Separating Regular Languages over Infinite Words with Respect to the Wagner Hierarchy

Christopher Hugenroth ✉
TU Ilmenau, Germany

Abstract

We investigate the separation problem for regular ω-languages with respect to the Wagner hierarchy where the input languages are given as deterministic Muller automata (DMA). We show that a minimal separating DMA can be computed in exponential time and that some languages require separators of exponential size. Further, we show that in this setting it can be decided in polynomial time whether a separator exists on a certain level of the Wagner hierarchy and that emptiness of the intersection of two languages given by DMAs can be decided in polynomial time. Finally, we show that separation can also be decided in polynomial time if the input languages are given as deterministic parity automata.

2012 ACM Subject Classification Theory of computation → Automata over infinite objects; Theory of computation → Regular languages

Keywords and phrases Separation, Regular, Wagner Hierarchy, Muller Automata, Parity Automata, Product Automata, Membership

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2021.46

Acknowledgements I want to thank Christof Löding who supervised my Master thesis on which this paper is based and Dietrich Kuske for clarifying discussions. Further, I want to thank the reviewers of former versions of this paper for their helpful suggestions.

1 Introduction

The membership problem for a class of languages asks to decide whether a given language is contained in this class. More generally, we can ask for a hierarchy of classes to decide membership for each of its classes. This problem has been studied for a variety of classes and hierarchies [11, 5, 6, 4, 12]. Solving the membership problem for a class helps us understand the expressive power of a class [8].

For example, the regular ω-languages are classified by the Wagner hierarchy [12, 7] which has been studied extensively. The hierarchy is infinite and refines both the Mostowski [7] and the Borel hierarchy with respect to the regular ω-languages [12]. The class of a language L is determined by the loop structure of any deterministic Muller automaton (DMA) that recognizes L, every DMA that recognizes L has the same structure. Using this, membership for the Wagner hierarchy can be decided efficiently if the language is given as a DMA [7].

Membership has also been studied with respect to the quantifier alternation hierarchy of first order logic for regular language of finite or infinite words [9], [8]. This hierarchy is infinite but the membership problem has only been solved for a few classes of the hierarchy. One approach towards solving membership for more classes uses the separation problem [8].

The Separation problem asks given two languages L_1, L_2 and a class C whether there is a language L in C that separates L_1 and L_2, i.e. $L_1 \subseteq L$ and $L_2 \cap L = \emptyset$. Separation is more general than membership because a language L is in C iff L and its complement can be separated by a language in C, so membership can be reduced to separation. Solving the separation problem for a class C requires understanding the discriminating power of C and therefore an even deeper understanding of a class than for solving membership [8].

41st IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2021).
Editors: Mikołaj Bojańczyk and Chandra Chekuri; Article No. 46; pp. 46:1–46:13
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

In this paper, we study the separation problem for the Wagner hierarchy to gain a deeper understanding of this hierarchy. Further, we might get some insights on how to solve the membership problem for the Mostowski hierarchy for infinite trees [5, 6] and the aforementioned quantifier alternation hierarchy which are similar to the Wagner hierarchy.

We assume that the input languages for the separation problem are given as DMAs. First, we show that the loop structure of every separating DMA is similar. Using this, a DMA whose language separates the two input languages and which is minimal with respect to the Wagner hierarchy can be computed in exponential time. Next, we show that this result is optimal in the sense that there are infinitely many pairs of DMAs for which every separating DMA is exponentially larger than the DMAs in the pair.

Surprisingly, *deciding* whether a separator exists is possible in polynomial time. This can be done by analyzing the loop structure of a special product automaton. This also works if the input languages are given as deterministic parity automata (DPA).

We can also use the separation algorithm to decide in polynomial time whether the languages of two DMAs (DPAs) are disjoint. Meanwhile, the intersection of DMAs (DPAs) has exponential size in general [1], so our algorithm has better complexity than a naive algorithm for disjointness and, to the best of our knowledge, is the first to do so.

2 Preliminaries

We denote the *natural numbers* by $\mathbb{N} = \{0, 1, 2, \dots\}$. The strict linear order on the natural numbers is denoted by $<$. All numbers in this paper are natural, we do not state that for every number explicitly. The projection function to the i-th component is denoted by pr_i.

An *alphabet* Σ is a finite, non-empty set of symbols. A *finite word* over Σ is a mapping from $\{0, 1, \dots, k-1\}$ to Σ for some $k \in \mathbb{N}$. Here, k is the *length* of w. The *empty word* ε is the unique word of length 0. The *class of all finite words* over Σ is Σ^*. An *infinite word* α over Σ is a mapping from $\mathbb{N}$ to Σ. The *class of all infinite* words over Σ is denoted by Σ^ω.

For $w \in \Sigma^*$ and $i, j \in \mathbb{N}$ with $i, j < |w|$ we define $w[i, j]$ as $w[i, j] = w(i) \dots w(j)$ for $i \leq j$ and $w[i, j] = \varepsilon$ for $i > j$. For infinite words the definition is analogous.

A symbol $a \in \Sigma$ *occurs infinitely often* in a word $\alpha \in \Sigma^\omega$ if there are infinitely many $i \in \mathbb{N}$ with $\alpha(i) = a$. For an infinite word $\alpha \in \Sigma^\omega$ let $\mathrm{Inf}(\alpha)$ be the *set of letters that occur infinitely often* in α.

2.1 Automaton Structures

An *automaton structure* is a tuple $A = (Q, \Sigma, \delta, q_0)$ where Q is a finite, non-empty set of states, Σ is an alphabet, $\delta : Q \times \Sigma \to Q$ is a transition function and $q_0 \in Q$ is the initial state. All automata considered in this paper are deterministic.

The *run* of an automaton structure $A = (Q, \Sigma, \delta, q_0)$ on a finite word $w \in \Sigma^*$ from $p \in Q$ is the word $\rho \in Q^*$ with $\rho(0) = p$ and $\rho(i+1) = \delta(\rho(i), w(i))$ for all $i \in \mathbb{N}$ with $i < |w|$. The *run* of an automaton structure A on an infinite word $\alpha \in \Sigma^\omega$ from a state $p \in Q$ is an infinite word $\rho \in Q^\omega$ with $\rho(0) = p$ and $\rho(i+1) = \delta(\rho(i), \alpha(i))$ for all $i \in \mathbb{N}$. We denote such a run by $\rho_A(p, w)$ for $w \in \Sigma^*$, respectively $\rho_A(p, \alpha)$ for $\alpha \in \Sigma^\omega$.

We write $p \xrightarrow[P]{w} q$ if there is a run of A on w from p to q and the set of states visited on this run is exactly P, i.e. $P = \{p' \in Q \mid \text{ there is } 0 \leq i \leq |w| \text{ such that } \rho_A(p, w)(i) = p'\}$. We write $p \xrightarrow{w} q$ if there is some P such that $p \xrightarrow[P]{w} q$. With $p \to q$ we denote that there is a word $w \in \Sigma^*$ with $p \xrightarrow{w} q$. We assume that all states are reachable from the initial state q_0 for every automaton structure A, i.e. $q_0 \to q$ for all $q \in Q$.

A *loop* of an automaton structure $A = (Q, \Sigma, \delta, q_0)$ is a non-empty, strongly connected set of states. So, $P \subseteq Q$ is a loop if there is a state $p \in P$ and a finite word $v \in \Sigma^*$ with $v \neq \varepsilon$ and $p \xrightarrow[P]{v} p$. Equivalently, P is a loop if there is an infinite word $\alpha \in \Sigma^\omega$ such that the infinite run ρ of A on α from q_0 satisfies $\mathrm{Inf}(\rho) = P$. A loop P of A is a *strongly connected component* (SCC) if there is no loop P' of A with $P' \supsetneq P$.

