7. *Consider a class U' on level $i+1$. The following hold:*

(1) $\mathtt{labels}(U') \cap \{gl(V) \mid V \text{ on level } i\} = \{gl(U) \mid U \in \mathtt{unc}_i(U')\}$.

(2) $\mathtt{labels}(U') = \emptyset$ *iff* $\mathtt{unc}_i(U') = \emptyset$.

Proof:

(1) Let U be a class on level i. By definition, $gl(U) \in \mathtt{labels}(U')$ iff $U' = \mathtt{lmd}(gl(U),i+1)$. By Lemma 4.6, it holds that $\mathtt{lmd}(gl(U),i+1) = \mathtt{neph}_i(U)$. By the definition of $\mathtt{unc}_i$, we have that $U' = \mathtt{neph}_i(U)$ iff $U \in \mathtt{unc}_i(U')$. Thus every label in $\mathtt{labels}(U')$ that occurs on level i labels some node in $\mathtt{unc}_i(U')$.

(2) If $\mathtt{unc}_i(U') \neq \emptyset$, then part (1) implies $\mathtt{labels}(U') \neq \emptyset$. In other direction, let $m = \mathtt{min}(\mathtt{labels}(U'))$. By Lemma 4.2.(5), there is a U on level i so that $gl(U) = m$, and by part (1) $U \in \mathtt{unc}_i(U')$.

□

Finally, we demonstrate how to compute $\lessdot_{i+1}$ only using information about the level i of T and the labeling for level $i+1$. As the labeling depends only on $\lessdot_i$, this removes the final piece of global information used in defining gl.

Lemma 4.8. *Let U' and V' be two classes on level $i+1$ of T, where $U' \neq V'$. Let V be the parent of V'. We have that $U' \lessdot_{i+1} V'$ iff there exists a class U on level i so that $gl(U) = gl(U')$ and $U \lessdot_i V$.*

Proof: If there is no class U on level i so that $gl(U) = gl(U')$, then $U' = \mathtt{first}(gl(U'))$. Since V' is not a descendant of U', it cannot be that $U' \lessdot_{i+1} V'$. If such a class U exists, then $U \lessdot_i V$ iff V is a descendant of $\mathtt{first}(gl(U))$, which is true iff V' is a descendant of $\mathtt{first}(gl(U'))$: the definition of $U' \lessdot_{i+1} V'$. □

4.3 Reusing Labels

As defined, the labeling function gl uses an unbounded number of labels. However, as there are at most $|Q|$ classes on a level, there are at most $|Q|$ labels in use on a level. We can thus use a fixed set of labels by reusing dead labels. For convenience, we use $2|Q|$ labels, so that we never need reuse a label that was in use on the previous level. We demonstrate how to use $|Q|-1$ labels in Appendix B. There are two barriers to reusing labelings. First, we can no longer take the numerically minimal element of $\texttt{labels}(U)$ as the label of U. Instead, we calculate which label is the oldest through $\preceq$. Second, we must ensure that a label that is good infinitely often is not reused infinitely often. To do this, we introduce a Rabin condition to reset each label before we reuse it.

We inductively define a sequence of labelings, l_i, each from the ith level of T to $\{0,\ldots,2|Q|\}$. As a base case, there is only one equivalence class U on level 0 of T, and define $l_0(U)=0$. Inductively, given the set of classes $\mathcal{U}_i$ on level i, the function l_i, and the set of classes $\mathcal{U}_{i+1}$ on level $i+1$, we define l_{i+1} as follows. Define the set of unused labels $\mathrm{FL}(l_i)$ to be $\{m \mid m \text{ is not in the range of } l_i\}$. As T has bounded width $|Q|$, we have that $|Q| \leq |\mathrm{FL}(l_i)|$. Let $\texttt{mj}_{i+1}$ be the $\langle \preceq_{i+1}, < \rangle$-minjection from $\{U' \text{ on level i+1} \mid \texttt{unc}_i(U')=\emptyset\}$ to $\mathrm{FL}(l_i)$. Finally, define the labeling l_{i+1} as

$$l_{i+1}(U') = \begin{cases} l_i(\texttt{min}_{\preceq_i}(\texttt{unc}_i(U'))) & \text{if } \texttt{unc}_i(U') \neq \emptyset, \\ \texttt{mj}_{i+1}(U') & \text{if } \texttt{unc}_i(U') = \emptyset. \end{cases}$$

Because we are reusing labels, we need to ensure that a label that is good infinitely often is not reused infinitely often. Say that a label m is *bad in* l_i if $m \notin \mathrm{FL}(l_{i-1})$, but $m \in \mathrm{FL}(l_i)$. We say that a label m is *good in* l_i if there is a class U on level $i-1$ and a class U' on level i such that $l_{i-1}(U) = l_i(U') = m$ and U' is either the F-child of U or is not a child of U at all.

Theorem 4.9 demonstrates that the Rabin condition of a label being good infinitely often, but bad only finitely often, is a necessary and sufficient condition to T being accepting. The proof, given in Appendix A, associates each label m in gl with the label $l_i(\texttt{first}(m))$.

Theorem 4.9. *A profile tree T is accepting iff there is a label m where $\{i \mid m \text{ is bad in } l_i\}$ is finite, and $\{i \mid m \text{ is good in } l_i\}$ is infinite.*

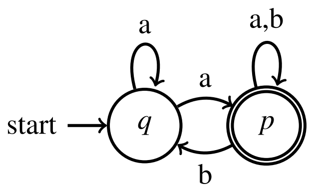
5 A New Determinization Construction for Büchi Automata

In this section we present a determinization construction for $\mathcal{A}$ based on the profile tree T. For clarity, we call the states of our deterministic automaton *macrostates*.

Definition 5.1. Macrostates over $\mathcal{A}$ are six-tuples $\langle S, \preceq, l, \lessdot, G, B\rangle$ where:

- $S \subseteq Q$ is a set of states.
- $\preceq$ is a linear preorder over S.
- $l\colon S \to \{0,\ldots,2|Q|\}$ is a labeling.
- $\lessdot \subseteq \preceq$ is another preorder over S.
- G, B are sets of good and bad labels used for the Rabin condition.

For two states q and r in Q, we say that $q \approx r$ if $q \preceq r$ and $r \preceq q$. We constrain the labeling l so that it characterizes the equivalence classes of S under $\preceq$, and the preorder $\lessdot$ to be a partial order over the equivalence classes of $\preceq$. Let $\mathbf{Q}$ be the set of macrostates.



(a) An automaton $\mathcal{B}$

$$\mathbf{q}_0 = \boxed{\langle \{q\}^0 \rangle,\ \emptyset,\ G = \emptyset,\ B = \emptyset}$$

$$\mathbf{q}_1 = \boxed{\langle \{q\}^0 \prec \{p\}^1 \rangle,\ q \lessdot p,\ G = \emptyset,\ B = \emptyset}$$

$$\mathbf{q}_2 = \boxed{\langle \{q\}^0 \prec \{p\}^2 \rangle,\ q \lessdot p,\ G = \{0\},\ B = \{1\}}$$

$$\mathbf{q}_3 = \boxed{\langle \{q\}^0 \prec \{p\}^1 \rangle,\ q \lessdot p,\ G = \{0\},\ B = \{2\}}$$

(b) The first four macrostates in the run of $D^R(\mathcal{B})$ on ab^ω.

Figure 2: An automaton and four macrostates. For each macrostate $\langle S, \preceq, l, \lessdot, G, B \rangle$, we first display the equivalence classes of S under $\preceq$ in angle brackets, superscripted with the labels of l. We then display the $\lessdot$ relation, and finally the sets G and B.

Before defining transitions between macrostates, we reproduce the pruning of edges from G' by restricting the transition function ρ with respect to S and $\preceq$. For a state $q \in S$ and $\sigma \in \Sigma$, let $\rho_{S,\preceq}(q,\sigma) = \{q' \in \rho(q,\sigma) \mid \text{for every } r \in \rho^{-1}(q',\sigma) \cap S,\ r \preceq q\}$. Thus, when a state has multiple incoming σ-transitions from S, the function $\rho_{S,\preceq}$ keeps only the transitions from states maximal under the $\preceq$ relation. For every state $q' \in \rho(S,\sigma)$, the set $\rho^{-1}_{S,\preceq}(q',\sigma) \cap S$ is an equivalence class under $\preceq$. We note that $\rho(S,\sigma) = \rho_{S,\preceq}(S,\sigma)$.

