# Complexity of Topological Properties of Regular $\omega$-Languages

Victor L. Selivanov*
A.P. Ershov Institute of Informatics Systems
Siberian Division of the Russian Academy of Sciences
vseliv@nspu.ru
and
Klaus W. Wagner
Institut für Informatik
Julius-Maximilians-Universität Würzburg
wagner@informatik.uni-wuerzburg.de

**Abstract**

We determine the complexity of topological properties (i.e., properties closed under the Wadge equivalence) of regular $\omega$-languages by showing that they are typically NL-complete (PSPACE-complete) for the deterministic Muller, Mostowski and Büchi automata (respectively, for the nondeterministic Rabin, Muller, Mostowski and Büchi automata). For the deterministic Rabin and Streett automata and for the nondeterministic Streett automata upper and lower complexity bounds for the topological properties are established.

## 1 Introduction

The study of decidability and complexity questions for properties of regular languages is a central research topic in automata theory. Its importance stems from the fact that finite automata are fundamental to many branches of computer science, e.g., databases, operating systems, verification, and hardware and software design.

There are many examples for decidable properties of regular languages (e.g., dot-depth one), while the decidability of other properties is still a challenging open question (e.g., dot-depth two, generalized star-height). Moreover, among the decidable properties there is a broad range of complexity results. For some of them, e.g., for the dot-depth one property, efficient algorithms are known that work in nondeterministic logarithmic space (NL) and hence in polynomial time. For other properties, a membership test needs more resources, e.g., deciding the aperiodicity property of regular aperiodic languages is PSPACE-complete.

In this paper we determine the complexity of topological properties of regular $\omega$-languages given by different types of $\omega$-automata. Topological properties are classes of $\omega$-languages which are closed under inverse continuous functions. Defining the *Wadge reducibility* $\leq_{\rm w}$ on the Cantor space as the many-one reducibility via continuous functions, the topological properties are the classes of $\omega$-languages which are closed under Wadge reducibility. The classes $\{L' \mid L' \leq_{\rm w} L\}$ for $\omega$-languages $L$ are called *elementary* topological properties; every topological property is the union of elementary topological properties. Obviously, there is a bijection between the elementary topological properties and the Wadge degrees.

To explain our results, let us recall some facts from [Wag79] where the Wadge degrees of regular $\omega$-languages (over any alphabet $A$ with at least two symbols) were determined, in particular the following results were established:

*Supported by DFG Mercator program and by RFBR grant 07-01-00543a.

1. The structure $(\mathcal{R}; \leq_{\mathrm{w}})$ of regular $\omega$-languages under the Wadge reducibility is almost well-ordered with order type $\omega^\omega$, i.e.,x for each ordinal $\alpha < \omega^\omega$ there is a regular $\omega$-languages $A_\alpha \in \mathcal{R}$, such that $A_\alpha <_{\mathrm{w}} A_\alpha \oplus \overline{A}_\alpha <_{\mathrm{w}} A_\beta$ for $\alpha < \beta < \omega^\omega$, and any regular set is Wadge-equivalent to one of the sets $A_\alpha$, $\overline{A}_\alpha$, and $A_\alpha \oplus \overline{A}_\alpha$ where $\alpha < \omega^\omega$.
2. The elementary topological properties of regular $\omega$-languages are $\mathcal{R}_\alpha =_{\mathrm{def}} \{L \mid L \leq_{\mathrm{w}} A_\alpha\}$, co-$\mathcal{R}_\alpha =_{\mathrm{def}} \{L \mid L \leq_{\mathrm{w}} \overline{A}_\alpha\}$, and $\mathcal{R}_{\alpha+1} \cap \text{co-}\mathcal{R}_{\alpha+1} = \{L \mid L \leq_{\mathrm{w}} A_\alpha \oplus \overline{A}_\alpha\}$. The Wadge-degrees of regular $\omega$-languages are $\mathcal{R}'_\alpha =_{\mathrm{def}} \{L \mid L \equiv_{\mathrm{w}} A_\alpha\} = \mathcal{R}_\alpha \setminus \text{co-}\mathcal{R}_\alpha$, co-$\mathcal{R}'_\alpha =_{\mathrm{def}} \{L \mid L \equiv_{\mathrm{w}} \overline{A}_\alpha\} =$ co-$\mathcal{R}_\alpha \setminus \mathcal{R}_\alpha$, and $\tilde{\mathcal{R}}_\alpha =_{\mathrm{def}} (\mathcal{R}_{\alpha+1} \cap \text{co-}\mathcal{R}_{\alpha+1}) \setminus (\mathcal{R}_\alpha \cup \text{co-}\mathcal{R}_\alpha)$
3. All elementary topological properties of regular $\omega$-languages and all Wadge-degrees of regular $\omega$-languages are decidable (the regular $\omega$-languages given by deterministic Muller automata).

A natural question is to determine the complexity of the classes listed under 2. for different popular types of $\omega$-automata such as deterministic or nondeterministic Büchi, Muller, Rabin, Streett and Mostowski (or parity) automata. To our knowledge, only a couple of results in this direction were established so far. They are collected in the following

**Theorem 1.1** *1.* **[KPB95, WY95]** *For every $\alpha < \omega^\omega$, given a deterministic Muller automaton $\mathcal{M}$, one can decide in polynomial time whether $L_\omega(\mathcal{M}) \in \mathcal{R}_\alpha$.*

*2.* **[SVW87]** *The problem of deciding, given a nondeterministic Büchi automaton $\mathcal{M}$ with input alphabet $A$, whether $L_\omega(\mathcal{M}) = A^\omega$, is* PSPACE*-complete.*

*3.* **[SVW87]** *The problem of deciding, given a nondeterministic Büchi automaton $\mathcal{M}$, whether $L_\omega(\mathcal{M}) = \emptyset$, is* NL*-complete.*

The Statements 2 and 3 above are related to the classes $\mathcal{R}_\alpha$ because $\mathcal{R}_0$ coincides with $\{\emptyset\} = \{L \mid L \leq_{\mathrm{w}} \emptyset\}$ and the dual class co-$(\mathcal{R}_0)$ for $\mathcal{R}_0$ coincides with $\{A^\omega\} = \{L \mid L \leq_{\mathrm{w}} A^\omega\}$.

We will determine the complexity of all elementary topological properties of regular $\omega$-languages and all Wadge-degrees of regular $\omega$-languages w.r.t. the mentioned types of $\omega$-automata. These results are represented in the following table. Let $\mathcal{C}$ be an elementary topological property of regular $\omega$-languages, i.e., $\mathcal{C} \in \{\mathcal{R}_\alpha, \text{co-}\mathcal{R}_\alpha, \mathcal{R}_{\alpha+1} \cap \text{co-}\mathcal{R}_{\alpha+1} \mid \alpha < \omega^\omega\}$ or a Wadge-degree of regular $\omega$-languages, i.e., $\mathcal{C} \in \{\mathcal{R}'_\alpha, \text{co-}\mathcal{R}'_\alpha, \tilde{\mathcal{R}}_{\alpha+1} \mid \alpha < \omega^\omega\}$. For deterministic Büchi automata this is restricted to $\mathcal{C} \subseteq \mathcal{R}_\omega$ because they can accept only such regular $\omega$-languages from $\mathcal{R}_\omega$. The lower bounds mean hardness for the complexity class in question.

| automata type | $\mathcal{C}$ | deterministic | | nondeterministic | |
|---|---|---|---|---|---|
| | | lower bound | upper bound | lower bound | upper bound |
| Muller | $= \mathcal{R}_0$ | NL | NL | NL | NL |
| | $\neq \mathcal{R}_0$ | NL | NL | PSPACE | PSPACE |
| Rabin | $= \mathcal{R}_0$ | NL | NL | NL | NL |
| | $\neq \mathcal{R}_0$ | P | $\mathrm{P}^{\mathrm{NP}}$ | PSPACE | PSPACE |
| Streett | $=$ co-$\mathcal{R}_0$ | NL | NL | P | co-NP |
| | $\neq$ co-$\mathcal{R}_0$ | P | $\mathrm{P}^{\mathrm{NP}}$ | PSPACE | EXPSPACE |
| Mostowski | $= \mathcal{R}_0$ | NL | NL | NL | NL |
| | $\neq \mathcal{R}_0$ | NL | NL | PSPACE | PSPACE |
| Büchi | $= \mathcal{R}_0$ | NL | NL | NL | NL |
| | $\neq \mathcal{R}_0$ | NL | NL | PSPACE | PSPACE |

In Sections 2 and 3 we recall the notation and necessary facts about $\omega$-languages, topology and finite automata. Section 4 recalls necessary information from [Wag79] about topological properties of regular $\omega$-languages. In Section 5 we establish upper complexity bounds for deterministic automata. In Sections 6 and 7 we show the results for deterministic Muller, Mostowski and Büchi automata, and Section 8 provides the results for Rabin and Streett automata. In Section 9 we show the results for all nondeterministic types of $\omega$-automata.

## 2 $\omega$-Languages and Topology

We use standard set-theoretic notation. For a set $S$, let $P(S)$ be the class of subsets of $S$. For a class $\mathcal{C} \subseteq P(S)$, let co-$\mathcal{C}$ be the dual class $\{\overline{C} \mid C \in \mathcal{C}\}$ and let $\mathrm{BC}(\mathcal{C})$ be the Boolean closure of $\mathcal{C}$.

Fix a finite alphabet $A$ containing more than one symbol. For simplicity we may assume that $A$ is one of the alphabets $A_k =_{\mathrm{def}} \{0, 1, \ldots, k-1\}$ for $k > 1$, so $0, 1 \in A$. Let $A^*$ and $A^\omega$ denote respectively the sets of all words and of all $\omega$-words (i.e. sequences $\alpha : \mathbb{N} \to A$) over $A$. The empty word is denoted by $\varepsilon$. Let $A^+ = A^* \setminus \{\varepsilon\}$ and $A^{\leq\omega} = A^* \cup A^\omega$. For $n \in \mathbb{N}$, let $A^n$ be the set of words of length $n$. Note that usually we work with the fixed alphabet $A$ but sometimes we consider several alphabets simultaneously. The "fixed-alphabet mode" is the default one.

We use some almost standard notation concerning words and $\omega$-words, so we are not too casual in reminding it here. For $w \in A^*$ and $\xi \in A^{\leq\omega}$, $w \sqsubseteq \xi$ means that $w$ is a initial part of $\xi$, $w \cdot \xi = w\xi$ denotes the concatenation, and $l = |w|$ is the length of $w = w(0) \cdots w(l-1)$. For $u \in A^*$ and $n < \omega$, let $u^n$ denote the concatenation of $n$ copies of the word $u$. For $w \in A^*, W \subseteq A^*$ and $L \subseteq A^{\leq\omega}$, let $w \cdot L = \{w\xi \mid \xi \in L\}$, let $W \cdot L = \{w\xi \mid w \in W, \xi \in L\}$, let $W^\omega = \{w_0 w_1 \cdots \in A^\omega \mid w_i \in W\}$.

The set $A^\omega$ carries the Cantor topology with the open sets $W \cdot A^\omega$, where $W \subseteq X^*$. Let $\mathcal{B}$ denote the class of Borel subsets of $A^\omega$, i.e. the least class containing the open sets and closed under complement and countable union. Borel sets are organized in a hierarchy the lowest levels of which are as follows: $G$ and $F$ are the classes of open and closed sets, respectively; $G_\delta$ ($F_\sigma$) is the class of countable intersections (unions) of open (resp. closed) sets; $G_{\delta\sigma}$ ($F_{\sigma\delta}$) is the class of countable unions (intersections) of $G_\delta$- (resp. of $F_\sigma$-) sets, and so on. In the modern notation of hierarchy theory, $\mathbf{\Sigma}^0_1 = G$, $\mathbf{\Sigma}^0_2 = F_\sigma$, $\mathbf{\Sigma}^0_3 = G_{\delta\sigma}$, $\mathbf{\Sigma}^0_4 = F_{\sigma\delta\sigma}$ and so on, $\mathbf{\Pi}^0_n =_{\mathrm{def}}$ co-$\mathbf{\Sigma}^0_n$ is the dual class for $\mathbf{\Sigma}^0_n$, and $\mathbf{\Delta}^0_n = \mathbf{\Sigma}^0_n \cap \mathbf{\Pi}^0_n$. The sequence $\{\mathbf{\Sigma}^0_{n+1}\}_{n<\omega}$ is known as *the finite Borel hierarchy*. It may be in a natural way extended on all countable ordinals. The resulting sequence called *the Borel hierarchy* exhausts the class $\mathcal{B}$. For any $n > 0$, the class $\mathbf{\Sigma}^0_n$ contains $\emptyset, A^\omega$ and is closed under countable unions and finite intersections, while the class $\mathbf{\Delta}^0_n$ is closed under complement and finite unions. For any $n > 0$, we have the strict inclusions $\mathbf{\Sigma}^0_n \cup \mathbf{\Pi}^0_n \subset \mathrm{BC}(\mathbf{\Sigma}^0_n) \subset \mathbf{\Delta}^0_{n+1}$.