Notice that for every loop P of A there is exactly one SCC S of A with $P \subseteq S$. If there are two SCCs S, S' with $P \subseteq S, S'$ we have $S \cap S' \neq \emptyset$. So, $S = S'$ since the union of non-disjoint loops is a loop again.

2.2 Deterministic Muller Automata

We denote the powerset of a set Q by $2^Q = \{P \mid P \subseteq Q\}$.

A *deterministic Muller automaton* (DMA) is a tuple $\mathcal{A} = (Q, \Sigma, \delta, q_0, \mathcal{F})$ where $A = (Q, \Sigma, \delta, q_0)$ is an automaton structure and $\mathcal{F} \subseteq 2^Q$ is an *acceptance condition*, we also write $\mathcal{A} = (A, \mathcal{F})$.

When we talk about a run or a loop of a DMA $\mathcal{A}$ we mean a run or a loop of its automaton structure. A loop P is accepting in $\mathcal{A}$ if $P \in \mathcal{F}$ and rejecting otherwise. A DMA $\mathcal{A} = (Q, \Sigma, \delta, q_0, \mathcal{F})$ *accepts* an infinite word $\alpha \in \Sigma^\omega$ if the run $\rho_{\mathcal{A}}(q_0, \alpha)$ satisfies $\mathrm{Inf}(\rho_{\mathcal{A}}(q_0, \alpha)) \in \mathcal{F}$, i.e. the set of states visited infinitely often is an accepting loop. The language *accepted* by $\mathcal{A}$ is the set of words $L(\mathcal{A})$ accepted by $\mathcal{A}$.

The size $|\mathcal{A}|$ of a DMA $\mathcal{A} = (Q, \Sigma, \delta, q_0, \mathcal{F})$ is $|Q| + |\mathcal{F}|$. Notice that $|\mathcal{F}|$ might be exponentially larger than $|Q|$. For a DMA $(A, \mathcal{F})$ we can compute a DMA $(A, \mathcal{F}')$ such that $L(A, \mathcal{F}) = L(A, \mathcal{F}')$ and $\mathcal{F}'$ contains only loops in polynomial time with respect to $|(A, \mathcal{F})|$.

An ω-language L is regular iff there is a DMA $\mathcal{A}$ with $L(\mathcal{A}) = L$.

3 Wagner Hierarchy

Wagner defined chains and superchains for deterministic Muller automata. He proved that if L is an ω-regular language then every DMA that recognizes L has the same maximal superchain length [12]. So, the maximal length of superchains is an invariant of the language. Hence, the regular ω-languages can be classified according to this invariant. This classification forms the Wagner hierarchy.

3.1 Chains

Let $\mathcal{A} = (Q, \Sigma, \delta, q_0, \mathcal{F})$ be a DMA. For $m \geq 1$ an *m-chain* of $\mathcal{A}$ is a sequence of loops $c = (P_1, \ldots, P_m)$ such that $P_i \subseteq P_{i+1}$ and $P_i \in \mathcal{F}$ iff $P_{i+1} \notin \mathcal{F}$ for all $0 < i < m$. We also speak of a chain c if it is clear from context that c is an m-chain for a certain $m \in \mathbb{N}$. A chain $c = (P_1, \ldots, P_m)$ is *positive* if P_1 is accepting and *negative* if P_1 is rejecting.

An m'-chain $c' = (P'_1, \ldots, P'_{m'})$ is *reachable* from an m-chain $c = (P_1, \ldots, P_m)$ if there are states $p \in P_1$, $p' \in P'_1$ with $p \to p'$. We denote this by $c \to c'$. Notice that this is equivalent to saying that every (or some) state of $P'_{m'}$ can be reached from every (or some) state of P_m.

3.2 Superchains

A superchain is a reachability-ordered sequence of chains which alternate between positive and negative chains. Formally, an (m, n)-*superchain* is a sequence $s = (c_1, \ldots, c_n)$ of m-chains such that c_i is positive iff c_{i+1} is negative and further $c_i \to c_{i+1}$ for all $0 < i < n$. An (m, n)-superchain $s = (c_1, \ldots, c_n)$ is *positive* if c_1 is positive and *negative* otherwise.

▶ Remark 1. Let $\mathcal{A} = (Q, \Sigma, \delta, q_0, \mathcal{F})$ be a DMA and $m > 1$. The following are equivalent:

1. $\mathcal{A}$ has an m-chain.
2. $\mathcal{A}$ has an $(m-1, n)$-superchain for all $n \in \mathbb{N}$.
3. $\mathcal{A}$ has an $(m-1, |Q|+1)$-superchain.

Let $\leq$ be the lexicographic order on $\mathbb{N}^2$. An (m', n')-superchain is *longer* than an (m, n)-superchain if $m' > m$ or if $m' = m$ and $n' > n$, i.e. $(m', n') > (m, n)$. This order prioritizes m over n which ensures that for each DMA there is a maximal length of superchains in this DMA. If we prioritize n this is not the case as one can see with Remark 1.

3.3 Wagner hierarchy

Wagner showed that the superchains of a DMA $\mathcal{A}$ are an invariant of the language $L(\mathcal{A})$ [12] in the following sense: For $m, n \in \mathbb{N}$ and two DMAs $\mathcal{A}_1$, $\mathcal{A}_2$ with $L(\mathcal{A}_1) = L(\mathcal{A}_2)$ it holds that $\mathcal{A}_1$ has an (m, n)-superchain if and only if $\mathcal{A}_2$ has an (m, n)-superchain. So, the existence of a superchain in $\mathcal{A}$ is completely determined by $L(\mathcal{A})$.

The *Wagner hierarchy* classifies the regular ω-languages based on superchains. We consider downward-closed classes that Wagner defined. For all $m, n \in \mathbb{N}$ there are three classes C_m^n, D_m^n, E_m^n which are defined as follows:

$$\begin{aligned} E_m^n &= \{L(\mathcal{A}) \mid \text{If } \mathcal{A} \text{ has an } (k,\ell)\text{-superchain then } (k,\ell) \leq (m,n)\} \\ D_m^n &= \{L(\mathcal{A}) \mid \text{If } \mathcal{A} \text{ has an } (k,\ell)\text{-superchain then } (k,\ell) \leq (m,n) \\ &\qquad \text{and all } (m,n)\text{-superchains in } \mathcal{A} \text{ are positive}\} \\ C_m^n &= \{L(\mathcal{A}) \mid \text{If } \mathcal{A} \text{ has an } (k,\ell)\text{-superchain then } (k,\ell) \leq (m,n) \\ &\qquad \text{and all } (m,n)\text{-superchains in } \mathcal{A} \text{ are negative}\} \end{aligned}$$

It holds $C_m^n, D_m^n \subseteq E_m^n$ and E_m^n is closed under complement.

4 Wagner Separation

Let X_m^n be a Wagner class. We say that two languages $L_1, L_2 \subseteq \Sigma^\omega$ are X_m^n*-separable* if there is a language $L \in X_m^n$ with $L_1 \subseteq L$ and $L_2 \cap L = \emptyset$. Notice that L_1, L_2 are E_m^n-separable iff L_2, L_1 are E_m^n-separable. Further, L_1, L_2 are C_m^n-separable iff L_2, L_1 are D_m^n-separable, if L separates L_1, L_2 then $\Sigma^\omega \setminus L$ separates L_2, L_1. We investigate the following problem:

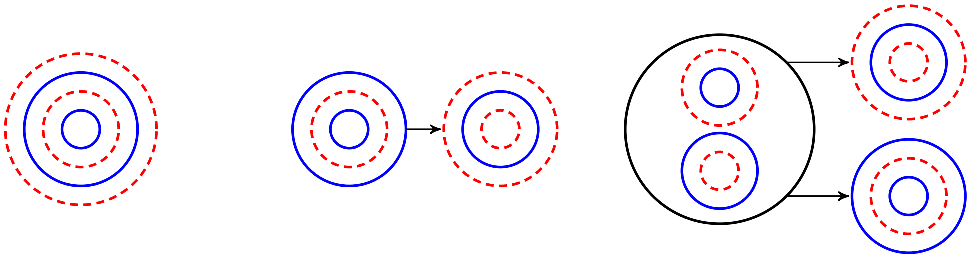
WAGNERSEPARATION
Given: Two DMAs $\mathcal{A}_1, \mathcal{A}_2$, $X \in \{C, D, E\}$ and $m, n \in \mathbb{N}$.
Decide: Are the languages $L(\mathcal{A}_1)$ and $L(\mathcal{A}_2)$ X_m^n-separable?