Example 5.2. Figure 2 displays the first four macrostates in a run of this determinization construction. Consider the state $\mathbf{q}_1 = \langle \{q,p\}, \preceq, l, \lessdot, \emptyset, \emptyset \rangle$ where $q \prec p$, $q \lessdot p$, $l(q) = 0$, and $l(p) = 1$. We have $\rho(q,a) = \{p,q\}$. However, $p \in \rho(p,a)$ and $q \prec p$. Thus we discard the transition from q to p, and $\rho_{S,\preceq}(q,a) = \{q\}$. In contrast, $\rho_{S,\preceq}(p,a) = \rho(p,a) = \{p\}$, because while $p \in \rho(q,a)$, it holds that $q \prec p$.

For $\sigma \in \Sigma$, we define the σ-successor of $\langle S, \preceq, l, \lessdot, G, B \rangle$ to be $\langle S', \preceq', l', \lessdot', G', B' \rangle$ as follows. First, $S' = \rho(S,\sigma)$. Second, define $\preceq'$ as follows. For states $q', r' \in S'$, let $q \in \rho^{-1}_{S,\preceq}(q',\sigma) \cap S$ and $r \in \rho^{-1}_{S,\preceq}(r',\sigma) \cap S$. As the parents of q' and r' under $\rho_{S,\preceq}$ are equivalence classes the choice of q and r is arbitrary.

- If $q \prec r$, then $q' \prec' r'$.
- If $q \approx r$ and $q' \in F$ iff $r' \in F$, then $q' \approx' r'$.
- If $q \approx r$, $q' \notin F$, and $r' \in F$, then $q' \prec' r'$.

Example 5.3. As a running example we detail the transition from $\mathbf{q}_1 = \langle \{q,p\}, \preceq, l, \lessdot, \emptyset, \emptyset \rangle$ to $\mathbf{q}_2 = \langle S', \preceq', l', \lessdot', G', B' \rangle$ on b. We have $S' = \rho(\{q,p\}, b) = \{q,p\}$. To determine $\preceq'$, we note that $p \in S$ is the parent of both $q \in S'$ and $p \in S'$. Since $q \notin F$, and $p \in F$, we have $q \prec' p$.

Third, we define the labeling l' as follows. As in the profile tree T, on each level we give the label m to the minimal descendants, under the $\preceq$ relation, of the first equivalence class to be labeled m. For a state $q \in S$, define the *nephews of* q to be $\mathtt{neph}(q,\sigma) = \mathtt{min}_{\preceq'}(\rho_{S,\preceq}(\{r \in S \mid q \lessdot r\}, \sigma))$. Conversely, for a state $r' \in S'$ we define the *uncles of* r' to be be $\mathtt{unc}(r',\sigma) = \mathtt{min}_{\preceq}(\{q \mid r' \in \mathtt{neph}(q,\sigma)\})$.

Each state $r' \in S'$ inherits the oldest label from its uncles. If r' has no uncles, it gets a fresh label. Let $\mathrm{FL}(l) = \{m \mid m \text{ not in the range of } l\}$ be the free labels in l, and let $\mathtt{mj}$ be the $\langle \preceq', < \rangle$-minjection from $\{r' \in S' \mid \mathtt{unc}(r',\sigma) = \emptyset\}$ to $\mathrm{FL}(l)$, where $<$ is the standard order on $\{0, \ldots, 2|Q|\}$. Let

$$l'(r') = \begin{cases} l(q), \ \text{ for some } q \in \mathtt{unc}(r',\sigma) & \text{if } \mathtt{unc}(r',\sigma) \neq \emptyset, \\ \mathtt{mj}(r') & \text{if } \mathtt{unc}(r',\sigma) = \emptyset. \end{cases}$$

Example 5.4. The nephews of $q \in S$ is the $\preceq'$-minimal subset of the set $\rho_{S,\preceq}(\{r \in S \mid q \lessdot r\}, \sigma)$. Since $q \lessdot q$ and $q \lessdot p$, we have that $\mathtt{neph}(q,b) = \mathtt{min}_{\preceq'}(\{q,p\}) = \{q\}$. Similarly, for $p \in S$ we have $p \lessdot p$ and $\mathtt{neph}(p,b) = \mathtt{min}_{\preceq'}(\{p,q\}) = \{q\}$. Thus for $q \in S'$, we have $\mathtt{unc}(q,b) = \mathtt{min}_{\preceq}(\{p,q\}) = \{q\}$ and we set $l'(q) = l(q) = 0$. For $p \in S'$, we have $\mathtt{unc}(p,b) = \emptyset$ and we set $l'(p)$ to the first unused label: $l'(p) = 2$.

Fourth, define the preorder $\lessdot'$ as follows. For states $q', r' \in S'$, define $q' \lessdot' r'$ iff $q' \approx' r'$ or there exists $q, r \in S$ so that: $r' \in \rho_{S,\preceq}(r,\sigma)$; $q \in \mathtt{unc}(q',\sigma)$; and $q \lessdot r$. The labeling l' depends on recalling which states descend from the first equivalence class with a given label, and $\lessdot'$ tracks these descendants.

Finally, for a label m let $S_m = \{r \in S \mid l(r) = m\}$ and $S'_m = \{r' \in S' \mid l'(r') = m\}$ be the states in S, resp S', labeled with m. Recall that a label m is good either when the branch it is following visits F-states, or the branch dies and it moves to another branch. Thus say m is *good* when: $S_m \neq \emptyset$; $S'_m \neq \emptyset$; and either $S'_m \subseteq F$ or $\rho_{S,\preceq}(S_m,\sigma) \cap S'_m = \emptyset$. G' is then $\{m \mid m \text{ is good}\}$. Conversely, a label is bad when it occurs in S, but not in S'. Thus the set of *bad* labels is $B' = \{m \mid S_m \neq \emptyset,\ S'_m = \emptyset\}$.

Example 5.5. As $p \in \rho_{S,\preceq}(p,b)$; $q \in \mathtt{unc}(q,b)$; and $q \lessdot p$, we have $q \lessdot' p$. Since $l(q) = 0$ and $l'(q) = 0$, but $q \notin \rho_{S,\preceq}(q,b)$, we have $0 \in G'$, and as nothing is labeled 1 in l', we have $1 \in B'$.

Lemma 5.6, proven in Appendix A states that $\langle S', \preceq', l', \lessdot', G', B' \rangle$ is a valid macrostate.

Lemma 5.6. *For a macrostate $\mathbf{q} \in \mathbf{Q}$ and $\sigma \in \Sigma$, the σ-successor of $\mathbf{q}$ is a macrostate.*

Definition 5.7. Define the DRW automaton $D^R(\mathcal{A})$ to be $\langle \Sigma, \mathbf{Q}, \mathbf{Q}^{in}, \rho_{\mathbf{Q}}, \alpha \rangle$, where:

- $\mathbf{Q}^{in} = \{\langle Q^{in}, \preceq_0, l_0, \lessdot_0, \emptyset, \emptyset \rangle\}$, where:
  - $\preceq_0 = \lessdot_0 = Q^{in} \times Q^{in}$
  - $l_0(q) = 0$ for all $q \in Q^{in}$
- For $\mathbf{q} \in \mathbf{Q}$ and $\sigma \in \Sigma$, let $\rho_{\mathbf{Q}}(\mathbf{q},\sigma) = \{\mathbf{q}'\}$, where $\mathbf{q}'$ is the σ-successor of $\mathbf{q}$
- $\alpha = \langle G_0, B_0 \rangle, \ldots, \langle G_{2|Q|}, B_{2|Q|} \rangle$, where for a label $m \in \{0, \ldots, 2|Q|\}$:
  - $G_m = \{\langle S, \preceq, l, \lessdot, G, B \rangle \mid m \in G\}$
  - $B_m = \{\langle S, \preceq, l, \lessdot, G, B \rangle \mid m \in B\}$

Theorem 5.8, proven in Appendix A, asserts the correctness of the construction and says that its blowup is comparable with known determinization constructions.

Theorem 5.8. *For an NBW $\mathcal{A}$ with n states, $L(D^R(\mathcal{A})) = L(\mathcal{A})$ and $D^R(\mathcal{A})$ has $n^{O(n)}$ states.*

There are two simple improvements to the new construction, detailed in Appendix B. First, we do not need $2|Q|$ labels: it is sufficient to use $|Q| - 1$ labels. Second, Piterman's technique of dynamic renaming can reduce the Rabin condition to a parity condition.