For $L, K \subseteq A^\omega$, $L$ is said to be *Wadge reducible* to $K$ (in symbols $L \leq_{\mathrm{w}} K$), if $L = g^{-1}(K)$ for some continuous function $g : A^\omega \to A^\omega$. The Wadge reducibility on $P(A^\omega)$ is a preorder. By $\equiv_{\mathrm{w}}$ we denote the induced equivalence relation which gives rise to the corresponding quotient partial ordering. Following a well established jargon, we call this ordering the structure of Wadge degrees [Wa72, Wa84]. The operation $L \oplus K = \{0 \cdot \xi \cup i \cdot \eta \mid 0 < i < k, \xi \in L, \eta \in K\}$ on subsets of $A^\omega_k$ induces the operation of least upper bound in the structures of Wadge degrees. Any level of the Borel hierarchy is closed under the Wadge reducibility in the sense that every set reducible to a set in the level is itself in that level. Moreover, every $\mathbf{\Sigma}$-level $\mathcal{C}$ (and also every $\mathbf{\Pi}$-level) of the Borel hierarchy has a Wadge complete set $C$ which means that $\mathcal{C} = \{L \mid L \leq_{\mathrm{w}} C\}$. For additional information on $\omega$-languages see e.g. [Sta97, Th90, Th96].

## 3 Finite Automata Accepting $\omega$-Languages

Finite automata may accept $\omega$-languages in different ways. Here we briefly recall some acceptance modes and corresponding facts that will be used later.

By *deterministic pre-automaton* (over $A$) we mean a triple $\mathcal{M} = (S, A, \delta)$ consisting of a finite non-empty set $S$ of states, an input alphabet $A$ and a transition function $\delta : S \times A \to S$. The transition function is naturally extended to the function $\delta : S \times A^* \to S$ defined by induction $\delta(s, \varepsilon) =_{\mathrm{def}} s$ and $\delta(s, xa) =_{\mathrm{def}} \delta(\delta(s, x), a)$ where $x \in A^*$ and $a \in A$. Furthermore, we define the function $\overline{\delta} : S \times A^* \to P(S)$ by $\overline{\delta}(s, x) =_{\mathrm{def}} \{\delta(s, u) \mid u \sqsubseteq x\}$. For input sequences from $A^\omega$ define the function $\delta : S \times A^\omega \to S^\omega$ by $\delta(s, \xi)(n) = \delta(s, \xi[n])$.

*Nondeterministic pre-automata* are defined in the same way only now the transition function is of the form $\delta : S \times A \to P(S)$ which is extended to the fuction $\delta : S \times A^* \to P(S)$ by $\delta(s, \varepsilon) =_{\mathrm{def}} \{s\}$ and $\delta(s, xa) =_{\mathrm{def}} \bigcup_{s \in \delta(s,x)} \delta(s, a)$ where $x \in A^*$ and $a \in A$. As is well known, deterministic pre-automata may be considered as a particular case of the nondeterministic ones. For input sequences from $A^\omega$ define the function $\delta : S \times A^\omega \to P(S^\omega)$ by $\delta(s,\xi) =_{\rm def} \{\eta \mid \eta(0) = s \wedge \forall i(\eta(i+1) \in \delta(\eta(i),\xi(i)))\}$.

Pre-automata equipped with appropriate additional structures are used as acceptors, i.e. devises accepting words or $\omega$-words. A *deterministic automaton* (dfa for short) is a quadruple $\mathcal{M} = (S, A, \delta, s_0, F)$ where $(S, A, \delta)$ is a pre-automaton, $s_0 \in S$, (the *initial state*), and $F \subseteq S$ (the set of *final states*). Such an automaton recognizes the language $L(\mathcal{M}) = \{x \in A^* \mid \delta(s,x) \in F\}$. *Nondeterministic automata* (nfa) are defined analogously. Such an automaton $(\mathcal{M}, s, F)$ accepts the language $L(\mathcal{M}) = \{x \in A^* \mid \delta(s,x) \cap F \neq \emptyset\}$. It is well-known that deterministic and nondeterministic automata accept the same class of languages which are called *regular* languages.

Unlike automata on finite words, for automata on $\omega$-words the acceptance conditions were defined in different way by different authors, and it is not clear which of these conditions are more natural than the others. As a result, there are several notions of automata accepting $\omega$-words (which we generally call *$\omega$-automata*). Let us briefly recall the most popular versions. For $\eta \in S^\omega$, let $\inf(\eta)$ be the set of all $s \in S$ which occur infinitely often in $\eta$.

A deterministic *Büchi automaton* is a quadruple $\mathcal{M} = (S, A, \delta, s_0, F)$ where $(S, A, \delta)$ is a determininstic pre-automaton, $s_0 \in S$, and $F \subseteq S$. It recognizes the set $L_\omega(\mathcal{M}) = \{\xi \in A^\omega \mid \inf(\delta(s_0,\xi)) \cap F \neq \emptyset\}$.

A deterministic *Muller automaton* is a quadruple $\mathcal{M} = (S, A, \delta, s_0, \mathcal{F})$ where $(S, A, \delta)$ is a determininstic pre-automaton, $s_0 \in S$, and $\mathcal{F} \subseteq P(S)$. It recognizes the set $L_\omega(\mathcal{M}) = \{\xi \in A^\omega \mid \inf(\delta(s_0,\xi)) \in \mathcal{F}\}$.

A deterministic *Rabin automaton* is a quadruple $\mathcal{M} = (S, A, \delta, s_0, \mathcal{F})$ where $(S, A, \delta)$ is a determininstic pre-automaton, $s_0 \in S$, and $\mathcal{F} \subseteq P(S) \times P(S)$. It recognizes the set $L_\omega(\mathcal{M}) = \{\xi \in A^\omega \mid \exists((E,F) \in \mathcal{F})(\inf(\delta(s_0,\xi)) \cap E = \emptyset \wedge \inf(\delta(s_0,\xi)) \cap F \neq \emptyset)\}$.

A deterministic *Mostowski automaton* (known also as Rabin chain automaton or parity automaton) is the special case of a deterministic Rabin automaton $\mathcal{M} = (S, A, \delta, s_0, \mathcal{F})$ where $\mathcal{F} = \{(E_1, F_1), (E_2, F_2), \ldots, (E_m, F_m)\}$ satisfies $E_1 \subseteq F_1 \subseteq E_2 \subseteq F_2 \subseteq \cdots \subseteq E_m \subseteq F_m$.

A deterministic *Streett automaton* is formally the same object as a deterministic Rabin automaton $\mathcal{M}$, but it recognizes the set $L'_\omega(\mathcal{M}) = \{\xi \in A^\omega \mid \forall((E,F) \in \mathcal{F})(\inf(\delta(s_0,\xi)) \cap E \neq \emptyset \vee \inf(\delta(s_0,\xi)) \cap F = \emptyset)\}$.

Notice that deterministic Streett automata are complementary to deterministic Rabin automata. This means $L'_\omega(\mathcal{M}) = A^\omega \setminus L_\omega(\mathcal{M})$ for every deterministic Rabin automaton $\mathcal{M}$.

The nondeterministic versions of the introduced types of automata are defined in the usual way: We start with a nondeterministic pre-automaton and instead of the acceptance condition $H(\inf(\delta(s_0,\xi)))$ we use the acceptance condition $\exists\eta(\eta \in \delta(s_0,\xi) \wedge H(\inf(\eta)))$, i.e. there is an infinite run such that the corresponding sequence of states satisfies the acceptance condition.

**Theorem 3.1** *For any $\omega$-language $L \subseteq A^\omega$ the following statements are equivalent:*

1. *$L$ is recognized by a deterministic Muller (Rabin, Mostowski, Streett) automaton.*
2. *$L$ is recognized by a nondeterministic Büchi (Muller, Rabin, Mostowski, Streett) automaton.*
3. *$L$ is a finite union of sets $U \cdot V^\omega$ where $U \subseteq A^*$ and $V \subseteq A^+$ are regular languages.*

The $\omega$-languages satisfying the assertions above are called *regular $\omega$-languages.* Let $\mathcal{R}$ be the class of regular $\omega$-languages.

**Theorem 3.2** 1. *$\mathcal{R} \subset \mathrm{BC}(\mathbf{\Sigma}^0_2)$.*

2. **[La69, SW74]** *The deterministic Büchi automata accept exactly the regular $\mathbf{\Pi}^0_2$-sets.*

For the above defined types of automata we introduce the abbreviatons B, M, R, P, and S for Büchi automata, Muller automata, Rabin automata, Mostowski (parity) automata, and Strett automata, resp., and D and N stand for deterministic and nondeterministic, resp. In this way, for example, NB is the name for nondeterministic Büchi automata.

Let $\mathcal{C}$ be a class of $\omega$-languages, and let $T$ be a type of automata. We consider the

*Problem* $(\mathcal{C})_T$:
*Given:* An automaton $\mathcal{M}$ of type $T$.
*Question:* Does $\mathcal{M}$ accept an $\omega$-language in $\mathcal{C}$?

Because of the duality of the deterministic Rabin acceptance and the deterministic Streett acceptance we have

**Proposition 3.3** *If $\mathcal{C}$ is a class of $\omega$-languages then $(\mathcal{C})_{\mathrm{DS}} \equiv_{\mathrm{m}}^{\log} (\text{co-}\mathcal{C})_{\mathrm{DR}}$.*

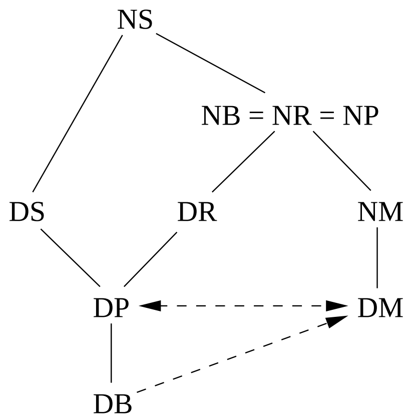
By Theorem 3.1 all the introduced classes of $\omega$-automata (besides deterministic Büchi automata) are equivalent in the sense that they recognize the same $\omega$-languages. Moreover, the well known proofs of these equivalences are effective, i.e. from a given automaton of some type one can compute an equivalent automaton of any other type. When one is interested in complexity considerations (as we are here), the computational resources needed for finding the equivalent automaton and its size become important.

We say that a type $T$ of $\omega$-automata is polynomial time reducible to a type $T'$ of $\omega$-automata (for short $T \leq^{\mathrm{p}} T'$) if there exists a polynomial time computable function $f$ such that, for every automaton $\mathcal{M}$ of type $T$, the result $f(\mathcal{M})$ is an automaton of type $T'$ which accepts the same $\omega$-language as $\mathcal{M}$. The following relationship to decision problems is obvious:

**Proposition 3.4** *Let $T$ and $T'$ be two types of $\omega$-automata, and let $\mathcal{C}$ be a class of $\omega$-languages. Then $T \leq^{\mathrm{p}} T'$ implies $(\mathcal{C})_T \leq_{\mathrm{m}}^{\mathrm{p}} (\mathcal{C})_{T'}$.*

Unfortunately, some of the well known reductions in Theorem 3.1 do not work in polynomial time. For some cases one can even prove that this is not possible. In [Sa88] an overview on possibility or impossibility of polynomial time reductions between different types of $\omega$-automata is given.

**Theorem 3.5 [Sa88]** *The following figure represents some results on polynomial time reductions between different types of $\omega$-automata. A solid line means that here exists a polynomial time reduction from the notion below to the notion above. A dotted arc means that polynomial time reduction in this direction is not proved and not disproved. Moreover, there are no further polynomial time reductions between these types of $\omega$-automata which do not already follow from the solid lines and dotted arcs.*




# 4 Topological Properties of Regular $\omega$-Languages

Topological properties are classes of $\omega$-languages which are closed under Wadge reducibility, i.e., under inverse continuous functions. Theses are just the classes $\{L \mid \exists L'(L' \in \mathcal{C} \wedge L \leq_{\mathrm{w}} L')\}$ where $\mathcal{C} \subseteq$

$P(A^\omega)$. We are interested in topological properties of *regular $\omega$-languages*, these are just the classes $\hat{\mathcal{C}} =_{\text{def}} \{L \mid \exists L'(L' \in \mathcal{C} \wedge L \leq_{\text{w}} L')\} \cap \mathcal{R}$ where $\mathcal{C} \subseteq \mathcal{R}$. If $[L]_{\text{w}}$ is the $\equiv_{\text{w}}$-equivalence class which includes $L \subseteq A^\omega$ (the *Wadge degree* of $L$) then there holds $\hat{\mathcal{C}} = \bigcup_{L\in\mathcal{C}} \widehat{[L]_{\text{w}}}$ for every $\mathcal{C} \subseteq \mathcal{R}$. That means: we know all topological properties of regular $\omega$-languages if we know all *elementary topological properties* $\widehat{[L]_{\text{w}}}$ of regular $\omega$-languages. Furthermore, we know these, if we know all *regular Wadge degrees* $[L]_{\text{w}} \cap \mathcal{R}$. We define the family $\mathcal{T} =_{\text{def}} \{[L]_{\text{w}} \cap \mathcal{R} \mid L \in \mathcal{R}\}$ of all regular Wadge degrees and the family $\hat{\mathcal{T}} =_{\text{def}} \{\widehat{[L]_{\text{w}}} \mid L \in \mathcal{R}\}$ of all elementary topological properties of regular $\omega$-languages.