In the following we define chains for DMAs that can recognize two languages and show how to construct a separating DMA that is minimal with respect to the Wagner hierarchy.

4.1 Generalization of Chains

Consider a DMA $\mathcal{A} = (Q, \Sigma, \delta, q_0, \mathcal{F})$. Wagner defined chains in $\mathcal{A}$ as sequences of loops that alternate between $\mathcal{F}$ and its complement. It turns out that a slight generalization of this concept is helpful for this paper. Let $\mathcal{F}_1, \mathcal{F}_2$ be two subsets of 2^Q. We say that $c = (P_1, \ldots, P_m)$ is an *m-chain with respect to* $(\mathcal{F}_1, \mathcal{F}_2)$ if

1. P_i is a loop for $1 \leq i \leq m$,
2. $P_i \subseteq P_{i+1}$ for $1 \leq i < m$,
3. $P_i \in \mathcal{F}_1 \cup \mathcal{F}_2$ for $1 \leq i \leq m$,
4. $P_i \in \mathcal{F}_1$ iff $P_{i+1} \in \mathcal{F}_2$ for $1 \leq i < m$.



(a) Blue-start, red-end 4-chain. **(b)** Blue-start $(3,2)$-superchain. **(c)** A $(3,2)$-forcing SCC.

Figure 1 Illustration of patterns in A. Each circle represents a loop in A. Notice that no matter how the black loop in (c) is colored there will be a $(3,2)$-superchain in A.

An m-chain $c = (P_1, \dots, P_m)$ with respect to $(\mathcal{F}_1, \mathcal{F}_2)$ is $\mathcal{F}_1$-start if $P_1 \in \mathcal{F}_1$ and otherwise it is $\mathcal{F}_2$-start. Notice that c is an m-chain in $\mathcal{A}$ (in the sense of Wagner) if and only if c is an m-chain with respect to $(\mathcal{F}, 2^Q \setminus \mathcal{F})$. Further, notice that there are arbitrarily long chains with respect to $(\mathcal{F}_1, \mathcal{F}_2)$ whenever these sets are not disjoint.

An *(m,n)-superchain with respect to* $(\mathcal{F}_1, \mathcal{F}_2)$ is a sequence $s = (c_1, \dots, c_n)$ such that each c_j is an m-chain with respect to $(\mathcal{F}_1, \mathcal{F}_2)$, c_{j+1} is reachable from c_j and c_j is $\mathcal{F}_1$-start iff c_{j+1} is $\mathcal{F}_2$-start for all $1 \leq j < n$. The superchain $s = (c_1, \dots, c_n)$ is $\mathcal{F}_1$-start if c_1 is $\mathcal{F}_1$-start and otherwise it is $\mathcal{F}_2$-start.

4.2 Blue and Red Chains

Let $L_B, L_R \subseteq \Sigma^\omega$ be two regular ω-languages. An automaton structure $A = (Q, \Sigma, \delta, q_0)$ *can accept* both languages L_B and L_R if there are acceptance conditions $\mathcal{F}_B$, $\mathcal{F}_R \subseteq 2^Q$ such that $L(A, \mathcal{F}_B) = L_B$ and $L(A, \mathcal{F}_R) = L_R$. For example, the product automaton structure of two automata can accept both their languages.

For the remainder of the section fix two languages L_B, L_R and an automaton structure $A = (Q, \Sigma, \delta, q_0)$ that can accept both L_B and L_R. There are unique sets of loops $\mathcal{F}_B, \mathcal{F}_R$ such that $L(A, \mathcal{F}_B) = L_B$ and $L(A, \mathcal{F}_R) = L_R$. We refer to $\mathcal{F}_B$ as the set of blue loops and to $\mathcal{F}_R$ as the set of red loops.

We study chains in A with respect to $(\mathcal{F}_B, \mathcal{F}_R)$ and their connection to chains in DMAs whose language separate L_B and L_R. In Section 4.4 we show that if there is an (m,n)-superchain in A with respect to $(\mathcal{F}_B, \mathcal{F}_R)$ then there is an (m,n)-superchain in every DMA that separates L_B and L_R. Additionally, there are loops in A that are neither blue nor red but nevertheless important for the existence of superchains in a separating DMA.

We say that an m-chain $c = (P_1, \dots, P_m)$ is *blue-end* (*red-end*) if $P_m \in \mathcal{F}_B$ ($P_m \in \mathcal{F}_R$). An SCC S in A is blue-end (red-end) if all of the longest chains in S are blue-end (red-end). An SCC S is (m,n)-*forcing* for A if all of the following conditions hold:

1. A has a blue-end $(m-1)$-chain $c_B = (P_1^B, \dots, P_{m-1}^B)$ and a red-end $(m-1)$-chain $c_R = (P_1^R, \dots, P_{m-1}^R)$.
2. These chains are contained in S, i.e. $P_{m-1}^B \subseteq S$ and $P_{m-1}^R \subseteq S$.
3. A has an $(m, n-1)$-superchain s_B whose first chain is blue-end and an $(m, n-1)$-superchain s_R whose first chain is red-end.
4. These superchains can be reached from S.

See Figure 1 for an illustration. Notice that if A has an (m,n)-superchain with respect to $(\mathcal{F}_B, \mathcal{F}_R)$ then the SCC that contains its first chain is (m,n)-forcing. We say that A is at most (m,n)-forcing if for every SCC in A that is (k,ℓ)-forcing we have $(k,\ell) \leq (m,n)$.

4.3 Separator

In the following we show that we can define two DMAs that separate L_B and L_R (if possible) and show later that one of them is minimal with respect to the Wagner hierarchy. To construct these DMAs we use the automaton structure A. If there is a loop in A that is blue and red then $L_B \cap L_R \neq \emptyset$, so the languages cannot be separated. Otherwise, $L(A, \mathcal{F}_B)$ separates L_B and L_R.

We give an intuitive description of the separator before defining it formally. We define an automaton structure A_{sep} that consists of three copies of A such that each loop of A is in one of the copies. The languages L_B and L_R can be accepted by A_{sep} using the sets $\mathcal{F}_B^0$, $\mathcal{F}_R^0$ of loops whose projection to A is in $\mathcal{F}_B$, $\mathcal{F}_R$ respectively.

Each SCC is then added to $\mathcal{F}_B^0$ or $\mathcal{F}_R^0$, this yields sets $\mathcal{F}_B^1$ and $\mathcal{F}_R^1$. For example, a blue-end SCC is in $\mathcal{F}_B^1$. If A is at most (m, n)-forcing then each superchain in A_{sep} with respect to $(\mathcal{F}_B^1, \mathcal{F}_R^1)$ has at most length (m, n).

Then, it remains to fix the acceptance of the loops that are not an SCC and have no color (neither blue nor red). We fix the acceptance of these remaining loops based on the longest chains with respect to $(\mathcal{F}_B^1, \mathcal{F}_R^1)$. This yields the sets $\mathcal{F}_B^2, \mathcal{F}_R^2$, analogously we construct $\mathcal{G}_B^1, \mathcal{G}_R^1, \mathcal{G}_B^2, \mathcal{G}_R^2$. The DMA whose language separates L_B and L_R and which are minimal with respect to the Wagner hierarchy are $(A_{\text{sep}}, \mathcal{F}_B^2)$ and $(A_{\text{sep}}, \mathcal{G}_B^2)$.