6 Discussion

In this paper we extended the notion of profiles from [6] and developed a theory of profile trees. This theory affords a novel determinization construction, where determinized-automaton states are sets of input-automaton states augmented with two preorders. In the future, a more thorough analysis could likely improve the upper bound on the size of our construction. We hope to see heuristic optimization techniques developed for this construction, just as heuristic optimization techniques were developed for Safra's construction [24].

More significantly, profile trees afford us the first theoretical underpinnings for determinization. Decades of research on Büchi determinization have resulted in a plethora of constructions, but a paucity of mathematical structures underlying their correctness. This is the first new major line of research in Büchi determinization since 1995, and we expect it to lead to further research in this important area.

One important question is to understand better the connection between profile trees and Safra's construction. A key step in the transition between Safra trees is to remove states if they appear in more than one node. This seems analogous to the pruning of edges from G'. The second preorder in our construction, namely the relation $\lesssim_i$, seems to encodes the order information embedded in Safra trees. Perhaps our approach could lead to declarative definition of constructions based on Safra and Muller-Schupp trees. In any case, it is our hope that profile trees will encourage the development of new methods to analyze and optimize determinization constructions.

References

[1] C.S. Althoff, W. Thomas, and N. Wallmeier. Observations on determinization of Büchi automata. In *ICALP*, 2005.

[2] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *J. ACM*, 2002.

[3] J.R. Büchi. On a decision method in restricted second order arithmetic. In *ICLMPS*, 1962.

[4] C. Courcoubetis and M. Yannakakis. The complexity of probabilistic verification. *J. ACM*, 1995.

[5] L. Doyen and J.-F. Raskin. Improved algorithms for the automata-based approach to model-checking. In *TACAS*, 2007.

[6] S. Fogarty, O. Kupferman, M.Y. Vardi, and Th. Wilke. Unifying Büchi complementation constructions. In *CSL*, 2011.

[7] S. Fogarty and M.Y. Vardi. Efficient Büchi universality checking. In *TACAS*, 2010.

[8] E. Friedgut, O. Kupferman, and M.Y. Vardi. Büchi complementation made tighter. *IJFCS*, 2006.

[9] D. Kähler and Th. Wilke. Complementation, disambiguation, and determinization of Büchi automata unified. In *ICALP*, 2008.

[10] J. Kretínský and J. Esparza. Deterministic automata for the (f, g)-fragment of LTL. In *CAV*, 2012.

[11] O. Kupferman and M.Y. Vardi. Weak alternating automata are not that weak. *TOCL*, 2001.

[12] O. Kupferman and M.Y. Vardi. Safraless decision procedures. In *FOCS*, 2005.

[13] L.H. Landweber. Decision problems for ω–automata. *MST*, 1969.

[14] C. Löding. Optimal bounds for the transformation of omega-automata. In *FSTTCS*, 1999.

[15] R. McNaughton. Testing and generating infinite sequences by a finite automaton. *ICONT*, 1966.

[16] D.E. Muller and P.E. Schupp. Simulating alternating tree automata by nondeterministic automata: New results and new proofs of theorems of Rabin, McNaughton and Safra. *TCS*, 1995.

[17] N. Piterman. From nondeterministic Büchi and Streett automata to deterministic parity automata. In *LICS*, 2006.

[18] A. Pnueli and R. Rosner. On the synthesis of a reactive module. In *POPL*, 1989.

[19] M.O. Rabin and D. Scott. Finite automata and their decision problems. *IBM JRD*, 1959.

[20] S. Safra. On the complexity of ω-automata. In *FOCS*, 1988.

[21] S. Schewe. Büchi complementation made tight. In *STACS*, 2009.

[22] S. Schewe. Tighter bounds for the determinisation of Büchi automata. In *FOSSACS*, 2009.

[23] S. Tasiran, R. Hojati, and R.K. Brayton. Language containment using non-deterministic omega-automata. In *CHARME*, 1995.

[24] M.-H. Tsai, S. Fogarty, M. Y. Vardi, and Y.-K. Tsay. State of Büchi complementation. In *CIAA*, 2010.

[25] M.Y. Vardi. Expected properties of set partitions. Research report, The Weizmann Institute of Science, 1980.

[26] M.Y. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *FOCS*, 1985.

A Proofs

A.1 Lemma 4.2

Lemma 4.2. *For classes U and V on level i of T, it holds that:*

(1) If $U \neq V$ then $gl(U) \neq gl(V)$.

(2) U is a descendant of $\texttt{first}(gl(U))$.

(3) If U is a descendant of $\texttt{first}(gl(V))$, then $V \preceq_i U$. Consequently, if $U \prec_i V$, then U is not a descendant of $\texttt{first}(gl(V))$.

(4) $\texttt{first}(gl(U))$ is the root or an F-class with a sibling.

(5) If $U \neq \texttt{first}(gl(U))$, then there is a class on level $i-1$ that has label $gl(U)$.

(6) If $gl(U) < gl(V)$ then $\texttt{first}(gl(U))$ is before $\texttt{first}(gl(V))$.

Proof: Parts (1) through (3) follow immediately from the fact that $U = \texttt{lmd}(gl(U),i)$. Part (4) follows from the fact that, for every class V on level i with non-F-child V', we have $V' = \texttt{lmd}(gl(U),i+1)$. Part (6) follows from the definition of $\texttt{labels}$: a new label is always larger than any earlier label. Finally, we prove part (5).

Assume U', on level i, is such that $gl(U') = m$ and $U' \neq \texttt{first}(m)$. By Part (2), there must be a descendant of $\texttt{first}(m)$ on level $i-1$. Let $U = \texttt{lmd}(m,i-1)$. To prove $gl(U) = m$, we show $m = \texttt{min}(\texttt{labels}(U))$. Consider $m' < m$ such that U is a descendant of $\texttt{first}(m')$, and thus U' is also descendant of $\texttt{first}(m')$. By Part (6), $\texttt{first}(m')$ occurs before $\texttt{first}(m)$. Since U is a descendant of both $\texttt{first}(m)$ and $\texttt{first}(m')$, it must be that $\texttt{first}(m)$ is a descendant of $\texttt{first}(m')$.

Since $m' < m$, if $U' = \texttt{lmd}(m',i)$ then $gl(U')$ would be m'. There must then exist a $V' \prec_i U'$ that is a descendant of $\texttt{first}(m')$, but not a descendant of $\texttt{first}(m)$. By the definition of lexicographic order, V' is lexicographically smaller than every descendant, on level i, of $\texttt{first}(m)$. Let V be the parent of V'. We have that V is a descendant of m' that is lexicographically smaller than every descendant, on level $i-1$, of $\texttt{first}(m)$. In specific, $V \prec_{i-1} U$, and thus $U \neq \texttt{lmd}(m',i)$. Thus $m = \texttt{min}(\texttt{labels}(U)) = gl(U)$. □

A.2 Theorem 4.9

To show a correlation between the labeling in Section 4 and the labeling here, we define a mapping, f, from the labels of l to $\{0,\ldots,2|Q|\}$. For a label m, where $\texttt{first}(m)$ occurs on level i, let $f(m) = l_i(\texttt{first}(m))$.

SF: f is overloaded: we should change this for the full version, but I think the overload is acceptable for an appendix.

Lemma A.1. *For classes U on level i and U' on level $i+1$, if $gl(U) = gl(U')$, then $l_i(U) = l_{i+1}(U') = f(gl(U))$.*

Proof: Let k be the number of levels between U and $\texttt{first}(gl(U))$. We prove this lemma by induction over k. As a base case, if $k = 0$, then $U = \texttt{first}(gl(U))$ and by definition $f(gl(U)) = l_i(U)$. Inductively, assume $k > 0$, and assume this lemma holds for every V at most $k-1$ steps removed from $\texttt{first}(gl(V))$. Since $k > 0$, then $U \neq \texttt{first}(gl(U))$. Let V be the node on level $i-1$ such that $gl(V) = gl(U)$. By the inductive hypothesis, $l_{i-1}(V) = l_i(U)$. Further, since $\texttt{first}(gl(V)) = \texttt{first}(gl(U))$, we have $l_i(U) = f(gl(U))$. We now show that $U = \texttt{min}_{\preceq_i}(\texttt{unc}_i(U'))$.