These families of classes were completely characterized in [Wag79] by some invariants of deterministic Muller automata. We recall in this section the definitions and results from this paper which we need here. In what follows let $\mathcal{M} = (S, A, \delta, s_0, \mathcal{F})$ be a deterministic Muller automaton.

A subset $S' \subseteq S$ is called a *loop* if there exist an $s \in S$ and $x, z \in A^*$ such that $\delta(s_0, x) = \delta(s, z) = s$ and $\overline{\delta}(s, z) = S'$. A loop $S_2$ is *reachable* from a loop $S_1$ if there exists an $s \in S_1$ and an $x \in A^*$ such that $\delta(s, x) \in S_2$.

For $m \geq 1$, an $m^+$ *chain* is a sequence $(S_1, S_2, \ldots, S_m)$ of loops such that $S_1 \subset S_2 \subset \cdots \subset S_m$, $S_1, S_3, \dots \in \mathcal{F}$, and $S_2, S_4, \dots \in P(S) \smallsetminus \mathcal{F}$. An $m^-$ *chain* is a sequence $(S_1, S_2, \ldots, S_m)$ of loops such that $S_1 \subset S_2 \subset \cdots \subset S_m$, $S_1, S_3, \dots \in P(S) \smallsetminus \mathcal{F}$, and $S_2, S_4, \dots \in \mathcal{F}$.

For $m, n \geq 1$, an $(m, n)^+$*superchain* is is a sequence $(T_1, T_2, \ldots, T_n)$ such that $T_1, T_3, \ldots$ are $m^+$chains, $T_2, T_4, \ldots$ are $m^-$chains, and the loops from $T_{i+1}$ are reachable from the loops of $T_i$ for $i = 1, 2, \ldots, n-1$. An $(m, n)^-$*superchain* is a sequence $(T_1, T_2, \ldots, T_n)$ such that $T_1, T_3, \ldots$ are $m^-$chains, $T_2, T_4, \ldots$ are $m^+$chains, and the loops from $T_{i+1}$ are reachable from the loops from $T_i$ for $i = 1, 2, \ldots, n-1$.

Now define the characteristics

$$\begin{array}{lll}
\mathrm{m}^+(\mathcal{M}) & =_{\text{def}} & \max\{m \mid \text{there exists an } m^+\text{chain in } \mathcal{M}\}, \\
\mathrm{m}^-(\mathcal{M}) & =_{\text{def}} & \max\{m \mid \text{there exists an } m^-\text{chain in } \mathcal{M}\}, \\
\mathrm{m}(\mathcal{M}) & =_{\text{def}} & \max\{\mathrm{m}^+(\mathcal{M}), \mathrm{m}^-(\mathcal{M})\}\}, \\
\mathrm{n}^+(\mathcal{M}) & =_{\text{def}} & \max\{n \mid \text{there exists an } (\mathrm{m}(\mathcal{M}), n)^+\text{superchain in } \mathcal{M}\}, \\
\mathrm{n}^-(\mathcal{M}) & =_{\text{def}} & \max\{n \mid \text{there exists an } (\mathrm{m}(\mathcal{M}), n)^-\text{superchain in } \mathcal{M}\}, \text{ and} \\
\mathrm{n}(\mathcal{M}) & =_{\text{def}} & \max\{\mathrm{n}^+(\mathcal{M}), \mathrm{n}^-(\mathcal{M})\}.
\end{array}$$

**Proposition 4.1** *Let $\mathcal{M} = (S, A, \delta, s_0, \mathcal{F})$ be a deterministic Muller automaton.*

1. $|\mathrm{m}^+(\mathcal{M}) - \mathrm{m}^-(\mathcal{M})| \leq 1$ *and* $|\mathrm{n}^+(\mathcal{M}) - \mathrm{n}^-(\mathcal{M})| \leq 1$.
2. $\mathrm{m}(\mathcal{M}) \cdot \mathrm{n}(\mathcal{M}) \leq |S|$.

The characteristics $\mathrm{m}^+(\mathcal{M})$, $\mathrm{m}^-(\mathcal{M})$, $\mathrm{n}^+(\mathcal{M})$, and $\mathrm{n}^-(\mathcal{M})$, are invariants of all automata accepting the same language:

**Theorem 4.2** *For deterministic Muller automata $\mathcal{M}$ and $\mathcal{M}'$, if $L_\omega(\mathcal{M}) = L_\omega(\mathcal{M}')$ then* $\mathrm{m}^+(\mathcal{M}) = \mathrm{m}^+(\mathcal{M}')$, $\mathrm{m}^-(\mathcal{M}) = \mathrm{m}^-(\mathcal{M}')$, $\mathrm{n}^+(\mathcal{M}) = \mathrm{n}^+(\mathcal{M}')$, *and* $\mathrm{n}^-(\mathcal{M}) = \mathrm{n}^-(\mathcal{M}')$.

Theorem 4.2 justifies the following definition. Let $L$ be an $\omega$-language and let $\mathcal{M}$ be a deterministic Muller automaton such that $L_\omega(\mathcal{M}) = L$. Then $\mathrm{m}^+(L) =_{\text{def}} \mathrm{m}^+(\mathcal{M})$, $\mathrm{m}^-(L) =_{\text{def}} \mathrm{m}^-(\mathcal{M})$, $\mathrm{n}^+(L) =_{\text{def}} \mathrm{n}^+(\mathcal{M})$, and $\mathrm{n}^-(L) =_{\text{def}} \mathrm{n}^-(\mathcal{M})$.

For $m, n \geq 1$, define the classes

$$\begin{array}{lll}
\mathrm{C}^n_m & =_{\text{def}} & \{L \mid \mathrm{m}(L) = m \wedge \mathrm{n}^+(L) = n - 1 \wedge \mathrm{n}^-(L) = n\}, \\
\mathrm{D}^n_m & =_{\text{def}} & \{L \mid \mathrm{m}(L) = m \wedge \mathrm{n}^+(L) = n \wedge \mathrm{n}^-(L) = n - 1\}, \\
\mathrm{E}^n_m & =_{\text{def}} & \{L \mid \mathrm{m}(L) = m \wedge \mathrm{n}^+(L) = \mathrm{n}^-(L) = n\}, \\
\hat{\mathrm{C}}^n_m & =_{\text{def}} & \{L \mid \mathrm{m}(L) < m \vee (\mathrm{m}(L) = m \wedge \mathrm{n}^+(L) < n)\}, \\
\hat{\mathrm{D}}^n_m & =_{\text{def}} & \{L \mid \mathrm{m}(L) < m \vee (\mathrm{m}(L) = m \wedge \mathrm{n}^-(L) < n)\}, \text{ and} \\
\hat{\mathrm{E}}^n_m & =_{\text{def}} & \{L \mid \mathrm{m}(L) < m \vee (\mathrm{m}(L) = m \wedge \mathrm{n}(L) \leq n)\}.
\end{array}$$

Some important relationships between these classes are given by the following theorem.

**Theorem 4.3** *Let $m, n \geq 1$.*

1. $\mathrm{D}^n_m = \text{co-}\mathrm{C}^n_m$ *and* $\hat{\mathrm{D}}^n_m = \text{co-}\hat{\mathrm{C}}^n_m$.
2. $\hat{\mathrm{C}}^n_m \cup \hat{\mathrm{D}}^n_m \subset \hat{\mathrm{E}}^n_m = \hat{\mathrm{C}}^{n+1}_m \cap \hat{\mathrm{D}}^{n+1}_m$.
3. $\hat{\mathrm{C}}^1_{m+1} \cap \hat{\mathrm{D}}^1_{m+1} = \bigcup_{n\geq 1} \hat{\mathrm{C}}^n_m = \bigcup_{n\geq 1} \hat{\mathrm{D}}^n_m = \bigcup_{n\geq 1} \hat{\mathrm{E}}^n_m = \{L \mid \mathrm{m}(L) \leq m\}$.
4. *The classes* $\mathrm{C}^n_m$, $\mathrm{D}^n_m$, *and* $\mathrm{E}^n_m$ *form a partition of the class of regular* $\omega$*-languages.*
5. $\mathrm{C}^n_m = \hat{\mathrm{C}}^n_m \setminus \hat{\mathrm{D}}^n_m$ *and* $\mathrm{D}^n_m = \hat{\mathrm{D}}^n_m \setminus \hat{\mathrm{C}}^n_m$.
6. $\mathrm{E}^n_m = \hat{\mathrm{E}}^n_m \setminus (\hat{\mathrm{C}}^n_m \cup \hat{\mathrm{D}}^n_m)$.

The following theorem shows the topological nature of the classes $\hat{\mathrm{C}}^n_m$, $\hat{\mathrm{D}}^n_m$ and $\hat{\mathrm{E}}^n_m$.

**Theorem 4.4** 1. *For $m, n \geq 1$, there hold $\hat{\mathrm{C}}^n_m = \widehat{\mathrm{C}^{\mathrm{n}}_{\mathrm{m}}}$, $\hat{\mathrm{D}}^n_m = \widehat{\mathrm{D}^{\mathrm{n}}_{\mathrm{m}}}$ and $\hat{\mathrm{E}}^n_m = \widehat{\mathrm{E}^{\mathrm{n}}_{\mathrm{m}}}$. Hence these classes are topological properties of regular $\omega$-languages.*

2. $\hat{\mathrm{C}}^1_1 = \{\emptyset\}$ *and* $\hat{\mathrm{D}}^1_1 = \{A^\omega\}$.
3. $\hat{\mathrm{C}}^2_1$ *is the class of regular open languages, and* $\hat{\mathrm{D}}^2_1$ *is the class of regular closed languages.*
4. $\hat{\mathrm{C}}^1_2$ *is the class of regular $G_\delta$-languages, and* $\hat{\mathrm{D}}^2_1$ *is the class of regular $F_\sigma$-languages.*
5. *For $m, n \geq 1$, the classes $\mathrm{C}^n_m$ and $\mathrm{D}^n_m$ are regular Wadge degrees.*
6. *For $n \geq 1$, the class $\mathrm{E}^n_1$ is a regular Wadge degree.*

From this theorem we know that the classes $\hat{\mathrm{C}}^n_m$ and $\hat{\mathrm{D}}^n_m$ for $m, n \geq 1$, and the classes $\hat{\mathrm{E}}^n_1$ for $n \geq 1$ are elementary topological properties of regular $\omega$-languages. So one has to look at the classes $\hat{\mathrm{E}}^n_m$ for $m \geq 2$ and $n \geq 1$, how they split into elementary topological properties of regular $\omega$-languages. For this reason define $\mathrm{d}^+S =_{\mathrm{def}} \{s \mid s \in S$ and an $(\mathrm{m}(\mathcal{M}), \mathrm{n}(\mathcal{M}))^+$superchain can be reached from $s\}$ and $\mathrm{d}^-S =_{\mathrm{def}} \{s \mid s \in S$ and an $(\mathrm{m}(\mathcal{M}), \mathrm{n}(\mathcal{M}))^-$superchain can be reached from $s\}$. Notice that $\mathrm{d}^+S \neq \emptyset$ implies $s_0 \in \mathrm{d}^+S$, that $\mathrm{d}^-S \neq \emptyset$ implies $s_0 \in \mathrm{d}^-S$, and that the defining condition $\mathrm{m}(\mathcal{M}) = m \wedge \mathrm{n}^+(\mathcal{M}) = \mathrm{n}^-(\mathcal{M}) = n$ of $\mathrm{E}^n_m$ is equivalent to $\mathrm{m}(\mathcal{M}) = m \wedge \mathrm{n}(\mathcal{M}) = n \wedge \mathrm{d}^+S \cap \mathrm{d}^-S \neq \emptyset$.