Separator Construction

Recall that $A = (Q, \Sigma, \delta, q_0)$ is an automaton structure that can accept both L_B and L_R. Let $m, n \in \mathbb{N}$ such that A is at most (m, n)-forcing.

We construct an automaton structure $A_{\text{sep}} = (Q_{\text{sep}}, \Sigma, \delta_{\text{sep}}, q_{\text{sep}})$. The states $Q_{\text{sep}} = Q \times \{N, B, R\}$ consist of three copies of the states of A. The initial state $q_{\text{sep}} = (q_0, N)$ is in the N-copy. These copies are used to remember in a run whether an m-chain has been seen and if so it also remembers whether the most recent m-chain was blue-end or red-end. For $p \in Q$, $a \in \Sigma$, $q = \delta(p, a)$ and $C \in \{N, B, R\}$ let

$$\delta_{\text{sep}}((p, C), a) = \begin{cases} (q, B) & \text{, if there is an SCC } S \text{ in } A \text{ with } q \in S \\ & \text{ and there is a blue-end } m\text{-chain in } S \\ (q, R) & \text{, if there is an SCC } S \text{ in } A \text{ with } q \in S \\ & \text{ and there is a red-end } m\text{-chain in } S \\ (q, C) & \text{, otherwise} \end{cases}$$

The transition function δ_{sep} is well-defined because in no SCC in A there are both a blue- and a red-end m-chain since A is at most (m, n)-forcing. Notice that if a run in A_{sep} enters a new copy then the corresponding run in A has to leave an SCC because the m-chains in the SCC changed. So, every loop P in A_{sep} is either in $Q \times \{B\}$, in $Q \times \{R\}$ or in $Q \times \{N\}$.

We define sets $\mathcal{F}_B^1$ and $\mathcal{F}_R^1$ that cover every SCC and every set whose projection is colored. Let $P \subseteq Q_{\text{sep}}$ be a set. Then, $P \in \mathcal{F}_B^1$ if one of the following is true:

1. $\text{pr}_1(P)$ is blue in A
2. $\text{pr}_1(P)$ is colorless, $P \subseteq Q \times \{B\}$, P is an SCC
3. $\text{pr}_1(P)$ is colorless, $P \subseteq Q \times \{N\}$, P is an SCC, $\text{pr}_1(P)$ is a blue-end SCC
4. $\text{pr}_1(P)$ is colorless, $P \subseteq Q \times \{N\}$, P is an SCC and there is no (m, n)-superchain reachable from $\text{pr}_1(P)$ whose first chain is red-end

Similarly, $P \in \mathcal{F}_R^1$ if one of the following is true:

1. $\mathrm{pr}_1(P)$ is red in A
2. $\mathrm{pr}_1(P)$ is colorless, $P \subseteq Q \times \{R\}$, P is an SCC
3. $\mathrm{pr}_1(P)$ is colorless, $P \subseteq Q \times \{N\}$, P is an SCC, there is an (m,n)-superchain reachable from $\mathrm{pr}_1(P)$ whose first chain is red-end and $\mathrm{pr}_1(P)$ is not a blue-end SCC

▶ **Lemma 2.** *The longest superchains in A_{sep} w.r.t. $(\mathcal{F}_B^1, \mathcal{F}_R^1)$ have at most length (m,n).*

The sets $\mathcal{F}_B^1$ and $\mathcal{F}_R^1$ are defined to minimize the chain length but there are other sets that also do this. We define two such sets $\mathcal{G}_B^1$ and $\mathcal{G}_R^1$ that only differ in the third and fourth condition from $\mathcal{F}_B^1$ and $\mathcal{F}_R^1$. A loop P is in $\mathcal{G}_B^1$ if one of the following is true

1. $\mathrm{pr}_1(P)$ is blue in A
2. $\mathrm{pr}_1(P)$ is colorless, $P \subseteq Q \times \{B\}$, P is an SCC
3. $\mathrm{pr}_1(P)$ is colorless, $P \subseteq Q \times \{N\}$, P is an SCC, there is an (m,n)-superchain reachable from $\mathrm{pr}_1(P)$ whose first chain is blue-end and $\mathrm{pr}_1(P)$ is not a red-end SCC

Similarly, $P \in \mathcal{G}_R^1$ if one of the following is true

1. $\mathrm{pr}_1(P)$ is red in A
2. $\mathrm{pr}_1(P)$ is colorless, $P \subseteq Q \times \{R\}$, P is an SCC
3. $\mathrm{pr}_1(P)$ is colorless, $P \subseteq Q \times \{N\}$, P is an SCC, $\mathrm{pr}_1(P)$ is a red-end SCC
4. $\mathrm{pr}_1(P)$ is colorless, $P \subseteq Q \times \{N\}$, P is an SCC and there is no (m,n)-superchain reachable from $\mathrm{pr}_1(P)$ whose first chain is blue-end

▶ **Lemma 3.** *The longest superchains in A_{sep} w.r.t. $(\mathcal{G}_B^1, \mathcal{G}_R^1)$ have at most length (m,n).*

To obtain a separator that is minimal with respect to the Wagner hierarchy it is not only important how long superchains are but also how they start.

▶ **Lemma 4.** *If there are no blue-start (red-start) (m,n)-superchains in A then there are no $\mathcal{F}_B^1$-start ($\mathcal{F}_R^1$-start) (m,n)-superchains in A_{sep} with respect to $(\mathcal{F}_B^1, \mathcal{F}_R^1)$ or there are no $\mathcal{G}_B^1$-start ($\mathcal{G}_R^1$-start) (m,n)-superchains in A_{sep} with respect to $(\mathcal{G}_B^1, \mathcal{G}_R^1)$.*

We define the acceptance for every loop P that is not in $\mathcal{F}_B^1$ or $\mathcal{F}_R^1$ based on the longest chains with respect to $(\mathcal{F}_B^1, \mathcal{F}_R^1)$ that start with a strict superloop of P. Let P be a loop in A_{sep} such that there is an SCC S with $P \subsetneq S$. We define $P^{\subsetneq} = \{P' \mid P \subsetneq P' \subseteq S\}$. For P consider the chains with respect to $(\mathcal{F}_B^1 \cap P^{\subsetneq}, \mathcal{F}_R^1 \cap P^{\subsetneq})$. A loop $P \subseteq Q_{\mathrm{sep}}$ is in $\mathcal{F}_B^2$ if $P \in \mathcal{F}_B^1$ or if $P \notin \mathcal{F}_R^1$ and the longest chains with respect to $(\mathcal{F}_B^1 \cap P^{\subsetneq}, \mathcal{F}_R^1 \cap P^{\subsetneq})$ are all $(\mathcal{F}_B^1 \cap P^{\subsetneq})$-start. So, a set $P \subseteq Q_{\mathrm{sep}}$ is in its complement $\mathcal{F}_R^2 = 2^Q \setminus \mathcal{F}_B^2$ if $P \notin \mathcal{F}_B^1$ and $P \in \mathcal{F}_R^1$ or if $P \notin \mathcal{F}_B^1$ and the longest chains with respect to $(\mathcal{F}_B^1 \cap P^{\subsetneq}, \mathcal{F}_R^1 \cap P^{\subsetneq})$ are all $(\mathcal{F}_R^1 \cap P^{\subsetneq})$-start. We define $\mathcal{G}_B^2$ and $\mathcal{G}_B^R$ analogously.