As $gl(U) = gl(U')$, we have that $gl(U) \in \texttt{labels}(U')$. By Lemma 4.7, this implies $U \in \texttt{unc}_i(U')$. To prove that $U = \texttt{min}_{\preceq_i}(\texttt{unc}_i(U'))$, let $V \in \texttt{unc}_i(U')$ be another class on level i. By Lemma 4.7, this implies $gl(V) \in \texttt{labels}(U')$, and thus $gl(U) < gl(V)$. As U' is a descendant of both $\texttt{first}(gl(U))$ and $\texttt{first}(gl(V))$, one is a descendant of the other. Since $gl(U) < gl(V)$, by Lemma 4.2.(6) it must be that $\texttt{first}(gl(V))$ is a descendant of $\texttt{first}(gl(U))$. Thus V is a descendant of $\texttt{first}(gl(U))$, and by Lemma 4.2.(3) we have $U \preceq V$. Therefore $U = \texttt{min}_{\preceq_i}(\texttt{unc}_i(U'))$, and $l_{i+1}(U') = l_i(U)$. □

Corollary A.2. *For every class U on level i, it holds that $l_i(U) = f(gl(U))$.*

Theorem 4.9. *A profile tree T is accepting iff there is a label m where $\{i \mid m \text{ is bad in } l_i\}$ is finite, and $\{i \mid m \text{ is good in } l_i\}$ is infinite.*

Proof: We prove a relation with Theorem 4.4. For the first direction, let m be a label that is successful infinitely often. We prove that $f(m)$ is bad in only finitely many l_i, and is good in infinitely many l_j. Let U on level j be $\texttt{first}(m)$. First, as m occurs on every level $k > j$, Lemma A.1 implies $f(m)$ occurs on k, and thus $f(m)$ is not bad in l_k. Second, let $k > j$ be a level on which m is successful. This implies there exist classes U on level $k-1$ and U' on level k, so that $gl(U) = gl(U') = m$ and U' is not the non-F-child of U. Lemma A.1 implies that $l_{k-1}(U) = l_k(U') = f(m)$, and thus that $f(m)$ is good in l_k. We thus conclude $f(m)$ is good in infinitely many l_k.

For the other direction, let m' be a label that is bad in l_i for finitely many i, and is good in l_i for infinitely many i. Since m' is bad only finitely often, there is some level after which m' is not bad. Let j be the first level after which m' ceases being bad on which m' occurs. This implies m' occurs on every level $k > j$. Let U on level j be such that $l_j(U) = m'$. Since m' does not occur on $j-1$, it must be that $\texttt{unc}_j(U) = \emptyset$: otherwise $l_j(U)$ would be $l_j(\texttt{min}_{\preceq_j}(\texttt{unc}_j(U')))$. Thus by Lemma 4.7 we have that $\texttt{labels}(U) = \emptyset$, and there is a label m in l so that $U = \texttt{first}(m)$, and $f(m) = m'$. By assumption, there are infinitely many $k > j$ so that m' succeeds in l_k. On each of these k's, there is a class U on level $k-1$ and U' on level k so that $l_{k-1}(U) = m'$, $l_k(U') = m'$, and U' is not the non-F-child of U. By Corollary A.2, $m = gl(U)$ is good on level k, and m is good infinitely often. □

A.3 Connecting T to $D^R(\mathcal{A})$.

In this Appendix we prove the machinery of $D^R(\mathcal{A})$ matches the inductive definitions of labeling over T. We first prove the the transitions of $D^R(\mathcal{A})$ are valid.

Lemma 5.6. *For a macrostate $\mathbf{q} \in \mathbf{Q}$ and $\sigma \in \Sigma$, the σ-successor of $\mathbf{q}$ is a macrostate.*

Proof: As $\langle S, \preceq, l, \lessdot, G, B\rangle$ is a macrostate, we have $\preceq$ is a linear preorder, $\lessdot \subseteq \preceq$, and for every $q,r,s,t \in S$: $q \approx r$ iff $l(q) = l(r)$; $q \approx r$ iff $q \lessdot r$ and $r \lessdot q$; and if $q \approx r$, $s \approx t$, and $q \lessdot s$, then $r \lessdot t$. We must prove this also holds for $\lessdot'$, $\preceq'$, and l' over states in S'. Below, let q', r', s', t' be states in S', and $q,r,s,t \in S$ be such that $q' \in \rho_{S,\preceq}(q,\sigma)$, $r' \in \rho_{S,\preceq}(r,\sigma)$, $s' \in \rho_{S,\preceq}(s,\sigma)$, and $t' \in \rho_{S,\preceq}(t,\sigma)$.

To demonstrate that $\preceq'$ is a linear preorder, we show it is reflexive, relates every two elements, and is transitive. That $\preceq'$ is reflexive follows from the definition. To show that $\preceq'$ relates every two elements, note that as $\preceq$ is a linear preorder, either $q \prec r$, $r \prec q$, or $q \approx r$. By the definition of $\preceq'$, either $q' \prec' r'$, $q' \approx' r'$, or $r' \prec' q$. To show that $\preceq'$ is transitive, assume $q' \preceq' r' \preceq' s'$. By definition of $\preceq'$ we then have $q \preceq r$ and $r \preceq s$. Since $\preceq$ is transitive, we have $q \preceq s$. In order for $q' \npreceq' s'$, it would need to be that $q \approx s$, $q' \in F$, and $s' \notin F$. If $q \approx s$, then $q \approx r$ and $r \approx s$. Thus if $r' \in F$, we would have $s' \preceq r'$, a contradiction. If $r' \notin F$, we would have $r' \preceq q'$, a contradiction. Thus it cannot be the case that $q \approx s$, $q' \in F$, and $s' \notin F$, and either $q' \approx' s'$, or $q' \prec' s'$, and $\preceq'$ is transitive and a linear preorder.

Next, we prove the labeling must give unique labels to the equivalence classes of S' under $\preceq$': that $q' \approx' r'$ iff $l'(q') = l'(r')$. By the above properties, if $q \approx r$, then $\texttt{neph}(q,\sigma) = \texttt{neph}(r,\sigma)$. Further, $\texttt{neph}(q,\sigma)$ is an equivalence class under $\preceq'$ or is empty. In one direction, let $q' \approx' r'$ and let $m = l'(q')$. That $q' \approx r'$ implies for every $q \in \texttt{unc}(q',\sigma)$ that $q' \approx' r'$, and thus $\texttt{unc}(q',\sigma) = \texttt{unc}(r',\sigma)$. If

$\mathtt{unc}(q',\sigma)=\emptyset$, then $\mathtt{unc}(r',\sigma)=\emptyset$. As a minjection maps equivalent elements to the same value, we have $l(q')=\mathtt{mj}(q')=\mathtt{mj}(r')=l(r')$. Alternately, if $\mathtt{unc}(q',\sigma)\neq\emptyset$ then $m=l(q)$ for $q\in\mathtt{unc}(q',\sigma)$, and $q\in\mathtt{unc}(r',\sigma)$, and $l(r')=m$.

Finally, we must demonstrate two things about $\lessdot'$: that $\lessdot'\subseteq\preceq'$, and that $\lessdot'$ is a partial order over the equivalence classes of $\preceq'$. Assume $q'\lessdot' r'$. If $q'\approx' r'$, then $q'\preceq' r'$. Otherwise there exists a $q_2\in\mathtt{unc}(q')$ so that $l(q_2)=l(q')$ and $q_2\lessdot r$. This implies both $r',q'\in\mathtt{neph}(q_2,\sigma)$. Since $l(q')=l(q_2)$, it must be that $q_2\in\mathtt{unc}(q',\sigma)$ and thus $q'\in\mathtt{neph}(q_2,\sigma)$. Thus $q'\preceq' r'$, and $\lessdot'\subseteq\preceq'$. This implies $q'\approx' r'$ iff $q'\lessdot' r'$ and $r'\lessdot' q'$ It remains to show if $q'\approx' r'$, $s'\approx t'$, and $q'\lessdot' s'$, then $r'\lessdot' t'$. If $q'\approx' s'$, then $r'\approx' t'$ and $r'\lessdot' t'$. Otherwise there exists a q_2 so that $l(q_2)=l(q')$ and $q_2\lessdot s$. Since $r'\approx' q'$, it holds that $l(r')=l(q_2)$. Since $s'\approx' t'$, it holds that $s\approx t$ and $q_2\lessdot t$. Thus $r'\lessdot' t'$, and we have satisfied all requirements for $\langle S',\preceq',l',\lessdot',G,B\rangle$ to be a macrostate. □

Lemma A.3 asserts that the set of states S and the preorder $\preceq$ correspond to the nodes on a level i of G' and the preorder $\preceq_i$. Further, the edges in G' correspond to transitions in $\rho_{S,\preceq}$. The proof relates σ-successors of macrostates and $\preceq_i$.