The *derivation* $\mathrm{d}\mathcal{M}$ of a Muller automaton $\mathcal{M} = (S, A, \delta, s_0, \mathcal{F})$ is defined as follows. If $\mathrm{m}(\mathcal{M}) = 1$ or $\mathrm{n}^+(\mathcal{M}) \neq \mathrm{n}^-(\mathcal{M})$ then $\mathrm{d}\mathcal{M} =_{\mathrm{def}} \mathcal{M}$. Otherwise $\mathrm{d}\mathcal{M}$ is defined as the Muller automaton $\mathrm{d}\mathcal{M} =_{\mathrm{def}} ((\mathrm{d}^+S \cap \mathrm{d}^-S) \cup \{s^+, s^-\}, A, \mathrm{d}\delta, s_0, \mathcal{F} \cap P(\mathrm{d}^+S \cap \mathrm{d}^-S))$ where $s^+, s^- \notin \mathrm{d}^+S \cap \mathrm{d}^-S$ and

$$\mathrm{d}\delta(s,a) =_{\mathrm{def}} \begin{cases} \delta(s,a), & \text{if } s, \delta(s,a) \in \mathrm{d}^+S \cap \mathrm{d}^-S, \\ s^+, & \text{if } s \in \mathrm{d}^+S \cap \mathrm{d}^-S \text{ and } \delta(s,a) \in \mathrm{d}^+S \setminus \mathrm{d}^-S, \\ s^-, & \text{if } s \in \mathrm{d}^+S \cap \mathrm{d}^-S \text{ and } \delta(s,a) \notin \mathrm{d}^+S, \\ s^+, & \text{if } s = s^+, \\ s^-, & \text{if } s = s^-. \end{cases}$$

For $r \geq 1$, define the $r$-th derivation of $\mathcal{M}$ by $\mathrm{d}^0\mathcal{M} =_{\mathrm{def}} \mathcal{M}$ and $\mathrm{d}^{r+1}\mathcal{M} =_{\mathrm{def}} \mathrm{d}(\mathrm{d}^r\mathcal{M})$.

**Theorem 4.5** *For deterministic Muller automata $\mathcal{M}$ and $\mathcal{M}'$, if $L_\omega(\mathcal{M}) = L_\omega(\mathcal{M}')$ then $L_\omega(\mathrm{d}\mathcal{M}) = L_\omega(\mathrm{d}\mathcal{M}')$, i.e., the derivation is an invariant of all automata accepting the same language.*

Theorem 4.5 justifies the following definition. Let $L$ be an $\omega$-language and let $\mathcal{M}$ be an deterministic Muller automaton such that $L_\omega(\mathcal{M}) = L$. Then $\mathrm{d}(L) =_{\mathrm{def}} L_\omega(\mathrm{d}\mathcal{M})$. For $\mathcal{C} \subseteq \mathcal{R}$ define $\mathrm{d}(\mathcal{C}) =_{\mathrm{def}} \{\mathrm{d}(L) \mid L \in \mathcal{C}\}$ and $\mathrm{d}^{-1}(\mathcal{C}) =_{\mathrm{def}} \{L \mid \mathrm{d}(L) \in \mathcal{C}\}$.

**Theorem 4.6** 1. *If $L \in \{\mathrm{C}^n_m \mid m, n \geq 1\} \cup \{\mathrm{D}^n_m \mid m, n \geq 1\} \cup \{\mathrm{E}^n_1 \mid n \geq 1\}$ then $\mathrm{d}(L) = L$.*

2. *If $L \in \{\mathrm{E}^n_m \mid m \geq 2, n \geq 1\}$ then $\mathrm{d}(L) \in \mathrm{C}^1_m \cap \mathrm{D}^1_m$.*

For a class $\mathcal{C} \subseteq \mathcal{R}$ and $m, n \geq 1$ we define $\mathrm{E}^n_m\mathcal{C} =_{\mathrm{def}} \{L \mid L \in \mathrm{E}^n_m \wedge \mathrm{d}(L) \in \mathcal{C}\} = \mathrm{E}^n_m \cap \mathrm{d}^{-1}(\mathcal{C})$. Now the family $\mathcal{T}$ of all regular Wadge degrees can be characterized as follows.

**Theorem 4.7** $\mathcal{T} = \{\mathrm{E}^{n_1}_{m_1}\mathrm{E}^{n_2}_{m_2}\ldots\mathrm{E}^{n_{r-1}}_{m_{r-1}}\mathrm{C}^{n_r}_{m_r} \mid r \geq 1, m_1 > m_2 > \cdots > m_r \geq 1, n_1, n_2, \ldots, n_r \geq 1\} \cup \{\mathrm{E}^{n_1}_{m_1}\mathrm{E}^{n_2}_{m_2}\ldots\mathrm{E}^{n_{r-1}}_{m_{r-1}}\mathrm{D}^{n_r}_{m_r} \mid r \geq 1, m_1 > m_2 > \cdots > m_r \geq 1, n_1, n_2, \ldots, n_r \geq 1\} \cup \{\mathrm{E}^{n_1}_{m_1}\mathrm{E}^{n_2}_{m_2}\ldots\mathrm{E}^{n_{r-1}}_{m_{r-1}}\mathrm{E}^{n_r}_{1} \mid r \geq 1, m_1 > m_2 > \cdots > m_{r-1} > 1, n_1, n_2, \ldots, n_r \geq 1\}$.

For our decision algorithms the following theorem will be important.

**Theorem 4.8** *For $m \geq 2$ and $n \geq 1$, if $\mathcal{C} \subseteq \mathrm{C}^1_m \cap \mathrm{D}^1_m$ then $\widehat{\mathrm{E}^n_m\mathcal{C}} = \hat{\mathrm{C}}^n_m \cup \hat{\mathrm{D}}^n_m \cup \mathrm{E}^n_m\hat{\mathcal{C}}$.*

An interesting relationship between the structure of $\mathcal{T}$ and $\hat{\mathcal{T}}$, resp., and the ordinal numbers below $\omega^\omega$ should be mentioned. It is well-known that every non-zero ordinal $\alpha < \omega^\omega$ can be presented in the form

$\alpha = n_1 \cdot \omega^{m_1} + n_2 \cdot \omega^{m_2} + \cdots + n_r \cdot \omega^{m_r}$ where $r \geq 1$, $m_1 > m_2 > \cdots > m_r \geq 0$ and $n_1, n_2, \ldots, n_r \geq 1$ (*).

This gives a bijection between the ordinals below $\omega^\omega$ and the classes of type $\mathrm{E}^{n_1}_{m_1}\mathrm{E}^{n_2}_{m_2}\ldots\mathrm{E}^{n_{r-1}}_{m_{r-1}}\mathrm{C}^{n_r}_{m_r}$. If $\alpha$ is presented in the form (*) then we define $\mathcal{R}'_\alpha =_{\mathrm{def}} \mathrm{E}^{n_1}_{m_1+1}\mathrm{E}^{n_2}_{m_2+1}\ldots\mathrm{E}^{n_{r-1}}_{m_{r-1}+1}\mathrm{C}^{n_r+1}_{m_r+1}$. Then co-$\mathcal{R}'_\alpha = \mathrm{E}^{n_1}_{m_1+1}\mathrm{E}^{n_2}_{m_2+1}\ldots\mathrm{E}^{n_{r-1}}_{m_{r-1}+1}\mathrm{D}^{n_r+1}_{m_r+1}$. For $\alpha = n_1 \cdot \omega^{m_1} + n_2 \cdot \omega^{m_2} + \cdots + n_{r-1} \cdot \omega^{m_{r-1}} + n_r$ where $r \geq 1$, $m_1 > m_2 > \cdots > m_{r-1} \geq 1$ and $n_1, n_2, \ldots, n_r \geq 1$ we obtain $\widetilde{\mathcal{R}}_{\alpha+1} = \mathrm{E}^{n_1}_{m_1+1}\mathrm{E}^{n_2}_{m_2+1}\ldots\mathrm{E}^{n_{r-1}}_{m_{r-1}+1}\mathrm{E}^{n_r+1}_{1}$ where $\widetilde{\mathcal{R}}_{\alpha+1} =_{\mathrm{def}} (\mathcal{R}_{\alpha+1} \cap \text{co-}\mathcal{R}_{\alpha+1}) \setminus (\mathcal{R}_\alpha \cup \text{co-}\mathcal{R}_\alpha)$ and $\mathcal{R}_\alpha =_{\mathrm{def}} \widehat{\mathcal{R}'_\alpha}$. Thus we have $\mathcal{T} = \{\mathcal{R}'_\alpha, \text{co-}\mathcal{R}'_\alpha, \widetilde{\mathcal{R}}_{\alpha+1} \mid \alpha < \omega^\omega\}$ and $\hat{\mathcal{T}} = \{\mathcal{R}_\alpha, \text{co-}\mathcal{R}_\alpha, \mathcal{R}_{\alpha+1} \cap \text{co-}\mathcal{R}_{\alpha+1} \mid \alpha < \omega^\omega\}$.

There holds $\mathcal{R}_\alpha \cup \text{co-}\mathcal{R}_\alpha \subseteq \mathcal{R}_{\alpha+1} \cap \text{co-}\mathcal{R}_{\alpha+1}$ for $\alpha < \omega^\omega$. Hence, $(\mathcal{T}; \leq_{\mathrm{w}})$ and $(\hat{\mathcal{T}}; \subseteq)$ have a quasi-linear structure.

Interestingly, w.r.t. many-one reductions between regular $\omega$-languages, finite state transducers are as powerful as arbitrary continuous functions. For $L, L' \subseteq A^\omega$, we write $L \leq_{\mathrm{fa}} L'$ if there exists a function $f : A^\omega \to A^\omega$ computed by a finite state transducer such that $\xi \in L \leftrightarrow f(\xi) \in L'$.

**Theorem 4.9** *For all regular $\omega$-languages $L$ and $L'$, there holds $L \leq_{\mathrm{fa}} L'$ if and only if $L \leq_{\mathrm{w}} L'$.*

Finally, a "part" of a Muller automaton $\mathcal{M}$ accepts an $\omega$-language which is Wadge-reducible to $L_\omega(\mathcal{M})$.

**Proposition 4.10** *If $\mathcal{M} = (S, A, \delta, s_0, \mathcal{F})$ is a deterministic Muller automaton and $x \in A^*$ then $L_\omega((S, A, \delta, \delta(s_0, x), \mathcal{F})) \leq_{\mathrm{w}} L_\omega(\mathcal{M})$.*

# 5 Upper Bounds for Deterministic Automata

Let $\mathcal{M} = (S, A, \delta, s_0, \mathcal{E})$ be a deterministic $\omega$-automaton of some type, where $\mathcal{E}$ describes an acceptance condition for this type. Obviously, $\mathcal{M}$ is equivalent to the deterministic Muller automaton $\tilde{\mathcal{M}} = (S, A, \delta, s_0, \{S' \mid S' \text{ satisfies condition } \mathcal{E}\})$, i.e., there holds $L_\omega(\mathcal{M}) = L_\omega(\tilde{\mathcal{M}})$. In fact, deterministic $\omega$-automata of arbitrary types can be considered as succinct presentations of deterministic Muller automata. Hence the definitions of chains, superchains, and the characteristics $\mathrm{m}^+$, $\mathrm{m}^-$, $\mathrm{n}^+$, and $\mathrm{n}^-$ apply also to these types of $\omega$-automata.

For $X \in \{\mathrm{M}, \mathrm{R}, \mathrm{S}, \mathrm{P}, \mathrm{B}\}$, let

$$\begin{aligned}\mathrm{Chain}_{\mathrm{DX}} =_{\mathrm{def}} & \{(\mathcal{M}, m, s, +) \mid \mathcal{M} \text{ is a deterministic } X\text{-automaton}, m \geq 1, \\ & \qquad \text{and } s \text{ belongs to an } m^+\text{chain of } \mathcal{M}\} \cup \\ & \{(\mathcal{M}, m, s, -) \mid \mathcal{M} \text{ is a deterministic } X\text{-automaton}, m \geq 1, \\ & \qquad \text{and } s \text{ belongs to an } m^-\text{chain of } \mathcal{M}\}\end{aligned}$$

and

$$\begin{aligned}\mathrm{Super}_{\mathrm{DX}} =_{\mathrm{def}} & \{(\mathcal{M}, m, n, s, +) \mid \mathcal{M} \text{ is a deterministic } X\text{-automaton}, m, n \geq 1, \\ & \qquad \text{and an } (m,n)^+\text{superchain of } \mathcal{M} \text{ is reachable from } s\} \cup \\ & \{(\mathcal{M}, m, n, s, -) \mid \mathcal{M} \text{ is a deterministic } X\text{-automaton}, m, n \geq 1, \\ & \qquad \text{and an } (m,n)^-\text{superchain of } \mathcal{M} \text{ is reachable from } s\}\end{aligned}$$

We observe

**Proposition 5.1** *Let $\mathcal{M}$ be a deterministic X-automaton, and let $m, n \geq 1$.*

1. $\mathrm{m}^+(\mathcal{M}) \geq m \iff$ *there exists an* $s \in S$ *such that* $(\mathcal{M}, m, s, +) \in \mathrm{Chain}_{\mathrm{DX}}$.
2. $\mathrm{m}^-(\mathcal{M}) \geq m \iff$ *there exists an* $s \in S$ *such that* $(\mathcal{M}, m, s, -) \in \mathrm{Chain}_{\mathrm{DX}}$.
3. $\mathrm{n}^+(\mathcal{M}) \geq n \iff (\mathcal{M}, \mathrm{m}(\mathcal{M}), n, s_0, +) \in \mathrm{Super}_{\mathrm{DX}}$.
4. $\mathrm{n}^-(\mathcal{M}) \geq n \iff (\mathcal{M}, \mathrm{m}(\mathcal{M}), n, s_0, -) \in \mathrm{Super}_{\mathrm{DX}}$.