We show that $\mathcal{F}_B^2$ is well-defined, the proof for $\mathcal{G}_B^2$ is analogous. The longest chains with respect to $(\mathcal{F}_B^1 \cap P^{\subsetneq}, \mathcal{F}_R^1 \cap P^{\subsetneq})$ are all $(\mathcal{F}_B^1 \cap P^{\subsetneq})$-end or all $(\mathcal{F}_R^1 \cap P^{\subsetneq})$-end because S is in $\mathcal{F}_B^1 \cap P^{\subsetneq}$ or in $\mathcal{F}_R^1 \cap P^{\subsetneq}$. So, the longest chains are also all $(\mathcal{F}_B^1 \cap P^{\subsetneq})$-start or all $(\mathcal{F}_R^1 \cap P^{\subsetneq})$-start. Further, there is always a chain in $P^{\subsetneq}$ because (S) is a chain.

▶ **Lemma 5.** *If there is a $\mathcal{F}_B^2$-start ($\mathcal{F}_R^2$-start) (k,ℓ)-superchain in A with respect to $(\mathcal{F}_B^2, \mathcal{F}_R^2)$ then there is a $\mathcal{F}_B^1$-start ($\mathcal{F}_R^1$-start) (k,ℓ)-superchain in A with respect to $(\mathcal{F}_B^1, \mathcal{F}_R^1)$.*

The analogous statement holds for $(\mathcal{G}_B^1, \mathcal{G}_R^1)$ and $(\mathcal{G}_B^2, \mathcal{G}_R^2)$. Finally, we can define the DMA $(A_{\mathrm{sep}}, \mathcal{F}_B^2)$ and $(A_{\mathrm{sep}}, \mathcal{G}_B^2)$ whose languages separate L_B and L_R if possible.

▶ **Lemma 6.** *If $L_B \cap L_R = \emptyset$ then $L(A_{sep}, \mathcal{F}_B^2)$, $L(A_{sep}, \mathcal{G}_B^2)$ separate L_B and L_R.*

4.4 Translating Chains

We show that the separator constructed in the previous section is minimal with respect to the Wagner hierarchy. Let $\mathcal{A}_B, \mathcal{A}_R, \mathcal{A}_{\text{sep}}$ be DMAs such that $L(\mathcal{A}_{\text{sep}})$ separates $L(\mathcal{A}_B)$ and $L(\mathcal{A}_R)$. Further, let A be an automaton structure that can accept both $L(\mathcal{A}_B)$ and $L(\mathcal{A}_R)$.

▶ **Lemma 7.** *If A has an (m,n)-forcing SCC then $\mathcal{A}_{sep}$ has an (m,n)-superchain.*

To prove Lemma 7 we consider the loops that appear in the chains and in the superchains of the (m,n)-forcing SCC. We find finite words that run repeatedly through these loops. These words have runs in any separating DMA $\mathcal{A}_{\text{sep}}$. These runs yield loops and ultimately superchains in $\mathcal{A}_{\text{sep}}$. An analysis of this proof yields the following result:

▶ **Lemma 8.** *If A has a blue-start (red-start) (m,n)-superchain then $\mathcal{A}_{sep}$ has a positive (negative) (m,n)-superchain.*

Combining all previous results yields the main result of this paper.

▶ **Theorem 9.** *Let $m,n \in \mathbb{N}$ and $L_B, L_R \subseteq \Sigma^\omega$. Let A be an automaton structure that can accept both L_B and L_R. The languages L_B and L_R are*
1. *E_m^n-separable iff A is at most (m,n)-forcing,*
2. *D_m^n-separable iff A is at most (m,n)-forcing and every (m,n)-superchain in A is blue-start,*
3. *C_m^n-separable iff A is at most (m,n)-forcing and every (m,n)-superchain in A is red-start.*

▶ **Corollary 10.** *L_B, L_R are C_m^n- and D_m^n-separable iff there are no (m,n)-superchains in A.*

5 Solving Separation

We use Theorem 9 to show how WagnerSeparation and related questions can be resolved.

5.1 Computing a Separator

We show that DMAs as defined in Section 4.3 can be computed in exponential time. For this we use product automata. Let A_1, A_2 be two automaton structures with $A_i = (Q_i, \Sigma, \delta_i, q_i)$ for $i \in \{1,2\}$. The *product automaton* of A_1 and A_2 is the automaton structure $A_1 \times A_2 = (Q_1 \times Q_2, \Sigma, \delta, (q_1, q_2))$, where δ applies the transition functions component wise, i.e. for $(p_1, p_2) \in Q_1 \times Q_2$ and $a \in \Sigma$ we have $\delta((p_1,p_2),a) = (\delta_1(p_1,a), \delta_2(p_2,a))$. The product automaton of two DMAs is defined as the product automaton of their automaton structures.

▶ **Theorem 11.** *Let $\mathcal{A}_1$, $\mathcal{A}_2$ with $\mathcal{A}_i = (Q_i, \Sigma, \delta_i, q_i, \mathcal{F}_i)$ be two DMA. A separating DMA that is minimal with respect to the Wagner hierarchy can be constructed in exponential time.*

The product automaton structure $A_1 \times A_2$ can be computed in polynomial time and it can accept both $L(\mathcal{A}_1)$ and $L(\mathcal{A}_2)$. The set of states of the automaton structure A_{sep} consists of three copies of $Q_1 \times Q_2$ which clearly can be computed in polynomial time. The transition function δ_{sep} and the acceptance conditions $\mathcal{F}_B^2$, $\mathcal{G}_B^2$ are defined based on the chain structure in the product automaton with respect to blue and red loops.

The chain structure with respect to the red and blue loops can be computed as follows. Iterate over all subsets of $Q_1 \times Q_2$ and color a subset blue if its projection to the first component is in $\mathcal{F}_1$. Color a subset red if its projection to the second component is in $\mathcal{F}_2$.

Next, construct a graph that has the colored loops as nodes. Introduce an edge between two loops if they have different colors and the first loop is a subset of the second. Then the paths of maximal length in this graph correspond to chains of maximal lengths. To compute the superchains of maximal length a new graph can be constructed with chains of maximal length as nodes. Further, there is an edge between two chains if the second chain is reachable from the first chain and they start with different colors. The superchains of maximal length then correspond to paths of maximal length in this graph. The constructed graphs are acyclic.

These graphs are of exponential size in $|\mathcal{A}_1|+|\mathcal{A}_2|$ or smaller. Using for example Dijkstra's algorithm, the paths of maximal length in an acyclic graph can be computed in polynomial time. Thus, the chain structure of A and therefore $(A_{\text{sep}}, \mathcal{F}_B^2)$, $(A_{\text{sep}}, \mathcal{G}_B^2)$ can be computed in exponential time.

5.2 Exponential blowup of Separators

We show that in some cases the size of every separator that is minimal with respect to the Wagner hierarchy is exponentially larger than the size of the input DMAs. So, if one wants to compute a complete separator (not just deciding the value of a certain bit) then exponential time is the best complexity one can hope for.

▶ **Theorem 12.** *For all $1 \leq m \leq k \in \mathbb{N}$ there are DMAs $\mathcal{A}_1, \mathcal{A}_2$ of size at most $2k$ such that every DMA $\mathcal{A}$ with no $(m+1)$-chain whose language separates $L(\mathcal{A}_1)$ and $L(\mathcal{A}_2)$ has size at least $2^{k-m} - k - 1$ and there is such a DMA.*

Proof. Let $\Sigma = \{1, \dots, k\}$. We define the DMA $\mathcal{A}_1$, $\mathcal{A}_2$ that have the same automaton structure A. Let $A = (Q, \Sigma, \delta, 1)$ with $Q = \{1, \dots, k\} \cup \{\bot\}$ and δ as follows: For a state $i \in Q$ and $j \in \Sigma$ let $\delta(i, j) = j$ if $i \neq j$ and $\delta(i, j) = \bot$ if $i = j$. Further, $\delta(\bot, j) = \bot$ for all $j \in \Sigma$, so $\bot$ is a sink state.