Lemma A.3. *Let G be the run* DAG *of $\mathcal{A}$ on w and let $\mathbf{q}_i=\langle S,\preceq,l,\lessdot,G,B\rangle$ be the i-th macrostate in the run of $D^R(\mathcal{A})$ on w:*

(1) $S=\{q\mid\langle q,i\rangle\in G\}$.
(2) For $q,r\in S$, it holds that $q\preceq r$ iff $\langle q,i\rangle\preceq_i\langle r,i\rangle$.
(3) For $q\in S$ and $q'\in Q$, it holds that $q'\in\rho_{S,\preceq}(q,\sigma_i)$ iff $\langle\langle q,i\rangle,\langle q',i{+}1\rangle\rangle\in E'$.

Proof:

We proceed by induction over i, at each step proving (1) and (2) for $i+1$, and proving (3) for i. As a base case, for $i=0$, we have $S=Q^{in}$and $\preceq=Q^{in}\times Q^{in}$. As $Q^{in}\cap F=\emptyset$, for every u,v on level 0 of G' $h_u=0=h_v$ and $u\preceq_0 v$. Inductively, assume that (1) and (2) holds for $\mathbf{q}_i=\langle S,\preceq,l,\lessdot,G,B\rangle$, and let $\mathbf{q}_{i+1}=\langle S',\preceq',l',\lessdot',G,B\rangle$ be the σ-successor of $\mathbf{q}_i$. We show that (3) holds for $\mathbf{q}_i$, and (1) (2) holds for $\mathbf{q}_{i+1}$.

(1) As $S'=\rho(S,\sigma_i)$, by the inductive hypothesis and the definition of V we have $S'=\{q'\mid\langle q',i{+}1\rangle\in G\}$.

(2) By definition, $q'\in\rho_{S,\preceq}(q,\sigma_i)$ iff $q'\in\rho(q,\sigma_i)$ and for every $r\in S$, if $q'\in\rho(r,\sigma_i)$ then $r\preceq q$. By the definition of G and the inductive hypothesis, this holds iff $\langle\langle q,i\rangle,\langle q',i{+}1\rangle\rangle\in E$ and for every $\langle r,i\rangle$, if $\langle\langle r,i\rangle,\langle q',i{+}1\rangle\rangle\in E$, then $\langle r,i\rangle\preceq_i\langle q,i\rangle$. This it the definition of $\langle\langle q,i\rangle,\langle q',i{+}1\rangle\rangle\in E'$, and thus (3) holds for $\mathbf{q}_i$.

(3) For $q',r'\in S'$, let q,r be such that $q'\in\rho_{S,\preceq}(q,\sigma_i)$ and $r'\in\rho_{S,\preceq}(r,\sigma_i)$. By the inductive hypothesis, this implies $\langle\langle q,i\rangle,\langle q',i{+}1\rangle\rangle\in E'$ and $\langle\langle r,i\rangle,\langle r',i{+}1\rangle\rangle\in E'$. By the definition of $\preceq'$, it holds that $q'\preceq' r'$ iff (a) $q\prec r$, (b) $q\approx r$ and $q'\in F$ iff $r'\in F$, or (c) $q\approx r$, $q'\notin F$, and $r'\in F$. Recall that f is the function assigning 1 to F-nodes, and 0 to non-F-nodes. By the inductive hypothesis, then, $q'\preceq r'$ iff (a) $\langle q,i\rangle\prec_i\langle r,i\rangle$, (b) $\langle q,i\rangle\approx_i\langle r,i\rangle$ and $f(\langle q',i{+}1\rangle)=f(\langle r',i{+}1\rangle)$, or (c) $\langle q,i\rangle\approx\langle r,i\rangle$, $f(\langle q',i{+}1\rangle)=0$ and $f(\langle r',i{+}1\rangle)=0$. By Lemma 3.1, these are precisely the situations in which $\langle q',i{+}1\rangle\preceq_{i+1}\langle r',i{+}1\rangle$.

□

Lemma A.4 demonstrate the correlation the labeling l' and the labeling l_i of the tree of equivalence classes. Lemma A.5 shows that, so defined, the preorder $\lessdot$ describes the minimal cousin relation of Definition 4.5. We simultaneously prove Lemmas A.4 and A.5 by induction.

Lemma A.4. *Let G be the run* DAG *of $\mathcal{A}$ on w and $\mathbf{q}_i=\langle S,\preceq,l,\lessdot,G,B\rangle$ be the ith macrostate in the run of $D^R(\mathcal{A})$ on w. For $q\in S$, it holds that $l(q)=l_i([\langle q,i\rangle])$.*

Lemma A.5. *Let G be the run* DAG *of $\mathcal{A}$ on w and $\mathbf{q}_i = \langle S, \preceq, l, \lessdot, G, B\rangle$ be the ith macrostate in the run of $D^R(\mathcal{A})$ on w. For $q, r \in S$ it holds that $q \lessdot r$ iff $[\langle q,i\rangle] \lessdot_i [\langle r,i\rangle]$*

Proof: We prove these by induction over i. As a base case, for $i = 0$, we have $S = Q^{in}$, $\lessdot = Q^{in} \times Q^{in}$, and $l(q) = 0$ for every $q \in S$. By definition, the 0th level of G' is $\{\langle q,0\rangle \mid q \in Q^{\in}\}$. As $Q^{in} \cap F = \emptyset$, for every u, v on level 0 of G' $h_u = 0 = h_v$ and $u \preceq_0 v$. Since there is only one equivalence class U, we have $U \lessdot_0 U$, and $l(U) = 0$.

Inductively, assume this holds for $\mathbf{q}_i = \langle S, \preceq, l, \lessdot, G, B\rangle$, and let $\mathbf{q}_{i+1} = \langle S', \preceq', l', \lessdot', G, B\rangle$ be the σ-successor of $\mathbf{q}_i$. Note by Lemma 5.6 that l' gives unique labels to the equivalence classes of $\preceq'$, and $\lessdot'$ is a partial order over the equivalence classes of $\preceq'$.

Proof of Prop. A.4 For $q' \in S'$, we prove $l'(q') = l_{i+1}([\langle q', i{+}1\rangle])$ as follows. First, by definition for $q \in S$ and $r' \in S'$, $r' \in \mathtt{neph}(q,\sigma)$ if there exists $r \in S$ so that $q \lessdot r$ and $r' \in \rho_{S,\preceq}(r,\sigma_i)$. By Lemma A.3.(3), the inductive hypothesis, and the definition of T, this holds if there is a V so that $[\langle r', i{+}1\rangle]$ is a child of V and $[\langle q,i\rangle] \lessdot_i V$: the definition of $[\langle r', i{+}1\rangle] \in \mathtt{neph}_i([\langle q,i\rangle])$. Thus $\mathtt{neph}_i([\langle q,i\rangle]) = \{[\langle r', i{+}1\rangle] \mid r' \in \mathtt{neph}(q,\sigma)\}$. By Lemma A.3.(2) this implies for every $r' \in S'$, $\mathtt{unc}_i([\langle r', i{+}1\rangle]) = \{[\langle q,i\rangle] \mid q \in \mathtt{unc}(r',\sigma)\}$. We have two cases. If $\mathtt{unc}(q',\sigma) \neq \emptyset$, then $\mathtt{unc}_i([\langle q', i{+}1\rangle]) \neq \emptyset$, and $l'(q') = l(q)$ for $q \in \mathtt{unc}(q',\sigma)$. By the inductive hypothesis and Lemma A.3.(2) this implies $[\langle q,i\rangle] = \mathtt{min}_{\preceq_i}(\mathtt{unc}_i([\langle q', i{+}1\rangle]))$ and $l_{i+1}([\langle q', i{+}1\rangle]) = l_i([\langle q,i\rangle]) = l(q)$. Alternately, if $\mathtt{unc}(q',\sigma) = \emptyset$, then $\mathtt{unc}_i([\langle q', i{+}1\rangle]) = \emptyset$. The inductive hypothesis implies that $\mathrm{FL}(l_i)$, the set of unused labels in l_i, is identical to $\mathrm{FL}(l)$, the set of unused labels in l. Thus the $\langle \preceq_{i+1}, <\rangle$-minjection from the classes on level $i{+}1$ of T to $\mathrm{FL}(l_i)$ corresponds to the $\langle \preceq', <\rangle$-minjection from S' to $\mathrm{FL}(l)$, and $\mathtt{mj}(q') = \mathtt{mj}_{i+1}([\langle q, i{+}1\rangle])$.