In what follows, we will make use of the fact that the class NL is closed under complementation and consequently, the NL-query-hierarchy collapses to NL. That is, if we use NL-oracles during an NL-computation then this can be simulated by an NL-computation without oracle. So it will be sufficient to present $\mathrm{L}^{\mathrm{NL}}$-algorithms or $\mathrm{NL}^{\mathrm{NL}}$-algorithms for the problems in question. This applies also to NL-computations with a fixed additional oracle $A$. That means, for example, $\mathrm{L}^{\mathrm{NL}^A} = \mathrm{NL}^{\mathrm{NL}^A} = \mathrm{NL}^A$. We should hint to some subtlety: When using oracles during an NL-computation, the oracle queries have to have the form $(x, z)$ where $x$ is the input of the base computation and $|z| \leq c \cdot \log |x|$ for some constant $c > 0$. For L-computation with oracle, there is no such restriction.

It turns out that, for deciding the topological degrees, the complexity of Chain plays a central role. Knowing its complexity, the complexity of the topological degrees follows in a uniform way.

**Lemma 5.2** *Let* $X \in \{\mathrm{M}, \mathrm{R}, \mathrm{S}, \mathrm{P}, \mathrm{B}\}$.

1. $\mathrm{Super}_{\mathrm{DX}} \in \mathrm{NL}^{\mathrm{Chain}_{\mathrm{DX}}}$.
2. *There exists an* $\mathrm{L}^{\mathrm{NL}^{\mathrm{Chain}_{\mathrm{DX}}}}$*-algorithm which, given a deterministic X-automaton* $\mathcal{M}$*, computes the characteristics* $\mathrm{m}^+(\mathcal{M})$*,* $\mathrm{m}^-(\mathcal{M})$*,* $\mathrm{n}^+(\mathcal{M})$*, and* $\mathrm{n}^-(\mathcal{M})$.
3. *There exists an* $\mathrm{L}^{\mathrm{NL}^{\mathrm{Chain}_{\mathrm{DX}}}}$*-algorithm which, given a deterministic X-automaton* $\mathcal{M}$*, computes* $\mathrm{d}\mathcal{M}$.
4. *For every* $\mathcal{C} \subseteq \mathcal{R}$*, if* $(\mathcal{C})_{\mathrm{DX}} \in \mathrm{NL}^{\mathrm{Chain}_{\mathrm{DX}}}$ *then* $(\mathrm{d}^{-1}\mathcal{C})_{\mathrm{DX}} \in \mathrm{NL}^{\mathrm{Chain}_{\mathrm{DX}}}$.

**Proof.** 1. For a deterministic $X$-automaton $\mathcal{M} = (S, A, \delta, s_0, \mathcal{E})$, $m, n \geq 1$, and $s \in S$ there holds

$$\begin{array}{rcl} (\mathcal{M}, m, n, s, +) \in \mathrm{Super}_{\mathrm{DX}} & \iff & \text{there exist } s_1, s_2, \ldots, s_n \in S \text{ such that} \\ & & s_i \text{ is reachable from } s_{i-1}, \text{ for } i = 1, 2, \ldots, m, \\ & & (\mathcal{M}, m, s_1, +), (\mathcal{M}, m, s_3, +), (\mathcal{M}, m, s_5, +), \cdots \in \mathrm{Chain}_{\mathrm{DX}}, \text{ and} \\ & & (\mathcal{M}, m, s_2, -), (\mathcal{M}, m, s_4, -), (\mathcal{M}, m, s_6, -), \cdots \in \mathrm{Chain}_{\mathrm{DX}}, \end{array}$$

and analogously for $(\mathcal{M}, m, n, s, -)$. This gives an $\mathrm{NL}^{\mathrm{Chain}_{\mathrm{DX}}}$-algorithm for $\mathrm{Super}_{\mathrm{DX}}$.

2. Easy by using Proposition 5.1 and Lemma 5.2.1.

3. For a given deterministic $X$-automaton $\mathcal{M} = (S, A, \delta, s_0, \mathcal{E})$, if $\mathrm{m}^+(\mathcal{M}) = 1$ or $\mathrm{n}^+(\mathcal{M}) \neq \mathrm{n}^+(\mathcal{M})$ then $\mathcal{M}$ is put out. Otherwise, the automaton $((\mathrm{d}^+S \cap \mathrm{d}^-S) \cup \{s^+, s^-\}, A, \mathrm{d}\delta, s_0, \mathcal{E}')$ has to be generated where $\mathcal{E}'$ is the restriction of $\mathcal{E}$ to subsets of $\mathrm{d}^+S \cap \mathrm{d}^-S$. The main problem here is to decide $s \in \mathrm{d}S^+$ and $s \in \mathrm{d}S^-$ for given $s \in S$. But this is equivalent to $(\mathcal{M}, \mathrm{m}(\mathcal{M}), \mathrm{n}(\mathcal{M}), s, +) \in \mathrm{Super}_{\mathrm{DX}}$ and $(\mathcal{M}, \mathrm{m}(\mathcal{M}), \mathrm{n}(\mathcal{M}), s, -) \in \mathrm{Super}_{\mathrm{DX}}$, resp., which can be checked by $\mathrm{NL}^{\mathrm{Chain}_{\mathrm{DX}}}$-queries.

4. Let $\mathcal{C} \subseteq \mathcal{R}$ be such that $(\mathcal{C})_{\mathrm{DX}} \in \mathrm{NL}^{\mathrm{Chain}_{\mathrm{DX}}}$. By the definition we have $\mathcal{M} \in (\mathrm{d}^{-1}\mathcal{C})_{\mathrm{DX}} \Leftrightarrow \mathrm{d}\mathcal{M} \in (\mathcal{C})_{\mathrm{DX}}$. By Lemma 5.2.3, an $\mathrm{L}^{\mathrm{NL}^{\mathrm{Chain}_{\mathrm{DX}}}}$-algorithm can produce $\mathrm{d}\mathcal{M}$ from $\mathcal{M}$, and then one more $\mathrm{NL}^{\mathrm{Chain}_{\mathrm{DX}}}$-query is asked to find out whether $\mathrm{d}\mathcal{M}$ is in $(\mathcal{C})_{\mathrm{DX}}$. This results in an $\mathrm{L}^{\mathrm{NL}^{\mathrm{Chain}_{\mathrm{DX}}}}$-algorithm to accept $(\mathrm{d}^{-1}\mathcal{C})_{\mathrm{DX}}$, hence $(\mathrm{d}^{-1}\mathcal{C})_{\mathrm{DX}} \in \mathrm{NL}^{\mathrm{Chain}_{\mathrm{DX}}}$. $\square$

**Theorem 5.3** *For* $X \in \{\mathrm{M}, \mathrm{R}, \mathrm{S}, \mathrm{P}, \mathrm{B}\}$ *and* $\mathcal{C} \in \mathcal{T}$*, the problems* $(\hat{\mathcal{C}})_{\mathrm{DX}}$ *and* $(\mathcal{C})_{\mathrm{DX}}$ *are in* $\mathrm{NL}^{\mathrm{Chain}_{\mathrm{DX}}}$.

**Proof.** By induction. We start with the classes $\hat{\mathrm{C}}^n_m$, $\hat{\mathrm{D}}^n_m$, and $\hat{\mathrm{E}}^n_m$. For $\hat{\mathrm{C}}^n_m$, the definition yields the equivalence $\mathcal{M} \in (\hat{\mathrm{C}}^n_m)_{\mathrm{DX}} \iff \mathrm{m}(\mathcal{M}) < m \vee (\mathrm{m}(\mathcal{M}) = m \wedge \mathrm{n}^+(\mathcal{M}) < n)$. Using this and Lemma 5.2.2 we can check $\mathcal{M} \in (\hat{\mathrm{C}}^n_m)_{\mathrm{DX}}$ by an L-algorithm with an $\mathrm{NL}^{\mathrm{Chain}_{\mathrm{DX}}}$-oracle. The argument for

$(\hat{\mathrm{D}}^n_m)_{\mathrm{DX}}$ and $(\hat{\mathrm{E}}^n_m)_{\mathrm{DX}}$ is completely analogous. By the Statements 5 and 6 of Theorem 4.3 we obtain $(\mathrm{C}^n_m)_{\mathrm{DX}}, (\mathrm{D}^n_m)_{\mathrm{DX}}, (\mathrm{E}^n_m)_{\mathrm{DX}} \in \mathrm{L}^{\mathrm{NL}^{\mathrm{Chain}_{\mathrm{DX}}}} = \mathrm{NL}^{\mathrm{Chain}_{\mathrm{DX}}}$.

For the induction step, let $m \geq 2$, $n \geq 1$, and $\mathrm{E}^n_m\mathcal{C} \in \mathcal{T}$. Consequently, $\mathcal{C} \subseteq \mathrm{C}^1_m \cap \mathrm{D}^1_m$. By Theorem 4.8 and the respective definitions we obtain

$$\mathcal{M} \in (\widehat{\mathrm{E}^n_m\mathcal{C}})_{\mathrm{DX}} \Longleftrightarrow \mathcal{M} \in (\hat{\mathrm{C}}^n_m)_{\mathrm{DX}} \vee \mathcal{M} \in (\hat{\mathrm{D}}^n_m)_{\mathrm{DX}} \vee (\mathcal{M} \in (\mathrm{E}^n_m)_{\mathrm{DX}} \wedge \mathcal{M} \in (\mathrm{d}^{-1}\hat{\mathcal{C}})_{\mathrm{DX}}) \text{ and}$$

$$\mathcal{M} \in (\mathrm{E}^n_m\mathcal{C})_{\mathrm{DX}} \Longleftrightarrow \mathcal{M} \in (\mathrm{E}^n_m)_{\mathrm{DX}} \wedge \mathcal{M} \in (\mathrm{d}^{-1}\mathcal{C})_{\mathrm{DX}}.$$

By the assumption of our induction we know that $(\hat{\mathcal{C}})_{\mathrm{DX}}$ and $(\mathcal{C})_{\mathrm{DX}}$ are in $\mathrm{NL}^{\mathrm{Chain}_{\mathrm{DX}}}$. By Lemma 5.2.4 also $(\mathrm{d}^{-1}\hat{\mathcal{C}})_{\mathrm{DX}}$ and $(\mathrm{d}^{-1}\mathcal{C})_{\mathrm{DX}}$ are in $\mathrm{NL}^{\mathrm{Chain}_{\mathrm{DX}}}$. Consequently, $(\widehat{\mathrm{E}^n_m\mathcal{C}})_{\mathrm{DX}}$ and $(\mathrm{E}^n_m\mathcal{C})_{\mathrm{DX}}$ are in $\mathrm{NL}^{\mathrm{Chain}_{\mathrm{DX}}}$. $\square$

# 6 Deterministic Muller Automata

In this section, let $\mathcal{M} = (S, A, \delta, s_0, \mathcal{F})$ be a deterministic Muller automaton where $\mathcal{F} = \{S_1, S_2, \ldots, S_r\}$.

We define a few problems needed for our algorithm. Let $m, n \geq 1$.

$$\begin{array}{rll}
(\mathcal{M}, i, j) \in \mathrm{Subset} & \Leftrightarrow_{\mathrm{def}} & S_i \subset S_j \\
(\mathcal{M}, i, j) \in \mathrm{Subseteq} & \Leftrightarrow_{\mathrm{def}} & S_i \subseteq S_j \\
(\mathcal{M}, s, s') \in \mathrm{Reach} & \Leftrightarrow_{\mathrm{def}} & \exists(x \in A^*)(\delta(s, x) = s') \\
(\mathcal{M}, i) \in \mathrm{Loop} & \Leftrightarrow_{\mathrm{def}} & S_i \text{ is a loop of } \mathcal{M} \\
(\mathcal{M}, i, j) \in \mathrm{Between}^+ & \Leftrightarrow_{\mathrm{def}} & \exists k(S_k \text{ is a loop of } \mathcal{M} \text{ and } S_i \subset S_k \subset S_j) \\
(\mathcal{M}, i, j) \in \mathrm{Between}^- & \Leftrightarrow_{\mathrm{def}} & (\mathcal{M}, i) \notin \mathrm{Between}^+ \wedge \exists S'(S' \text{ is a loop of } \mathcal{M} \text{ and } S_i \subseteq S' \subseteq S_j) \\
(\mathcal{M}, i) \in \mathrm{Outside}^+ & \Leftrightarrow_{\mathrm{def}} & \exists k(S_k \text{ is a loop of } \mathcal{M} \text{ and } S_i \subset S_k) \\
(\mathcal{M}, i) \in \mathrm{Outside}^- & \Leftrightarrow_{\mathrm{def}} & (\mathcal{M}, i) \notin \mathrm{Outside}^+ \wedge \exists S'(S' \text{ is a loop of } \mathcal{M} \text{ and } S_i \subseteq S') \\
(\mathcal{M}, i) \in \mathrm{Inside}^+ & \Leftrightarrow_{\mathrm{def}} & \exists k(S_k \text{ is a loop of } \mathcal{M} \text{ and } S_k \subset S_i) \\
(\mathcal{M}, i) \in \mathrm{Inside}^- & \Leftrightarrow_{\mathrm{def}} & (\mathcal{M}, i) \notin \mathrm{Inside}^+ \wedge \exists S'(S' \text{ is a loop of } \mathcal{M} \text{ and } S' \subseteq S_i)
\end{array}$$

**Lemma 6.1** *1. The problems* Subset, Subseteq, Reach, Loop, $\mathrm{Between}^+$, $\mathrm{Between}^-$, $\mathrm{Outside}^+$, $\mathrm{Outside}^-$, $\mathrm{Inside}^+$, *and* $\mathrm{Inside}^-$ *are in* NL.