The DMA $\mathcal{A}_1 = (Q, \Sigma, \delta, 1, \mathcal{F}_1)$, $\mathcal{A}_2 = (Q, \Sigma, \delta, 1, \mathcal{F}_2)$ differ in their acceptance conditions. The set $\{1, \dots, k-m+1\}$ is in $\mathcal{F}_1$. If $\{1, \dots, i\}$ is in $\mathcal{F}_1$ then $\{1, \dots, i+1\}$ is in $\mathcal{F}_2$ for $i < m$. Similarly, if $\{1, \dots, i\} \in \mathcal{F}_2$ then $\{1, \dots, i+1\} \in \mathcal{F}_1$ for $i < m$. Further, $\{\bot\} \in \mathcal{F}_1$ and no other sets are in $\mathcal{F}_1$, $\mathcal{F}_2$.

Clearly, the automaton structure A can recognize both $L(\mathcal{A}_1)$ and $L(\mathcal{A}_2)$. Further, the longest chain in A with respect to $(\mathcal{F}_1, \mathcal{F}_2)$ is the $\mathcal{F}_1$-first m-chain $c = (P_1, \dots, P_m)$ with $P_i = \{1, \dots, k-m+i\}$. Let $\mathcal{F} = \mathcal{F}_1 \cup 2^{\{1,\dots,k-m\}}$ and consider $\mathcal{A} = (Q, \Sigma, \delta, 1, \mathcal{F})$. Then $L(\mathcal{A})$ separates $L(\mathcal{A}_1)$, $L(\mathcal{A}_2)$ and $\mathcal{A}$ has no $(m+1)$-chain.

Let $\mathcal{A}_{\text{sep}} = (Q_{\text{sep}}, \Sigma, \delta_{\text{sep}}, q_{\text{sep}}, \mathcal{F}_{\text{sep}})$ be a DMA with no $(m+1)$-chain that separates $L(\mathcal{A}_1)$ and $L(\mathcal{A}_2)$ and let $r = |Q_{\text{sep}}|$. To prove $|\mathcal{A}_{\text{sep}}| \geq 2^{k-m} - m - 1$, we define an injective function $f : (2^{\{1,\dots,k-m\}} \setminus \{\emptyset, \{1\}, \dots, \{k\}\}) \to \mathcal{F}_{\text{sep}}$.

Definition of the function

For a set $P = \{p_1, \dots, p_{|P|}\} \subseteq \{1, \dots, k-m\}$ with $|P| > 1$ we define the word $w_P = p_1 \dots p_{|P|}$ where $p_1 < p_2 < \dots < p_{|P|}$. Let P be a non-empty subset of $\{1, \dots, k-m\}$ and $P_i = \{1, \dots, k-m+i\}$ with $1 \leq i \leq m$ and consider the words

$$w_0 = w_P^r,$$
$$w_i = (w_{i-1} w_{P_i})^r, \text{ for } 1 \leq i \leq m.$$

Consider the finite run ρ of $\mathcal{A}_{\text{sep}}$ on w_m. For $1 \leq i \leq m$ consider the state in which $\mathcal{A}_{\text{sep}}$ is before reading $w_i = (w_{i-1} w_{P_i})^r$ and the states after each $w_{i-1} w_{P_i}$. These are $r+1 = |Q_{\text{sep}}|+1$ many, so one state must occur twice. So, for $1 \leq i \leq m$ there are positions $0 \leq d_i < e_i \leq |w_m|$ in the word w_m and a number of repetitions $r_i \in \mathbb{N}$ with $\rho(d_i) \xrightarrow{(w_{i-1} w_{P_i})^{r_i}} \rho(e_i)$. Because

w_{i-1} is a proper infix of w_i there is a sequence of indices $d_{m+1} \leq \cdots \leq d_1 < e_1 \leq \cdots \leq e_{m+1}$ with $\rho(d_i) \xrightarrow{(w_{i-1}w_{P_i})^{r_i}} \rho(e_i)$ for $0 \leq i \leq m$. Consider $Q_i = \{\rho(j) \mid d_i \leq j \leq e_i\}$ for $1 \leq i \leq m$.

Similarly, we get d_0, e_0, r_0 with $\rho(d_0) \xrightarrow{(w_P)^{r_0}} \rho(e_0)$ and $d_1 \leq d_o < e_0 \leq e_1$. We set $f(P) = Q_0 = \{\rho(j) \mid d_0 \leq j \leq e_0\}$.

Image of the function

We show that $(Q_1, \ldots, Q_m)$ is an $(2^Q \setminus \mathcal{F}_{\text{sep}})$-start m-chain in $\mathcal{A}_{\text{sep}}$. Notice that each Q_i is a loop and that $Q_0 \subseteq Q_1 \subseteq \cdots \subseteq Q_m$. Further, notice that $P_i \in \mathcal{F}_1$ if i is even and $P_i \in \mathcal{F}_2$ if i is odd for $1 \leq i \leq m$. So, $w_m(w_i)^\omega \in L(\mathcal{A}_1)$ if i is even and $w_m(w_i)^\omega \in L(\mathcal{A}_2)$ if i is odd for $1 \leq i \leq m$. Because $L(\mathcal{A}_{\text{sep}})$ separates $L(\mathcal{A}_1)$ and $L(\mathcal{A}_2)$ we have $w_m(w_i)^\omega \in L(\mathcal{A}_{\text{sep}})$ iff i is odd. Let ρ_i be the run of $\mathcal{A}_{\text{sep}}$ on $w_m(w_i)^\omega$. Then, $\text{Inf}(\rho_i) = Q_i$ and therefore $Q_i \in \mathcal{F}_{\text{sep}}$ iff i is odd.

Now assume that $f(P) = Q_0 \notin \mathcal{F}_{\text{sep}}$ for some $P \subseteq \{1, \ldots, k-m\}$. Then $c = (Q_0, \ldots, Q_m)$ is an $(m+1)$-chain in $\mathcal{A}_{\text{sep}}$ which contradicts the assumption. Thus, $f(P) \in \mathcal{F}_{\text{sep}}$.

The function is injective

Assume that there are $P, P' \subseteq \{1, \ldots, k-m\}$ such that $|P|, |P'| > 1$, $P \neq P'$ and $f(P) = f(P')$. There are states $p \in f(P), p' \in f(P')$ and $r_0, r_0' \in \mathbb{N}$ such that $p \xrightarrow[f(P)]{w_P^{r_0}} p$ and $p' \xrightarrow[f(P)]{w_{P'}^{r_0'}} p'$. We have $P \neq P'$, so $w_P \neq w_{P'}$. Without loss of generality there is a letter a in w_P that does not occur in $w_{P'}$. This letter is mapped to some state q in $f(P)$ and some letter b of $w_{P'}$ is mapped to this state as well because $f(P) = f(P')$. Thus, there are $x, x' \in \Sigma^*$, $a, b \in \Sigma$, $a \neq b$ and a state $q \in f(P)$ such that $q_{\text{sep}} \xrightarrow{xa} q$ and $q_{\text{sep}} \xrightarrow{x'b} q$.

So, the words xa and $x'b$ are mapped to the same state. But $xaa\alpha \notin L(\mathcal{A}_{\text{sep}})$ for all $\alpha \in \Sigma^\omega$ while there are $\alpha \in \Sigma^\omega$ with $xba\alpha \in L(\mathcal{A}_{\text{sep}})$. These words cannot be distinguished by $\mathcal{A}_{\text{sep}}$, contradiction.