Proof of Prop. A.5 There are two cases in which $q' \lessdot' r'$. First, if $q' \approx' r'$, then by Lemma A.3.(2) $[\langle q', i{+}1\rangle] = [\langle r', i{+}1\rangle]$ and, as $\lessdot_{i+1}$ is reflexive, $[\langle q', i{+}1\rangle] \lessdot_{i+1} [\langle r', i{+}1\rangle]$. Otherwise $q' \not\approx' r'$ and $q' \lessdot' r'$ iff there exists $r, q_2 \in S$ so that $q_2 \in \mathtt{unc}(q')$, $r' \in \rho_{S,\preceq}(r,\sigma_i)$, and $q_2 \lessdot r$. By Lemma A.3.(2) and (3) this entails $q' \lessdot' r'$ iff there exists U and V so that V is the parent of $[\langle r', i{+}1\rangle]$, $l_i(U) = l_{i+1}([\langle q', i{+}1\rangle])$, and $U \lessdot_i V$. By Lemmas 4.8 and A.1, this is precisely the condition under which $[\langle q', i{+}1\rangle] \lessdot_{i+1} [\langle r', i{+}1\rangle]$. □

Lemma A.6 shows that the presence of a label in G corresponds to the success success of a label in l_i.

Lemma A.6. *Let G be the run* DAG *of $\mathcal{A}$ on w and $\mathbf{q}_i = \langle S, \preceq, l, \lessdot, G, B\rangle$ be the ith macrostate in the run of $D^R(\mathcal{A})$ on w. For every label m, it holds that $m \in G$ iff m is good in l_i and $m \in B$ iff m is bad in l_i.*

Proof: Let $\mathbf{q}_i = \langle S, \preceq, l, \lessdot, G, B\rangle$ and $\mathbf{q}_{i+1} = \langle S', \preceq', l', \lessdot', G, B\rangle$. Recall that, with respect to i, $R = \{r \in S \mid l(r) = m\}$ and $R' = \{r' \in S' \mid l'(r') = m\}$. By Lemma 5.6, we have that R is an equivalence class under $\preceq$ and R' is an equivalence class under $\preceq'$. By definition, m dies in l_i when m is in the range of l_i, but not in the range of l_{i+1}. By Lemma A.4 this is true iff $R \neq \emptyset$, but $R' = \emptyset$: the definition of m dies in $\langle \mathbf{q}_i, \mathbf{q}_{i+1}\rangle$.

Similarly, m succeeds in l_{i+1} if there are classes U on level $i-1$ and U' on level i so that $l_i(U) = l_{i+1}(U') = m$, and U' is not the non-F-child of U. By Lemmas A.3.(1) and A.4, a U and U' exist so that $l_i(U) = l_{i+1}(U') = m$, iff $U = \{\langle r,i\rangle \mid r \in R\}$ and $U' = \{\langle r', i+1\rangle \mid r' \in R'\}$. This entails that m succeeds in l_{i+1} iff $R \neq \emptyset$ and $R' \neq \emptyset$, and either U' is an F-class, or U' is not a child of U. If U' is an F-class then $R' \subseteq F$. If U' is not a child of U, then by the definition of T and Lemma A.3.(3) there is no $r \in R$, $r' \in R'$ where $r' \in \rho_{S,\preceq}(r,\sigma_i)$. This entails $\{\rho_{S,\preceq}(r,\sigma) \mid r \in R\} \cap R' = \emptyset$. We conclude that m succeeds in l_{i+1} iff m succeeds in $\langle \mathbf{q}_i, \mathbf{q}_{i+1}\rangle$. □

Finally, we must bound the number of preorders $\lessdot$ to bound the size of the automaton.

Lemma A.7. *For a level i, the preorder $\lessdot_i$ is a tree order over the classes on level i of T.*

Proof: Let $\mathcal{U}$ be the set of classes on level i of T. By definition, $\lessdot_i$ is a tree order if for every $V \in \mathcal{U}$, the $\{U \mid U \leq V\}$ is totally ordered by $\lessdot_i$. Consider two classes $U \lessdot_i V$ and $W \lessdot_i V$. By definition, V is a descendant of both $\texttt{first}(gl(U))$ and $\texttt{first}(gl(W))$. Since T is a tree, one of $\texttt{first}(gl(U))$ or $\texttt{first}(gl(W))$ is a descendant of the other. Without loss of generality, assume $\texttt{first}(gl(U))$ is a descendant of $\texttt{first}(gl(W))$. Since U is a descendant of $\texttt{first}(gl(U))$, it is a descendant of $\texttt{first}(gl(W))$ too, and $W \lessdot_i U$. □

Theorem 5.8. *For an NBW $\mathcal{A}$ with n states, $L(D^R(\mathcal{A})) = L(\mathcal{A})$ and $D^R(\mathcal{A})$ has $n^{O(n)}$ states.*

Proof: That $L(D^R(\mathcal{A})) = L(\mathcal{A})$ follows from Theorem 4.9 and Lemma A.6. To bound the number of macrostates $\langle S, \preceq, l, \lessdot, G, B\rangle$, we observe that the number of subsets S and linear orders $\preceq$ is $n^{O(n)}$ [25]. The number of labelings is likewise $n^{O(n)}$. By Lemma A.7 and Lemma A.5, $\lessdot$ is a tree-order over the equivalence classes of S under $\preceq$. By Cayley's formula, the number of tree orders is bounded by n^{n-2}. Thus the number of macrostates is bounded by $n^{O(n)}$. □

B Smaller Constructions

We here present two variants of Definition 5.7, both of which only use a variation of macrostates where labels are restricted to $\{0,\ldots,|Q|-1\}$. The first is a *Rabin-edge automaton*, in which the acceptance condition is a set $\langle G_0, B_0\rangle, \ldots, \langle G_k, B_k\rangle$ of pairs of sets of transitions: thus $G_j, B_j \subseteq Q^2$ for $0 \leq j \leq k$. A run is accepting iff there exists $0 \leq j \leq k$ so that $\langle q_i, q_{i+1}\rangle \in G_j$ for infinitely many i's, while $\langle q_i, q_{i+1}\rangle \in B_j$ for only finitely many i's. The second is a *parity-edge automaton*, the acceptance condition is a parity function γ: $Q^2 \to \{0,\ldots,k\}$, and a run is accepting if the smallest element of $\{j \mid j = \gamma(q_i, q_{i+1})$ for infinitely many i's$\}$ is even. Define the set of *tight macrostates* to be four-tuples $\langle S, \preceq, l, \lessdot\rangle$, where S, $\preceq$, and $\lessdot$ are defined as for normal macrostates, and where $l\colon S \to \{0,\ldots,|Q|-1\}$ is a tighter labeling. Let $\mathbf{Q}^t$ be the set of tight macrostates.

B.1 Tight Rabin Variant:

Given a tight macrostate $\mathbf{q} \in \mathbf{Q}^t$ and $\sigma \in \Sigma$, define the *Rabin σ-successor of* $\mathbf{q}$ to be $\mathbf{q}' = \langle S', \preceq', l', \lessdot'\rangle$ where S', $\preceq'$, and $\lessdot'$ are defined as in Section 5, and l' is defined as follows:

(1) For $q \in S$, let $\texttt{neph}(q,\sigma) = \texttt{min}_{\preceq'}(\{r' \mid \text{exists } r \in S,\ q \lessdot r,\ r' \in \rho_{S,\preceq}(r,\sigma)\})$, as in Section 5.

(2) For $r' \in S'$, let $\texttt{unc}(r',\sigma) = \texttt{min}_{\preceq}(\{q \mid r' \in \texttt{neph}(q,\sigma)\})$, as in Section 5.

(3) $\text{FL}(l) = \{m \mid m \text{ not in the range of } l\} \cup \{l(q) \mid \text{ for every } r' \in S', q \notin \texttt{unc}(r',\sigma)\}$.

(4) $\texttt{mj}$ is the $\langle \preceq', <\rangle$-minjection from $\{r' \in S' \mid \texttt{unc}(r', \sigma = \emptyset\}$ to $\text{FL}(l)$.