*2.* $\mathrm{Chain}_{\mathrm{DM}} \in \mathrm{NL}$.

**Proof.** 1. It is evident that Subset, Subseteq are in L and that Reach is in NL.

For Loop we use the obvious equivalences

$$\begin{array}{rll}
(\mathcal{M}, i) \in \mathrm{Loop} & \Longleftrightarrow & \text{there exist } s \in S_i \text{ and } x, z \in A^* \text{ such that } \delta(s_0, x) = s,\ \delta(s, z) = s, \\
& & \text{and } \overline{\delta}(s, z) = S_i \\
& \Longleftrightarrow & \text{there exist } x, u = u_1u_2\ldots u_l \in A^* \text{ such that } \delta(s_0, x) = s_1,\ \delta(s_j, u_j) = s_{j+1} \text{ for} \\
& & i = 1, 2, \ldots, l-1,\ \delta(s_l, u_l) = s_1, \text{ and } \overline{\delta}(s_1, u) \subseteq S_i \text{ where } s_1, s_2, \ldots, s_l \text{ are the} \\
& & \text{elements of } S_i \text{ in the order they appear on the input tape.}
\end{array}$$

At the beginning the algorithm guesses nondeterministically an $x \in A^*$ (without storing it) and simulates $\mathcal{M}$ with start state $s_0$ on $x$ until $\mathcal{M}$ reaches the state $s_1$. Now it guesses nondeterministically an $u_1 \in A^*$ (without storing it) and simulates $\mathcal{M}$ with start state $s_1$ on $u_1$ until $\mathcal{M}$ reaches the state $s_2$ where it is checked whether every reached state is in $S_i$. Then it proceeds in the same way with $u_2, u_3, \ldots, u_l$ (set $s_{l+1} =_{\mathrm{def}} s_1$). If all checks are positive then the algorithm accepts. This is clearly an NL-algorithm.

Easy $\mathrm{L}^{\mathrm{NL}}$-algorithms for $\mathrm{Between}^+$, $\mathrm{Outside}^+$, and $\mathrm{Inside}^+$ are given just by their definitions.

To decide $\mathrm{Between}^-$, the condition $(\mathcal{M}, i) \notin \mathrm{Between}^+$ is in co-NL, and the condition $\exists S'(S'$ is a loop in $\mathcal{M}$ and $S_i \subseteq S' \subseteq S_j)$ can be verified by an NL-algorithm which works just as the one for Loop but it checks whether every reached state is in $S_j$ instead of $S_i$. Similar algorithms can be designed for $\mathrm{Outside}^-$ and $\mathrm{Inside}^-$.

2. For $\mathrm{Chain}_{\mathrm{DM}}$ we consider the following obvious equivalences

$$
\begin{aligned}
(\mathcal{M}, 2m+1, s, +) \in \mathrm{Chain}_{\mathrm{DM}} \iff & \text{ there exist } i_1, i_2, \ldots, i_{m+1} \text{ such that } S_{i_1}, S_{i_2}, \ldots, S_{i_{m+1}} \text{ are loops,} \\
& s \in S_{i_{m+1}}, \text{ and there exist loops } R_1, R_2, \ldots, R_m \in P(S) \smallsetminus \mathcal{F} \\
& \text{such that } S_{i_1} \subset R_1 \subset S_{i_2} \subset R_2 \subset \cdots \subset R_m \subset S_{i_{m+1}} \\
\iff & \text{ there exist } i_1, i_2, \ldots, i_m, j_1, j_2, \ldots, j_m \text{ such that } s \in S_{j_m}, \\
& \text{and for all } \mu = 1, 2, \ldots, m \text{ there holds:} \\
& \text{– } S_{i_\mu} \text{ and } S_{j_\mu} \text{ are loops and } S_{j_\mu} \subseteq S_{i_{\mu+1}} \text{ if } \mu < m, \\
& \text{– there is no loop } S_k \text{ such that } S_{i_\mu} \subset S_k \subset S_{j_\mu}, \text{ and} \\
& \text{– there is a loop } S' \text{ such that } S_{i_\mu} \subset S' \subset S_{j_\mu} \\
\iff & \text{ there exist } i_1, i_2, \ldots, i_m, j_1, j_2, \ldots, j_m \text{ such that } s \in S_{j_m}, \\
& \text{and for all } \mu = 1, 2, \ldots, m \text{ there holds:} \\
& (\mathcal{M}, i_\mu) \in \mathrm{Loop}, (\mathcal{M}, j_\mu) \in \mathrm{Loop}, (\mathcal{M}, j_\mu, i_{\mu+1}) \in \mathrm{Subset} \text{ if } \mu < m, \\
& \text{and } (\mathcal{M}, i_\mu, j_\mu) \in \mathrm{Between}^-.
\end{aligned}
$$

From the latter equivalence we can easily get an $\mathrm{NL}^{\mathrm{NL}}$-algorithm to accept $(\mathcal{M}, 2m+1, s, +) \in \mathrm{Chain}_{\mathrm{DM}}$. For $(\mathcal{M}, 2m, s, +) \in \mathrm{Chain}_{\mathrm{DM}}$, $(\mathcal{M}, 2m+1, s, -) \in \mathrm{Chain}_{\mathrm{DM}}$, and $(\mathcal{M}, 2m, s, -) \in \mathrm{Chain}_{\mathrm{DM}}$ and we obtain similar equivalences and algorithms. □

Now we prove the main result of this section.

**Theorem 6.2** *For $\mathcal{C} \in \mathcal{T}$, the problems $(\hat{\mathcal{C}})_{\mathrm{DM}}$ and $(\mathcal{C})_{\mathrm{DM}}$ are* NL*-complete.*

**Proof.** The membership of these problems to NL follows immediately from Theorem 5.3 and Lemma 6.1.2.

For the hardness in NL we reduce the complement of an NL-complete version of the graph accessibility to our problems (which completes the proof because of co-NL = NL).

The graph accessibility problem remains NL-complete if the instances are restricted to directed acyclic graphs $(V, E)$ with one source $s$ and two sinks $t^+$ and $t^-$ where all non-sinks have outdegree 2. Let GAP$'$ denote the set of such instances $(V, E, s, t^+, t^-)$ such that there exists a path from $s$ to $t^+$.

We start with deterministic Muller automata with input alphaber $A$. Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be deterministic Muller automata with input alphabet $A$ such that $L_\omega(\mathcal{M}_1) \in \mathcal{C}$ and $\mathrm{m}(\mathcal{M}_1) < \mathrm{m}(\mathcal{M}_2)$. Note that this implies $L_\omega(\mathcal{M}_2) \notin \hat{\mathcal{C}}$.

Given an instance $(V, E, s, t^+, t^-)$ to GAP$'$, we construct from this and the automata $\mathcal{M}_1$ and $\mathcal{M}_2$ a new deterministic Muller automaton $\mathcal{M}$ as follows. The initial part of $\mathcal{M}$ is the graph $(V, E)$, where for every non-sink $v \in V$ one outgoing edge is used for the input symbol 0, and the other outgoing edge is used for all other input symbols. The sink $t^-$ is identified with the initial state of $\mathcal{M}_1$, and the sink $t^+$ is identified with the initial state of $\mathcal{M}_2$. The initial state of $\mathcal{M}$ is $s$, and final sets of $\mathcal{M}$ are the final sets of $\mathcal{M}_1$ and the final sets of $\mathcal{M}_2$.

If $(V, E, s, t^+, t^-)$ is not in GAP$'$ then there is no path from $s$ to $t^+$ and hence the loops in $\mathcal{M}_2$ are not loops in $\mathcal{M}$ (because the cannot be reached from the initial state of $\mathcal{M}$). If there is no path from $s$ to $t^+$ then there is a path from $s$ to $t^-$. Hence the loops of $\mathcal{M}$ are just the loops of $\mathcal{M}_1$. Consequently, $L_\omega(\mathcal{M})$ and $L_\omega(\mathcal{M}_1)$ are in the same Wadge degree, i.e., $L_\omega(\mathcal{M}) \in \mathcal{C}$.

If $(V, E, s, t^+, t^-)$ is in GAP$'$ then there is a path from $s$ to $t^+$ and hence the loops in $\mathcal{M}_2$ are also loops in $\mathcal{M}$. Also $t^-$ can be reachable from $s$, and hence the loops of $\mathcal{M}_1$ can also be loops of $\mathcal{M}$. However, because of $\mathrm{m}(\mathcal{M}_1) < \mathrm{m}(\mathcal{M}_2)$ the chains in $\mathcal{M}_1$ do not contribute to the superchains in $\mathcal{M}$. Consequently, $L_\omega(\mathcal{M})$ and $L_\omega(\mathcal{M}_2)$ are in the same Wadge degree, i.e., $L_\omega(\mathcal{M}) \notin \hat{\mathcal{C}}$. □

# 7 Deterministic Mostowski and Büchi Automata

For these types of automata we can prove the same results as for deterministic Muller automata.

**Lemma 7.1** *The problems* $\mathrm{Chain}_{\mathrm{DP}}$ *and* $\mathrm{Chain}_{\mathrm{DB}}$ *are in* NL.

**Proof.** Since Büchi automata are special Mostowski automata, it is sufficient to prove the lemma for Mostowski automata. Let $\mathcal{M} = (S, A, \delta, s_0, \{(E_1, F_1), (E_2, F_2), \ldots, (E_r, F_r)\})$ be a deterministic Mostowski automaton auch that $E_1 \subseteq F_1 \subseteq E_2 \subseteq F_2 \subseteq \cdots \subseteq E_r \subseteq F_r$. Because of these inclusions, the condition $\bigwedge_{i=1}^{r}(S' \cap E_i \neq \emptyset \vee S' \cap F_i = \emptyset)$ is equivalent to the condition $\bigvee_{i=1}^{r+1}(S' \cap E_i \neq \emptyset \wedge S' \cap F_{i-1} = \emptyset)$, where $F_0 =_{\text{def}} \emptyset$ and $E_{r+1} =_{\text{def}} S$.

Now we obtain the following equivalences:

$$
\begin{aligned}
(\mathcal{M}, m, s, +) \in (\text{Chain})_{\text{DP}} &\iff s \text{ belongs to an } m^{+}\text{chain in } \mathcal{M} \\
&\iff s \text{ belongs to an } m^{+}\text{chain in } \mathcal{M}' \\
&\iff \text{there exist loops } S_1 \subset S_2 \subset \cdots \subset S_m \text{ such that} \\
&\qquad \textstyle\bigvee_{i=1}^{r}(S_\mu \cap E_i = \emptyset \wedge S_\mu \cap F_i \neq \emptyset) \text{ for } \mu = 1, 3, 5, \ldots \text{ and} \\
&\qquad \textstyle\bigwedge_{i=1}^{r}(S_\mu \cap E_i \neq \emptyset \vee S_\mu \cap F_i = \emptyset) \text{ for } \mu = 2, 4, 6, \ldots \\
&\iff \text{there exist loops } S_1 \subset S_2 \subset \cdots \subset S_m \text{ and } i_1 > i_2 > \cdots > i_m \text{ such that} \\
&\qquad S_\mu \cap E_{i\mu} = \emptyset \text{ and } S_\mu \cap F_{i\mu} \neq \emptyset \text{ for } \mu = 1, 3, 5, \ldots \text{ and} \\
&\qquad S_\mu \cap E_{i\mu} \neq \emptyset \text{ and } S_\mu \cap F_{i\mu-1} = \emptyset \text{ for } \mu = 2, 4, 6, \ldots \\
&\iff \text{there exist } s \in S,\ x, u_1, u_2, \ldots, u_m \in A^*, \text{ and } i_1 > i_2 > \cdots > i_m \text{ such that} \\
&\qquad \delta(s_0, x) = s, \\
&\qquad \delta(s, u_i) = s \text{ for } \mu = 1, 2, \ldots, m, \\
&\qquad \overline{\delta}(s, u_\mu) \cap E_{i\mu} = \emptyset \text{ and } \overline{\delta}(s, u_\mu) \cap F_{i\mu} \neq \emptyset \text{ for } \mu = 1, 3, 5, \ldots, \text{ and} \\
&\qquad \overline{\delta}(s, u_\mu) \cap E_{i\mu} \neq \emptyset \text{ and } \overline{\delta}(s, u_\mu) \cap F_{i\mu-1} = \emptyset \text{ for } \mu = 2, 4, 6, \ldots.
\end{aligned}
$$

The latter can be tested by the following NL algorithm.