Thus, f is an injective mapping from $2^{\{1,\ldots,k-m\}} \setminus \{\emptyset, \{1\}, \ldots, \{k\}\}$ to $\mathcal{F}_{\text{sep}}$ and therefore $2^{k-m} - k - 1 \leq |Q_{\text{sep}}| + |\mathcal{F}_{\text{sep}}| = |\mathcal{A}_{\text{sep}}|$. ◀

The proof idea is based on a construction in [10]. The construction can be extended to show a lower bound for a symmetric separation concept, that is $L_2 \subseteq L$ and $L \cap L_1 = \emptyset$ is also allowed, by adding a copy of A where $\mathcal{F}_1$ and $\mathcal{F}_2$ are swapped.

5.3 Deciding Separability

Let $\mathcal{A}_1$, $\mathcal{A}_2$ be two DMAs with $\mathcal{A}_i = (Q_i, \Sigma, \delta_i, q_i, \mathcal{F}_i)$ and consider their product automaton structure $A_1 \times A_2$. Further, let $\mathcal{F}_B = \{P \subseteq Q_1 \times Q_2 \mid \text{pr}_1(P) \in \mathcal{F}_1\}$ and $\mathcal{F}_R = \{P \subseteq Q_1 \times Q_2 \mid \text{pr}_2(P) \in \mathcal{F}_2\}$. According to Theorem 9 it suffices to determine the loop structure of $A_1 \times A_2$ with respect to $(\mathcal{F}_B, \mathcal{F}_R)$ to decide Wagner-separability.

However, there might be exponentially many loops in $\mathcal{F}_B$ or in $\mathcal{F}_R$. We show that it suffices to consider only certain maximal loops whose number is polynomial in $|\mathcal{A}_1| + |\mathcal{A}_2|$. A similar idea has already been used in [3].

The set $\mathcal{M} \subseteq 2^{Q_1 \times Q_2}$ *contains a set* $P \subseteq Q_1 \times Q_2$ if there are $P_1 \in \mathcal{F}_1$, $P_2 \in \mathcal{F}_2$ such that P is a maximal loop in $P_1 \times P_2$ with respect to set-inclusion. There are at most polynomially many sets of the form $P_1 \times P_2$ and each such set contains at most polynomially many maximal loops because the union of two loops is again a loop. Further, the set $\mathcal{M}_B$ contains a loop P if $P \in \mathcal{M}$ and $\text{pr}_1(P) \in \mathcal{F}_1$. Similarly, $\mathcal{M}_R$ contains a loop P if $P \in \mathcal{M}$ and $\text{pr}_2(P) \in \mathcal{F}_2$.

▶ **Lemma 13.** *Let S be an SCC of $A_1 \times A_2$ and $m \in \mathbb{N}$, $m \geq 1$.*
There is an $\mathcal{F}_B$-start ($\mathcal{F}_R$-start) m-chain in S with respect to $(\mathcal{F}_B, \mathcal{F}_R)$ iff there is an $\mathcal{M}_B$-start ($\mathcal{M}_R$-start) $(m-1)$-chain in S with respect to $(\mathcal{M}_B, \mathcal{M}_R)$.

Proof. "⇒" Let $c = (P_1, \ldots, P_m)$ be an $\mathcal{F}_B$-start m-chain with respect to $(\mathcal{F}_B, \mathcal{F}_R)$. For $1 \leq i < m$ and $b = i \bmod 2$ the set $\mathrm{pr}_{2-b}(P_i) \times \mathrm{pr}_{1+b}(P_{i+1})$ contains a maximal loop P'_i with $\mathrm{pr}_{2-b}(P'_i) = \mathrm{pr}_{2-b}(P_i)$ because $P_i \subseteq \mathrm{pr}_{2-b}(P_i) \times \mathrm{pr}_{1+b}(P_{i+1})$ can be extended to such a maximal loop. Because $P_i \subseteq P_{i+1}$ we have $P'_i \subseteq P'_{i+1}$ for $1 \leq i < m$. So, $c' = (P'_1, \ldots, P'_{m-1})$ is an $(m-1)$-chain with respect to $(\mathcal{M}_B, \mathcal{M}_R)$. Further, $P_1 \in \mathcal{F}_B$ iff $P'_1 \in \mathcal{M}_B$, so c' is $\mathcal{M}_B$-start.

"⇐" Let $c = (P_1, \ldots, P_{m-1})$ be an $\mathcal{M}_B$-start $(m-1)$-chain with respect to $(\mathcal{M}_B, \mathcal{M}_R)$. Every chain with respect to $(\mathcal{M}_B, \mathcal{M}_R)$ is a chain with respect to $(\mathcal{F}_B, \mathcal{F}_R)$ because $\mathcal{M}_B \subseteq \mathcal{F}_B$ and $\mathcal{M}_R \subseteq \mathcal{F}_R$. So, c is an $(m-1)$-chain with respect to $(\mathcal{F}_B, \mathcal{F}_R)$. Further, c is $\mathcal{F}_B$-start.

Consider the case $P_{m-1} \in \mathcal{M}_B$. By the definition of $\mathcal{M}$ there are $R_1 \in \mathcal{F}_1, R_2 \in \mathcal{F}_2$ such that $P_{m-1} \subseteq R_1 \times R_2$. So, there is a loop P_m with $P_{m-1} \subseteq P_m \subseteq Q_1 \times R_2$ and $\mathrm{pr}_2(P_m) = R_2$. Thus, $P_m \in \mathcal{M}_R$ and $(P_1, \ldots, P_{m-1}, P_m)$ is an $\mathcal{F}_B$-start m-chain with respect to $(\mathcal{F}_B, \mathcal{F}_R)$.

The case that c is $\mathcal{M}_R$-start follows analogously. ◀

▶ **Theorem 14.** *The problem* WagnerSeparation *can be decided in polynomial time.*

Proof. According to Lemma 13, it suffices to determine the superchains with respect to $(\mathcal{M}_B, \mathcal{M}_R)$. This can be done in polynomial time as mentioned in Section 5.3. ◀

The intersection of two DMAs can cause an exponential blowup [2]. In contrast to this, Lemma 14 implies that the disjointness (emptiness of the intersection) of two DMAs can be checked in polynomial time.

▶ **Corollary 15.** *Given two DMAs $\mathcal{A}_1$, $\mathcal{A}_2$, it can be decided in polynomial time whether $L(\mathcal{A}_1) \cap L(\mathcal{A}_2)$ is empty.*

Proof. $L(\mathcal{A}_1) \cap L(\mathcal{A}_2) \neq \emptyset$ iff there are arbitrarily long chains in $A_1 \times A_2$ with respect to $(\mathcal{F}_B, \mathcal{F}_R)$ iff $L(\mathcal{A}_1)$, $L(\mathcal{A}_2)$ are not E^1_m-separable for $m = |Q_1| \cdot |Q_2| + 1$. The proof of these equivalences mirrors the proof of Remark 1. ◀

5.4 Wagner Separation for Parity Automata

In this section we consider the Wagner separation problem with deterministic parity automata as input. A *deterministic parity automaton* (DPA) is a tuple $\mathcal{A} = (Q, \Sigma, \delta, q_0, \Omega)$ where $A = (Q, \Sigma, \delta, q_0)$ is an automaton structure and $\Omega : Q \to \mathbb{N}$ is the priority function of $\mathcal{A}$. An infinite word $\alpha \in \Sigma^\omega$ is *accepted* by $\mathcal{A}$ if the maximal priority seen infinitely often by the run ρ of A in α is even. That is, $\max(\{\Omega(q) \mid q \in \mathrm{Inf}(\rho)\})$ is even. For a set $P \subseteq Q$ let $\Omega(P) = \{\Omega(p) \mid p \in P\}$. Since every DPA can be transformed into an equivalent DPA with $\Omega(Q) = \{0, \ldots, |Q|\}$ we define the size of a DPA $\mathcal{A} = (Q, \Sigma, \delta, q_0, \Omega)$ as $|Q|$.