(5) For $r' \in S'$, let $l'(r') = \begin{cases} l(q),\ q \in \texttt{unc}(r',\sigma) & \text{if } \texttt{unc}(r',\sigma) \neq \emptyset, \\ \texttt{mj}(r') & \text{if } \texttt{unc}(r',\sigma) = \emptyset. \end{cases}$

For $\sigma \in \Sigma$ and label $m \in \{0,\ldots,|Q|-1\}$, given a tight macrostate $\mathbf{q} = \langle S, \preceq, l, \lessdot\rangle \in \mathbf{Q}^t$ and its Rabin σ-successor $\mathbf{q}' = \langle S', \preceq', l', \lessdot'\rangle$ let $R = \langle r \in S \mid l(r) = m\rangle$ and $R' = \langle r' \in S' \mid l'(r') = m\rangle$. Say that *m Rabin-dies in* $\langle \mathbf{q}, \mathbf{q}'\rangle$ when $R \neq \emptyset$ and $m \in \text{FL}(l)$. Say that *m Rabin-succeeds in* $\langle \mathbf{q}, \mathbf{q}'\rangle$ when it does not die in $\langle \mathbf{q}, \mathbf{q}'\rangle$, $R \neq \emptyset$, $R' \neq \emptyset$, and either $R' \subseteq F$ or $\rho_{S,\preceq}(R,\sigma) \cap R' = \emptyset$.

Definition B.1. Define the DREW automata $D^T(\mathcal{A})$ to be $\langle \Sigma, \mathbf{Q}^t, \mathbf{Q}^{in}, \rho_{\mathbf{Q}}, \alpha\rangle$ where:

- $\mathbf{Q}^{in}$ is as defined in Definition 5.7
- For $\mathbf{q} \in \mathbf{Q}^t$ and $\sigma \in \Sigma$, $\rho_{\mathbf{Q}}(\mathbf{q},\sigma) = \{\mathbf{q}'\}$ where $\mathbf{q}'$ is the Rabin σ-successor of $\mathbf{q}$.
- $\alpha = \langle G_0, B_0\rangle, \ldots, \langle G_{|Q|-1}, B_{2|Q|-1}\rangle$ where for a label $m \in \{0,\ldots,2|Q|\}$:
  - $G_m = \{\langle \mathbf{q},\mathbf{q}'\rangle \mid m$ Rabin-succeeds in $\langle \mathbf{q},\mathbf{q}'\rangle\}$.
  - $B_m = \{\langle \mathbf{q},\mathbf{q}'\rangle \mid m$ Rabin-dies in $\langle \mathbf{q},\mathbf{q}'\rangle\}$

Theorem B.2. *For an NBW $\mathcal{A}$, $L(D^T(\mathcal{A})) = L(\mathcal{A})$.*

Proof: For every word w, we show that the run $\mathbf{q}_0,\mathbf{q}_1,\ldots$ of $D^R(\mathcal{A})$ on w is accepting iff the run $\mathbf{q}^p_0,\mathbf{q}^p_1,\ldots$ of $D^P(\mathcal{A})$ on w is accepting. For convenience, let $\mathbf{q}_i = \langle S_i, \preceq_i, l_i, \lessdot_i\rangle$. We first note that for every i, it holds that $\mathbf{q}^p_i == \langle S_i, \preceq_i, l^p_i, \lessdot_i\rangle$: that that is to say $\mathbf{q}_i$ and $\mathbf{q}^p_i$ match on S_i, $\preceq_i$, and $\lessdot_i$. For S and $\preceq$, this is easy to see: the definitions of S' and $\preceq$' are identical in σ-successors and Rabin-σ-successors. For $\lessdot$, this follows from the fact that $\lessdot'$ is defined solely with respect to $\rho_{S,\preceq}$ and $\mathtt{unc}$, which do not change from σ-successors to Rabin-σ-successors. We pause to note that, for every i and $q \in S_i$, $q' \in S_{i+1}$, we have that $l_{i+1}(q') = l_i(q)$, iff $q \in \mathtt{unc}(q',\sigma_i)$, which holds iff both $l^p_{i+1}(q') = l^p_i(q)$ and $l^p_i(q) \notin\in \mathrm{FL}(l^p_i)$.

In one direction, assume there is a label m that dies in finitely many $\langle \mathbf{q}_i,\mathbf{q}_{i+1}\rangle$, and succeeds in infinitely many $\langle \mathbf{q}_i,\mathbf{q}_{i+1}\rangle$. We pause to note that, for every i and $q \in S_i$, $q' \in S_{i+1}$, we have that $l_{i+1}(q') = l_i(q)$, iff $q \in \mathtt{unc}(q',\sigma_i)$, which holds iff both $l^p_{i+1}(q') = l^p_i(q)$ and $l^p_i(q) \notin\in \mathrm{FL}(l^p_i)$. Let j be the first index so that m occurs in $\mathbf{q}_j$, but for every $k > j$, m does not die in $\langle \mathbf{q}_k,\mathbf{q}_{k+1}\rangle$. Let $q \in S_j$ be such that $l_j(q) = m$, and let $m' = l^p_j(q)$. For $k > j$, define $R_k = \{r \in S_k \mid l_k(r) = m\}$, and $R^p_k = \{r \in S_k \mid l^p_k(r) = m'\}$. Since m does not die in $\langle \mathbf{q}_k,\mathbf{q}_{k+1}\rangle$, R_k and R_{k+1} are both non-empty, and by our above observations m' does not Rabin-die in $\langle \mathbf{q}^p_k,\mathbf{q}^p_{k+1}\rangle$. Further, $R^p_k = R_k$, and $R^p_{k+1} = R_{k+1}$. This implies that if m succeeds in $\langle \mathbf{q}_k,\mathbf{q}_{k+1}\rangle$, then m' Rabin-succeeds $\langle \mathbf{q}^p_k,\mathbf{q}^p_{k+1}\rangle$. Thus m' Rabin-dies in finitely many $\langle \mathbf{q}^p_i,\mathbf{q}^p_{i+1}\rangle$, and Rabin-succeeds in infinitely many $\langle \mathbf{q}^p_i,\mathbf{q}^p_{i+1}\rangle$, and $D^T(\mathcal{A})$ accepts w.

In the other direction if $D^T(\mathcal{A})$ accepts w, this implies is a label m that Rabin-dies in finitely many $\langle \mathbf{q}^p_i,\mathbf{q}^p_{i+1}\rangle$, and Rabin-succeeds in infinitely many $\langle \mathbf{q}^p_i,\mathbf{q}^p_{i+1}\rangle$. Let j be the first index so that m occurs in $\mathbf{q}^p_j$, but for every $k > j$, m does not Rabin-die in $\langle \mathbf{q}^p_k,\mathbf{q}^p_{k+1}\rangle$. Let $q \in S_j$ be such that $l^p_j(q) = m$, and let $m' = l_j(q)$. For $k > j$, define $R^p_k = \{r \in S_k \mid l^p_k(r) = m\}$, and $R_k = \{r \in S_k \mid l_k(r) = m'\}$, Since m does not Rabin-die in $\langle \mathbf{q}^p_k,\mathbf{q}^p_{k+1}\rangle$, $m \notin F_{l^p_k}$ and R_k, R_{k+1} are both non-empty. By our above observations m' does not die in $\langle \mathbf{q}_k,\mathbf{q}_{k+1}\rangle$. Further, $R^p_k = R_k$, and $R^p_{k+1} = R_{k+1}$. This implies that if m Rabin-succeeds in $\langle \mathbf{q}^p_k,\mathbf{q}^p_{k+1}\rangle$, then m' succeeds $\langle \mathbf{q}_k,\mathbf{q}_{k+1}\rangle$. Thus m' dies in finitely many $\langle \mathbf{q}_i,\mathbf{q}_{i+1}\rangle$, and succeeds in infinitely many $\langle \mathbf{q}_i,\mathbf{q}_{i+1}\rangle$, and $D^R(\mathcal{A})$ accepts w. □

B.2 Parity Variant

The parity variation simply shifts labels down, instead of giving arbitrary free labels to new nodes. This means labels in the automaton are no longer consistent with with the labels l_i over T. To simplify this, we use an intermediate labeling that keeps labels consistent between two levels, but can use the labels $\{|Q|,\ldots 2|Q|\}$. Given a tight macrostate $\mathbf{q} \in \mathbf{Q}^t$ and $\sigma \in \Sigma$, define the *parity σ-successor of* $\mathbf{q}$ to be $\mathbf{q}' = \langle S', \preceq', l', \lessdot'\rangle$ where S', $\preceq'$, and $\lessdot'$ are defined as in Section 5, and l' is defined as follows:

(1) For $q \in S$, let $\mathtt{neph}(q,\sigma) = \mathtt{min}_{\preceq'}(\{r' \mid \text{exists } r \in S,\ q \lessdot r,\ r' \in \rho_{S,\preceq}(r,\sigma)\})$
(2) For $r' \in S'$, let $\mathtt{unc}(r',\sigma) = \mathtt{min}_{\preceq}(\{q \mid r' \in \mathtt{neph}(q,\sigma)\})$
(3) $\mathtt{mj}$ is the $\langle \preceq', <\rangle$-minjection from $\{r' \in S' \mid \mathtt{unc}(r',\sigma = \emptyset\}$ to $\{|Q|,\ldots,2|Q|\}$
(4) For $r' \in S'$, define the intermediate labeling

$$l^{int}(r') = \begin{cases} l(q),\ q \in \mathtt{unc}(r',\sigma) & \text{if } \mathtt{unc}(r',\sigma) \neq \emptyset, \\ \mathtt{mj}(r') & \text{if } \mathtt{unc}(r',\sigma) = \emptyset. \end{cases}$$

(5) For $r' \in S'$, define the final labeling $l'(r') = |\{l^{int}(q') \mid l^{int}(q') < l^{int}(r')\}|$

For $\sigma \in \Sigma$ and label $m \in \{0,\ldots,|Q|-1\}$, given a tight macrostate $\mathbf{q} = \langle S, \preceq, l, \lessdot \rangle \in \mathbf{Q}^t$ and its parity σ-successor $\mathbf{q}' = \langle S', \preceq', l', \lessdot' \rangle$ let l^{int} be the intermediate labeling defined above. Let $R = \langle r \in S \mid l(r) = m \rangle$ and $R' = \langle r' \in S' \mid l^{int}(r') = m \rangle$. Note that R' is defined with respect to the intermediate labeling. Say that a label m *parity-dies in* $\langle \mathbf{q}, \mathbf{q}' \rangle$ if $m \in R$, but $m \notin R'$. Say that m *parity-succeeds in* $\langle \mathbf{q}, \mathbf{q}' \rangle$ when $R \neq \emptyset$, $R' \neq \emptyset$, and either $R' \subseteq F$ or $\rho_{S,\preceq}(R,\sigma) \cap R' = \emptyset$. Define the priority function γ: $\mathbf{Q}^t \times \mathbf{Q}^t \to \{1,\ldots,2|Q|\}$ so that $\gamma(\langle \mathbf{q}\mathbf{q}' \rangle)$ is $\mathtt{min}(\{2m+2 \mid m \text{ parity-succeeds in } \langle \mathbf{q}, \mathbf{q}' \rangle\} \cup \{2m+1 \mid m \text{ parity-dies in } \langle \mathbf{q}, \mathbf{q}' \rangle\})$.

Definition B.3. Define the DPEW automata $D^P(\mathcal{A})$ to be $\langle \Sigma, \mathbf{Q}^t, \mathbf{Q}^{in}, \rho_{\mathbf{Q}}, \gamma \rangle$ where:

- $\mathbf{Q}^{in}$ is as defined Definition 5.7
- For $\mathbf{q} \in \mathbf{Q}$ and $\sigma \in \Sigma$, $\rho_{\mathbf{Q}}(\mathbf{q}, \sigma) = \{\mathbf{q}'\}$ where $\mathbf{q}'$ is the parity σ-successor of $\mathbf{q}$.

Theorem B.4. *For an NBW $\mathcal{A}$, $L(D^P(\mathcal{A})) = L(\mathcal{A})$.*

Proof: As above, for every word w, we show that the run $\mathbf{q}_0, \mathbf{q}_1, \ldots$ of $D^R(\mathcal{A})$ on w is accepting iff the run $\mathbf{q}_0^p, \mathbf{q}_1^p, \ldots$ of $D^P(\mathcal{A})$ on w is accepting. For convenience, let $\mathbf{q}_i = \langle S_i, \preceq_i, l_i, \lessdot_i \rangle$. Again, it holds that for every i $\mathbf{q}_i^p == \langle S_i, \preceq_i, l_i^p, \lessdot_i \rangle$: $\mathbf{q}_i$ and $\mathbf{q}_i^p$ match on S_i, $\preceq_i$, and $\lessdot_i$. For every i, let l_i^{int} be the intermediate labeling defined above. However, it is no longer that case that the labels of a branch will be consistent from $\mathbf{q}_i^p$ to $\mathbf{q}_{i+1}^p$. Instead, we must look for consistency in the intermediate labeling. for every i and $q \in S_i$, $q' \in S_{i+1}$, we have that $l_{i+1}(q') = l_i(q)$ iff $l_{i+1}^{int}(q') = l_i^p(q)$. If $l_{i+1}^p(q') \neq l_{i+1}^{int}(q')$, this implies there was a label $n < l_i^p(q)$ that occurs in the range of l_i^p, but not in the range of l_{i+1}^{int}.

In one direction, assume there is a label m that dies in finitely many $\langle \mathbf{q}_i, \mathbf{q}_{i+1} \rangle$, and succeeds in infinitely many $\langle \mathbf{q}_i, \mathbf{q}_{i+1} \rangle$. Let j be the first index so that m occurs in $\mathbf{q}_j$, but for every $k > j$, m does not die in $\langle \mathbf{q}_k, \mathbf{q}_{k+1} \rangle$. For every $j' > j$, let $q'_j \in \S_{j'}$ be such that $l_{j'}(q_{j'}) = m$. Note that the values of $l_{j'}^p(q_{j'})$ can only decrease: new labels are only introduced above $|Q|$, and $l_j^p(q_j) < |Q|$. Thus at some point the labels of $q_{j'}$ cease decreasing, and reach a stable point. Let k be this point, and let $m' = l_k^p(q_k)$. For a level $k' > k$, define $R_{k'} = \{r \in S_{k'} \mid l_{k'}(r) = m\}$, and $R_{k'}^p = \{r \in S_{k'} \mid l_{k'}^{int}(r) = m'\}$. Since the labels of $q_{k'}$ have stopped decreasing, we have that $R_{k'}^p = R_{k'}$. For every $k' > k$, it holds that m' does not parity-die in $\langle \mathbf{q}_{k'}^p, \mathbf{q}_{k'+1}^p \rangle$. Further, every label $n < m'$ must occur on every level $k' > k$: otherwise $l_{k'}^p(q_{k'})$ would not equal $l_{k'}^{int}(q_{k'})$. Thus for every $k' > k$, there is no label $n < m'$ that parity-dies in $\langle \mathbf{q}_{k'}^p, \mathbf{q}_{k'+1}^p \rangle$. Therefore $\gamma(\mathbf{q}_{k'}^p, \mathbf{q}_{k'+1}^p) \geq 2m'+1$. Now consider a level $k' > k$ where m succeeds in $\langle \mathbf{q}_{k'}, \mathbf{q}_{k'+1} \rangle$. By the note above, $R_{k'}^p = R_{k'}$, and m' parity-succeeds in $\langle \mathbf{q}_{k'}^p, \mathbf{q}_{k'+1}^p \rangle$, and $\gamma(\mathbf{q}_{k'}^p, \mathbf{q}_{k'+1}^p) = 2m'+2$. We have thus shown that the smallest priority occurring infinitely often in $2m'+2$, and thus w is accepted by $D^P(\mathcal{A})$.

In the other direction if $D^P(\mathcal{A})$ accepts w, this implies is a label m and level j so that for every $k > j$, it holds $\gamma(\mathbf{q}_k^p, \mathbf{q}_{k+1}^p) \geq 2m+2$, and for infinitely many $k > j$ it holds $\gamma(\mathbf{q}_k^p, \mathbf{q}_{k+1}^p) = 2m+2$. As noted above, this implies for every $k > j$ and $n \leq m$, n does not parity-die in $\langle \mathbf{q}_k^p, \mathbf{q}_{k+1}^p \rangle$, and for infinitely many $k > j$, m parity-succeeds in $\langle \mathbf{q}_k^p, \mathbf{q}_{k+1}^p \rangle$. Thus we conclude that for every $k > j$ and $q \in S_k$, $l_k^p(q) = m$ iff $l_k^{int}(q) = m$. Let $q \in S_j$ be such that $l_j^p(q) = m$, and let $m' = l_j(q)$. For every $k > j$, let $R_k^p = \{r \in S_k \mid l_k^p(r) = m\}$, and let $R_k = \{r \in S_k \mid l_k(r) = m'\}$. Again, we have that $R_k^p = R_k$, thus for every $k > j$, m' does not die in $\langle \mathbf{q}_k, \mathbf{q}_{k+1} \rangle$, and and for infinitely many $k > j$ we have m' succeeds in $\langle \mathbf{q}_k, \mathbf{q}_{k+1} \rangle$. Thus w is accepted by and $D^R(\mathcal{A})$. $\square$