- Guess $s \in S$.
- Guess $x$ letterwise and check $\delta(s_o, x) = s$.
- For $\mu = 1, 2, \ldots, m$:
  - Guess $i_\mu$. If $\mu > 1$ check $i_\mu < i_{\mu+1}$.
  - If $\mu$ is odd then guess $u_\mu$ letterwise and check $\delta(s, u_i) = s$, $\overline{\delta}(s, u_\mu) \cap E_{i\mu} = \emptyset$, and $\overline{\delta}(s, u_\mu) \cap F_{i\mu} \neq \emptyset$.
  - If $\mu$ is even then guess $u_\mu$ letterwise and check $\delta(s, u_i) = s$, $\overline{\delta}(s, u_\mu) \cap E_{i\mu} \neq \emptyset$, and $\overline{\delta}(s, u_\mu) \cap F_{i\mu-1} = \emptyset$.
- Accept if all checks are o.k.

For $(\mathcal{M}, m, s, -) \in (\text{Chain})_{\text{DP}}$ we obtain analogous equivalences and an analogous algorithm. □

To understand Statement 2 of the following theorem remember that deterministic Büchi automata can accept just the sets from $\hat{\mathrm{C}}_2^1$, i.e., from $\mathrm{C}_2^1$, $\mathrm{C}_1^n$, $\mathrm{D}_1^n$, and $\mathrm{E}_1^n$ for $n \geq 1$

**Theorem 7.2** *1. For every $\mathcal{C} \in \mathcal{T}$, the problems $(\hat{\mathcal{C}})_{\text{DP}}$ and $(\mathcal{C})_{\text{DP}}$ are* NL*-complete.*

*2. The problems $(\mathrm{C}_2^1)_{\text{DB}}$, $(\hat{\mathrm{C}}_1^n)_{\text{DB}}$, $(\mathrm{C}_1^n)_{\text{DB}}$, $(\hat{\mathrm{D}}_1^n)_{\text{DB}}$, $(\mathrm{D}_1^n)_{\text{DB}}$, $(\hat{\mathrm{E}}_1^n)_{\text{DB}}$, and $(\mathrm{E}_1^n)_{\text{DB}}$ are* NL*-complete for $n \geq 1$.*

**Proof.** The membership of these problems to NL is given by Theorem 5.3 and Lemma 7.1.

The NL-hardness for deterministic Mostowski automata can be shown in exactly the same way as for deterministic Muller automata, see Theorem 6.2.

For deterministic Büchi automata we need a modification of the above proof. For $\mathrm{C}_1^n$, $\mathrm{D}_1^n$, and $\mathrm{E}_1^n$ the proof works if we choose $\mathcal{M}_2$ such that $L_\omega(\mathcal{M}_2) \in \mathrm{C}_1^{n+1}$. For $\mathrm{C}_2^1$ we choose $\mathcal{M}_1$ and $\mathcal{M}_2$ such that $L_\omega(\mathcal{M}_1) = \emptyset$ and $L_\omega(\mathcal{M}_2) \in \mathrm{C}_2^1$. This shows $\text{GAP}' \leq_{\text{m}}^{\log} (\mathrm{C}_2^1)_{\text{DB}}$. □

# 8 Deterministic Rabin and Streett Automata

We start with the complexity of chains and superchains. Just by guessing a possible chain or superchain and testing whether it is really one we obtain

**Proposition 8.1** *The problems* $\mathrm{Chain}_{\mathrm{DR}}$, $\mathrm{Chain}_{\mathrm{DS}}$, $\mathrm{Super}_{\mathrm{DR}}$, *and* $\mathrm{Super}_{\mathrm{DS}}$ *are in* NP.

From Theorem 5.3 we obtain immediately that the problems $(\hat{\mathcal{C}})_{\mathrm{DR}}$ and $(\mathcal{C})_{\mathrm{DR}}$ are in $\mathrm{P}^{\mathrm{NP}}$ for all $\mathcal{C} \in \mathcal{T}$. However, in some cases there are better upper bounds in terms of the Boolean hierarchy $\{\mathrm{NP}(n)\}_{n\geq 1}$ over NP (see e.g. [WW85]); recall that NP(1) coincides with NP, NP(2) is the class of differences of NP-sets and NP(3) is the class of sets $(A \setminus B) \cup C$ where $A, B, C$ are NP-sets. Unfortunately, in most cases there remains a gap between upper bound and lower bound. We consider Rabin automata first.

**Theorem 8.2** *1. The problem* $(\mathrm{C}_1^1)_{\mathrm{DR}}$ *is* NL*-complete.*

2. *The problem* $(\mathrm{D}_1^1)_{\mathrm{DR}}$ *is* P*-hard and in* co-NP*.*
3. *The problems* $(\mathrm{C}_m^n)_{\mathrm{DR}}$ *and* $(\mathrm{D}_m^n)_{\mathrm{DR}}$ *for* $m+n>2$*, and the problems* $(\mathrm{E}_m^n)_{\mathrm{DR}}$ *for* $m,n \geq 1$ *are* P*-hard and in* NP(2)*.*
4. *The problems* $(\hat{\mathrm{C}}_m^n)_{\mathrm{DR}}$ *and* $(\hat{\mathrm{D}}_m^n)_{\mathrm{DR}}$ *for* $m+n>2$*, and the problems* $(\hat{\mathrm{E}}_m^n)_{\mathrm{DR}}$ *for* $m,n \geq 1$ *are* P*-hard and in* co-NP(3)*.*
5. *For every* $\mathcal{C} \in \mathcal{T} \setminus \bigcup_{m,n\geq 1}\{\mathrm{C}_m^n, \mathrm{D}_m^n, \mathrm{E}_m^n\}$*, the problems* $(\hat{\mathcal{C}})_{\mathrm{DR}}$ *and* $(\mathcal{C})_{\mathrm{DR}}$ *are* P*-hard and in* $\mathrm{P}^{\mathrm{NP}}$*.*

**Proof.** 1. The upper bound follows from the fact that $(\mathrm{C}_1^1)_{\mathrm{NR}}$ is in NL (Theorem 9.1). The hardness follows from the NL-hardness of $(\mathrm{C}_1^1)_{\mathrm{DP}}$ (Theorem 7.2) because Mostowski automata are special cases of Rabin automata.

Now we consider the upper bounds for the Statements 2, 3, 4, and 5.

2. From Theorem 9.2 we conclude $(\mathrm{C}_1^1)_{\mathrm{DS}} \in$ co-NP, and Proposition 3.3 yields $(\mathrm{D}_1^1)_{\mathrm{DR}} \in$ co-NP.

For 3 and 4, the upper bounds follow from the definition of the classes in question, the fact that the characteristics used in this definitions can be expressed by $\mathrm{Chain}_{\mathrm{DR}}$ and $\mathrm{Super}_{\mathrm{DR}}$ (Proposition 5.1), and by Proposition 8.1.

For 5, the upper bound is is an immediate consequence of Proposition 8.1 and Theorem 5.3.

Now we prove the P-hardness results for the Statements 2, 3, 4, and 5. Because of Proposition 3.3 it is sufficient to prove that $(\hat{\mathcal{C}})_{\mathrm{DS}}$ and $(\mathcal{C})_{\mathrm{DS}}$ are P-hard for all $\mathcal{C} \in \mathcal{T} \setminus \{\mathrm{D}_1^1\}$.

Let $\mathcal{C} \in \mathcal{T} \setminus \{\mathrm{D}_1^1\}$, and let $\tilde{\mathcal{M}} = (\tilde{S}, A, \tilde{\delta}, \tilde{s}_0, \mathcal{E})$ be a deterministic Streett automaton such that $L\omega(\tilde{\mathcal{M}}) \in \mathcal{C}$.

We reduce the P-complete satisfiability problem for anti Horn formulas to $(\mathcal{C})_{\mathrm{DS}}$ and $(\hat{\mathcal{C}})_{\mathrm{DS}}$. For a given anti Horn formula $H = H_1 \wedge H_2 \wedge \cdots \wedge H_r$ with variables $x_1, \ldots, x_k$ we construct the deterministic Streett automaton $\mathcal{M}_H^m =_{\mathrm{def}} (S, A, \delta, s_0, \{(E_1, F_1), \ldots, (E_{r+m}, F_{r+m})\} \cup \mathcal{E})$ where

- $S =_{\mathrm{def}} \{s_0, s_1, \ldots, s_k, s'_1, \ldots, s'_k, r_1, \ldots, r_{2m+1}, t\} \cup \tilde{S}$,
- $E_i =_{\mathrm{def}} \{s_j \mid x_j \in H_i\}$ and $F_i =_{\mathrm{def}} \left\{ \begin{array}{ll} \{s_j\} & \text{if } \overline{x_j} \text{ in } H_i \\ \{t\} & \text{if no negated variable is in } H_i \end{array} \right\}$ for $i = 1, \ldots, r$,

  $E_{r+i} =_{\mathrm{def}} \{r_{2i+1}\}$ and $F_{r+i} =_{\mathrm{def}} \{r_{2i}\}$ for $i = 1, \ldots, m$,
- $\delta(s_0, a) =_{\mathrm{def}} \left\{ \begin{array}{ll} t & \text{if } a = 0 \\ \tilde{s}_0 & \text{if } a \neq 0 \end{array} \right.$,

  $\delta(s_j, a) =_{\mathrm{def}} s'_j$ for $j = 1, \ldots, k$ and $a \in A$,

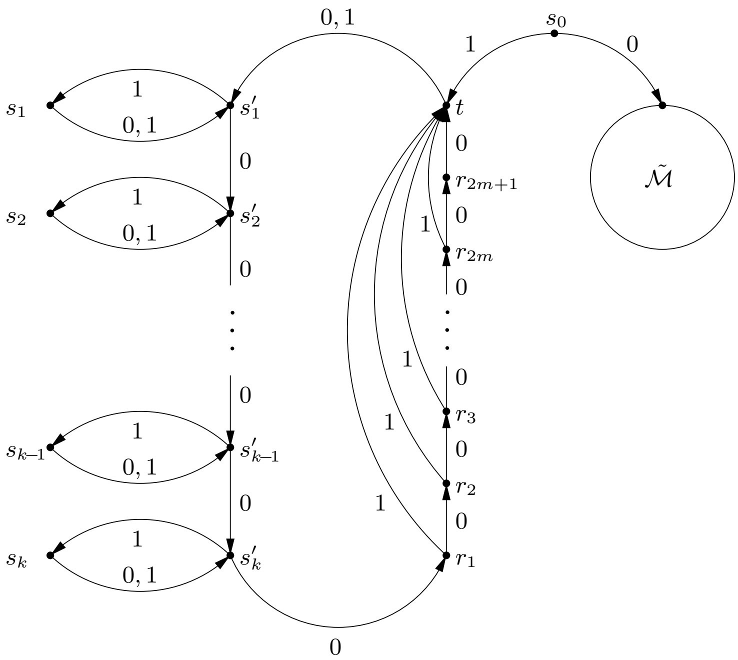
  $\delta(s'_j, a) =_{\mathrm{def}} \left\{ \begin{array}{ll} s'_{j+1} & \text{if } a = 0 \\ s_j & \text{if } a \neq 0 \end{array} \right\}$ for $j = 1, \ldots, k-1$ and $\delta(s'_k, a) =_{\mathrm{def}} \left\{ \begin{array}{ll} r_1 & \text{if } a = 0 \\ s_k & \text{if } a \neq 0 \end{array} \right.$,

  $\delta(r_j, a) =_{\mathrm{def}} \left\{ \begin{array}{ll} r_{j+1} & \text{if } a = 0 \\ t & \text{if } a \neq 0 \end{array} \right\}$ for $j = 1, \ldots, 2m$ and $\delta(r_{2m+1}, a) =_{\mathrm{def}} t$ for $a \in A$,

  $\delta(t, a) =_{\mathrm{def}} s'_1$ for $a \in A$, and

  $\delta(s, a) =_{\mathrm{def}} \tilde{\delta}(s, a)$ for $s \in \tilde{S}$ and $a \in A$.

The transition function $\delta$ is illustrated by the next figure where the input symbol 1 stands for every symbol from $A \setminus \{0\}$.




The only loops of $\mathcal{M}_H^m$ are the loops of $\tilde{\mathcal{M}}$ and the sets $S' \cup \{s'_1, \ldots, s'_k, r_1, \ldots, r_l, t\}$ with $S' \subseteq \{s_1, \ldots, s_k\}$ and $1 \leq l \leq 2m+1$. A loop $S' \cup \{s'_1, \ldots, s'_k, r_1, \ldots, r_l, t\}$ fulfills the acceptance condition of $\mathcal{M}_H^m$ if and only if the assignment $I$ defined by $I(x_l) = 1 \Leftrightarrow_{\text{def}} s_l \in S'$ satisfies $H$ and $l$ is odd.