An ω-language L is regular iff there is a DPA $\mathcal{A}$ with $L(\mathcal{A}) = L$, so DMAs and DPAs define the same class of languages. However, there are languages for which every DMA is exponentially larger than the smallest DPA for the language and there are languages for which every DPA is exponentially larger than the smallest DMA for the language [1]. So, it makes a difference with respect to computational complexity whether a language is given as a DMA or as a DPA.

WagnerSeparationParity
Given: Two DPAs $\mathcal{A}_1, \mathcal{A}_2$, $X \in \{C, D, E\}$ and $m, n \in \mathbb{N}$.
Decide: Are $L(\mathcal{A}_1)$ and $L(\mathcal{A}_2)$ X^n_m-separable?

As in the previous section, we use maximal chains to show that this problem can be solved in polynomial time. This implies that disjointness of two languages given as DPA can be decided in polynomial time, too.

▶ **Theorem 16.** *WagnerSeparationParity can be decided in polynomial time.*

Proof. Let $\mathcal{A}_1, \mathcal{A}_2$ be two DPAs, $\mathcal{A}_i = (Q_i, \Sigma, \delta_i, q_i, \Omega_i)$ and A_i the corresponding automaton structure for $i \in \{1, 2\}$. Consider their product automaton and the acceptance conditions $\mathcal{F}_B = \{P \subseteq Q_1 \times Q_2 \mid \max(\Omega_1(\text{pr}_1(P)))$ is even and P is a loop$\}$ and $\mathcal{F}_R = \{P \subseteq Q_1 \times Q_2 \mid \max \Omega_2(\text{pr}_2(P)))$ is even and P is a loop$\}$. As shown in Section 5.3 the chain structure with respect to $(\mathcal{F}_B, \mathcal{F}_R)$ suffices to decide separability. However, $\mathcal{F}_1$ and $\mathcal{F}_2$ might be of exponential size.

For $q, q' \in Q_1 \times Q_2$, $k_1, k_2 \in \mathbb{N}$ we denote with $q \xrightarrow[\leq(k_1,k_2)]{} q'$ that there is a word $w \in \Sigma^*$ and a run ρ of $A_1 \times A_2$ on w from q to q' such that for all $0 \leq j \leq |\rho|$ we have $\text{pr}_1(\rho(j)) \leq k_1$ and $\text{pr}_2(\rho(j)) \leq k_2$. For $q \in Q_1 \times Q_2$, $k_1, k_2 \in \Omega(Q)$ consider the set $P_q^{k_1,k_2} = \{q' \mid q \xrightarrow[\leq(k_1,k_2)]{} q' \xrightarrow[\leq(k_1,k_2)]{} q\}$. Consider $\mathcal{M}_B = \{P_q^{k_1,k_2} \neq \emptyset \mid q \in Q_1 \times Q_2, k_1, k_2 \in \Omega(Q)$ and k_1 is even$\}$ and $\mathcal{M}_R = \{P_q^{k_1,k_2} \neq \emptyset \mid q \in Q_1 \times Q_2, k_1, k_2 \in \Omega(Q)$ and k_2 is even$\}$. These sets can be computed in polynomial time.

Let S be an SCC of $A_1 \times A_2$. We show that there is an m-chain c in S with respect to $(\mathcal{M}_B, \mathcal{M}_R)$ iff there is an m-chain c' in S with respect to $(\mathcal{F}_B, \mathcal{F}_R)$. Further, we show that c is $\mathcal{M}_B$-start iff c' is $\mathcal{F}_B$-start.

"$\Rightarrow$" We have $\mathcal{M}_B \subseteq \mathcal{F}_B$ and $\mathcal{M}_R \subseteq \mathcal{F}_R$, so every m-chain with respect to $(\mathcal{M}_B, \mathcal{M}_R)$ is an m-chain with respect to $(\mathcal{F}_B, \mathcal{F}_R)$ and it is $\mathcal{M}_B$-start iff it is $\mathcal{F}_B$-start.

"$\Leftarrow$" Let $c' = (P'_1, \ldots, P'_m)$ an m-chain in $A_1 \times A_2$ with respect to $(\mathcal{F}_B, \mathcal{F}_R)$. Let $p \in P'_1$, $k_1^i = \max(\Omega_1(\text{pr}_1(P'_i)))$ and $k_2^i = \max(\Omega_2(\text{pr}_2(P'_i)))$. Then $c = (P_p^{k_1^1,k_2^1}, \ldots, P_p^{k_1^m,k_2^m})$ is an m-chain with respect to $(\mathcal{M}_B, \mathcal{M}_R)$. Further, c is $\mathcal{M}_B$ start iff c' is $\mathcal{F}_B$-start and the chains are in the same SCC.

Thus, the chain structure with respect to $(\mathcal{M}_B, \mathcal{M}_R)$ is the same as it is with respect to $(\mathcal{F}_B, \mathcal{F}_R)$. So, separability can be decided using $(\mathcal{M}_B, \mathcal{M}_R)$ as done in Section 5.3. ◀

6 Conclusion

We have seen that separation of two languages given by two DMAs with respect to the Wagner hierarchy can be viewed as analyzing the loop structure in their product automaton. Using this result one can compute a separator in exponential time. We showed that there are languages of DMAs whose separator requires exponential size. So, if one wants to compute the complete separator then exponential time is optimal. However, we can decide separation with respect to the Wagner hierarchy in polynomial time using maximal loops. This implies that we can decide disjointness of two languages given as DMAs in polynomial time. A variation of the separation problem where the languages are given as DPAs can be solved in polynomial time as well.

A separating DMA can be large because it has to list all accepting loops explicitly. Meanwhile, we can decide separation efficiently because we can restrict ourselves to maximal loops. These maximal loops, in a sense, give a more succinct representation of the acceptance condition of a DMA.

In a follow-up paper we will investigate a new automaton model based on this succinct representation. We hope that this new model has better properties with respect to computational complexity than current automata models.

References

1. Udi Boker. On the (in) succinctness of Muller automata. In *26th EACSL Annual Conference on Computer Science Logic (CSL 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.
2. Udi Boker. Why these automata types? In *LPAR*, volume 18, pages 143–163, 2018.
3. Olivier Carton and Ramón Maceiras. Computing the Rabin index of a parity automaton. *RAIRO-Theoretical Informatics and Applications*, 33(6):495–505, 1999.
4. Thomas Colcombet and Christof Löding. The nesting-depth of disjunctive μ-calculus for tree languages and the limitedness problem. In *International Workshop on Computer Science Logic*, pages 416–430. Springer, 2008.
5. Damian Niwiński and Igor Walukiewicz. Relating hierarchies of word and tree automata. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 320–331. Springer, 1998.
6. Damian Niwiński and Igor Walukiewicz. Deciding nondeterministic hierarchy of deterministic tree automata. *Electronic Notes in Theoretical Computer Science*, 123:195–208, 2005.
7. Dominique Perrin and Jean-Éric Pin. *Infinite words: automata, semigroups, logic and games*. Academic Press, 2004.
8. Thomas Place and Marc Zeitoun. Separating regular languages with first-order logic. In *Proceedings of the Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–10, 2014.
9. Thomas Place and Marc Zeitoun. The tale of the quantifier alternation hierarchy of first-order logic over words. *ACM SIGLOG News*, 2(3):4–17, 2015.
10. Shmuel Safra. *Complexity of automata on infinite objects*. PhD thesis, The Weizmann Institute of Science, 1989.
11. Marcel Paul Schützenberger. On finite monoids having only trivial subgroups. *Inf. Control.*, 8(2):190–194, 1965.
12. Klaus Wagner. On ω-regular sets. *Information and control*, 43(2):123–177, 1979.