If $H$ is satisfiable then there is an satisfying assignment $I$. For $l = 1, \ldots, 2m+1$

- the loops $\{s_j \mid I(x_j) = 1\} \cup \{s'_1, \ldots, s'_k, r_1, \ldots, r_l, t\}$ for odd $l$ satisfy the acceptance condition of $\mathcal{M}_H^m$ and
- the loops $\{s_j \mid I(x_j) = 1\} \cup \{s'_1, \ldots, s'_k, r_1, \ldots, r_l, t\}$ for even $l$ do not satisfy the acceptance condition of $\mathcal{M}_H^m$.

Hence these loops form a $(2m+1)^+$chain, i.e., $m^+(\mathcal{M}_H^m) \geq 2m+1$ and $\mathcal{M}_H^m \notin (\hat{\mathcal{C}})_{\text{DS}}$.

If $H$ is not satisfiable then only loops of $\tilde{\mathcal{M}}$ can satisfy the acceptance conditions of $\mathcal{M}_H^m$. Consequently, the non-$\tilde{\mathcal{M}}$ part of $\mathcal{M}_H^m$ can enrich the chain and superchain structure of $\tilde{\mathcal{M}}$ only if $\mathrm{m}(\tilde{\mathcal{M}}) = 1$, $\mathrm{n}^+(\tilde{\mathcal{M}}) = 1$, and $\mathrm{n}^-(\tilde{\mathcal{M}}) = 0$. But this means $L_\omega(\tilde{\mathcal{M}}) \in \mathrm{D}_1^1$ which case we do not deal with here. □

Because of Proposition 3.3 we obtain

**Theorem 8.3** *1. The problem* $(\mathrm{D}_1^1)_{\text{DS}}$ *is* NL-*complete.*

*2. The problem* $(\mathrm{C}_1^1)_{\text{DS}}$ *is* P-*hard and in* co-NP.

*3. The problems* $(\mathrm{C}_m^n)_{\text{DS}}$ *and* $(\mathrm{D}_m^n)_{\text{DS}}$ *for* $m+n > 2$*, and the problems* $(\mathrm{E}_m^n)_{\text{DS}}$ *for* $m, n \geq 1$ *are* P-*hard and in* NP(2).

*4. The problems* $(\hat{\mathrm{C}}_m^n)_{\text{DS}}$ *and* $(\hat{\mathrm{D}}_m^n)_{\text{DS}}$ *for* $m+n > 2$*, and the problems* $(\hat{\mathrm{E}}_m^n)_{\text{DS}}$ *for* $m, n \geq 1$ *are* P-*hard and in* co-NP(3).

*5. For every* $\mathcal{C} \in \mathcal{T} \setminus \bigcup_{m,n\geq 1}\{\mathrm{C}_m^n, \mathrm{D}_m^n, \mathrm{E}_m^n\}$*, the problems* $(\hat{\mathcal{C}})_{\text{DS}}$ *and* $(\mathcal{C})_{\text{DS}}$ *are* P-*hard and in* $\mathrm{P}^{\mathrm{NP}}$.

It should be noticed that we would obtain exact complexity results for deterministic Rabin and Streett automata if we could show that $\mathrm{Chain}_{\text{DR}}$ (or, equivalently, $\mathrm{Chain}_{\text{DS}}$) is in P. By Theorem 5.3, Theorem 8.2, and Theorem 8.3 we obtain

**Theorem 8.4** *Assume* $\mathrm{Chain}_{\text{DR}} \in \mathrm{P}$.

1. *For all $\mathcal{C} \in \mathcal{T} \setminus \{\mathrm{C}_1^1\}$, the problems $(\hat{\mathcal{C}})_{\mathrm{DR}}$ and $(\mathcal{C})_{\mathrm{DR}}$ are* P*-complete.*
2. *For all $\mathcal{C} \in \mathcal{T} \setminus \{\mathrm{D}_1^1\}$, the problems $(\hat{\mathcal{C}})_{\mathrm{DS}}$ and $(\mathcal{C})_{\mathrm{DS}}$ are* P*-complete.*

However, we even do not know the complexity of the problem $(\mathrm{D}_1^1)_{\mathrm{DR}}$, that is the problem of whether every loop of a given deterministic Rabin automaton satisfies the acceptance condition of this automaton. We know that this problem in P-hard and in co-NP, but we do not know whether this problem is in P or co-NP-complete.

# 9 Nondeterministic Automata

Let $\mathcal{M} = (S, A, \delta, s_0, \mathcal{E})$ be a nondeterministic $\omega$-automaton of some type. A set $S' \subseteq S$ is a *loop* of $\mathcal{M}$ if there are $l \geq 1$, $x \in A^*$, $a_1, \ldots, a_l \in A$, and $s_1, \ldots, s_l \in S$ such that $\{s_1, \ldots, s_l\} = S'$, $s_1 \in \delta(s_0, x)$, $s_{j+1} \in \delta(s_j, a_j)$ for $j = 1, \ldots, l-1$, and $s_1 \in \delta(s_l, a_l)$.

**Theorem 9.1** *Let $T \in \{\mathrm{NR}, \mathrm{NM}, \mathrm{NP}, \mathrm{NB}\}$.*
1. *The problem $(\mathrm{C}_1^1)_T$ is* NL*-complete.*
2. *For every $\mathcal{C} \in \mathcal{T} \setminus \{\mathrm{C}_1^1\}$, the problems $(\mathcal{C})_T$ and $(\hat{\mathcal{C}})_T$ are* PSPACE*-complete.*

**Proof.** 1. Upper bound. Since Mostowski and Büchi automata are special cases of Rabin automata, we can restrict ourselves to Muller and Rabin automata.

Let $\mathcal{M} = (S, A, \delta, s_0, \{(E_1, F_1), \ldots, (E_r, F_r)\})$ be a nondeterministic Rabin automaton. We conclude

$$\begin{aligned} \mathcal{M} \notin (\mathrm{C}_1^1)_{\mathrm{NR}} &\Leftrightarrow \text{there exists a loop of } \mathcal{M} \text{ which satisfies the acceptance condition of } \mathcal{M} \\ &\Leftrightarrow \text{there exists } x \in A^*, l \geq 1, a_1, \ldots, a_l \in A, s_1, \ldots, s_l \in S, \text{ and } i \in \{1, \ldots, r\} \\ &\quad\ \text{such that } s_1 \in \delta(s_0, x), s_{j+1} \in \delta(s_j, a_j) \text{ for } j = 1, \ldots, l-1, s_1 \in \delta(s_l, a_l), \\ &\quad\ \{s_1, \ldots, s_l\} \cap E_i = \emptyset \text{ and } \{s_1, \ldots, s_l\} \cap F_i \neq \emptyset. \end{aligned}$$

The latter can be checked by an NL-algorithm. Hence $(\mathrm{C}_1^1)_{\mathrm{NR}} \in$ co-NL = NL.

For a nondeterministic Muller automaton $\mathcal{M} = (S, A, \delta, s_0, \{S_1, \ldots, S_r\})$ we obtain

$$\begin{aligned} \mathcal{M} \notin (\mathrm{C}_1^1)_{\mathrm{NM}} &\Leftrightarrow \text{there exists } i \in \{1, \ldots, r\} \text{ such that } S_i \text{ is a loop of } \mathcal{M} \\ &\Leftrightarrow \text{there exists } i \in \{1, \ldots, r\} \text{ (let } s_1, \ldots, s_l \text{ be the states in } S_i \text{ in the order they appear} \\ &\quad\ \text{on the input tape)}, x, u = u_1 \ldots u_l \in A^*, \text{and } v_1, \ldots, v_l \in S^* \text{ such that, for } j = 1, \ldots, l: \\ &\quad\ - v_i \text{ is a possible sequence of states for input } u_j \text{ to } \mathcal{M} \text{ with initial state } s_j, \\ &\quad\ - \text{the last state of } v_j \text{ is } s_{j+1} (\text{set } s_{l+1} =_{\mathrm{def}} s_1), \text{ and} \\ &\quad\ - \text{all states of } v_j \text{ are in } S_i. \end{aligned}$$

The latter can be checked by an NL-algorithm. Hence $(\mathrm{C}_1^1)_{\mathrm{NM}} \in$ co-NL = NL.

The hardness follows from the fact that these problems for the corresponding deterministic types are already NL-hard.

2. Upper bound. Since a nondeterministic Muller automaton can be converted in polynomial time into an equivalent nondeterministic Rabin automaton, and since Mostowski and Büchi automata are special cases of Rabin automata (see Theorem 3.5) we can restrict ourselves to the case of nondeterministic Rabin automata.

In [Sa88] an algorithm $A_1$ is given which converts a nondeterministic Rabin automaton $\mathcal{M}$ with $n$ states into an equivalent deterministic Muller automaton $\mathcal{M}'$ with $2^{\mathcal{O}(n \log n)}$ states. The states of $\mathcal{M}'$ are trees of maximum size $n$ which are labeled with sets of states of $\mathcal{M}$. The procedure to find the next state for an input symbol is done in space polynomial in $n$. Hence the algorithm $A_1$ works in polynomial space.

Now we combine this polynomial space algorithm $A_1$ which converts the nondeterministic Rabin automaton $\mathcal{M}$ into a deterministic Muller automaton $\mathcal{M}'$ with the NL-algorithm $A_2$ from Theorem 8.3 checking

$L_\omega(\mathcal{M}') \in \mathcal{C}$ or $L_\omega(\mathcal{M}') \in \hat{\mathcal{C}}$, resp. The problem arises that we cannot write down the exponentially long $\mathcal{M}'$ in polynomial space. We overcome this difficulty as follows. We do not store $\mathcal{M}'$ but only the position of the input head of $A_2$ when working on $\mathcal{M}'$. Any time when $A_2$ needs another bit of $\mathcal{M}'$ we let re-run the computation of $A_1$ on $\mathcal{M}$ until the moment when this bit is produced. Combining $A_1$ and $A_2$ in such a way we obtain a PSPACE-computation.

Hardness. We reduce the PSPACE-complete problem of whether a given finite nondeterministic automaton (on finite words) accepts $A^*$ (see [MS72]) to our problems. Fix two regular $\omega$-languages $L_1 \notin \hat{\mathcal{C}}$ and $L_2 \in \mathcal{C}$. Define the homomorphism $h : A^* \to A^*$ by $h(a) =_{\mathrm{def}} aa$ for $a \in A$, and set $\tilde{A} =_{\mathrm{def}} A^2 \setminus \{aa \mid a \in A\}$. For every language $L \subseteq A^*$ we define the $\omega$-language $L' =_{\mathrm{def}} 0h(L)\tilde{A}A^\omega \cup 0h(A^*)\tilde{A}L_1 \cup 0h(A)^\omega \cup (A \setminus \{0\})L_2$. Obviously, for a given nondeterministic finite automaton $\mathcal{M}$ accepting $L$ one can construct in logarithmic space a nondeterministic automaton of any type that accepts $L'$.

If $L = A^*$ then $L' =_{\mathrm{def}} 0h(A^*)\tilde{A}A^\omega \cup 0h(A)^\omega \cup (A \setminus \{0\})L_2 = 0A^\omega \cup (A \setminus \{0\})L_2$. Because of $\mathcal{C} \neq \mathrm{C}_1^1$ we have $\mathrm{m}^+(L_2) \geq 1$. Since $\mathrm{m}^+(A^\omega) = 1$ and $\mathrm{m}^-(A^\omega) = 0$, the $0A^\omega$ part of $L'$ connot enrich the chain and superchain structure of the $(A \setminus \{0\})L_1$ part of $L'$. Consequently, $L'$ and $L_2$ are in the same Wadge degree, i.e., $L' \in \mathcal{C}$.

If $L \neq A^*$ then there exists an $x_0 \in A^* \setminus L$. Because of $0h(x_0)01A^\omega \cap 0h(L)\tilde{A}A^\omega = \emptyset$ we obtain $0h(x_0)01A^\omega \cap L' = 0h(x_0)01L_1$. Consider a deterministic Muller automaton $(S, A, \delta, s_0, \mathcal{F})$ accepting $L'$. Then $(S, A, \delta, \delta(s_0, 0h(x_0)01), \mathcal{F})$ accepts $L_1$. Because of Proposition 4.10 we have $L_1 \leq_{\mathrm{w}} L'$, and by $L_1 \notin \hat{\mathcal{C}}$ we obtain $L' \notin \hat{\mathcal{C}}$. □

**Theorem 9.2** *1. The problem* $(\mathrm{C}_1^1)_{\mathrm{NS}}$ *is* P*-hard and in* co-NP*.*

*2. For every* $\mathcal{C} \in \mathcal{T} \setminus \{\mathrm{C}_1^1\}$ *the problems* $(\mathcal{C})_{\mathrm{NS}}$ *and* $(\hat{\mathcal{C}})_{\mathrm{NS}}$ *are* PSPACE*-hard and in* EXPSPACE*.*

**Proof.** 1. Upper bound. Let $\mathcal{M} = (S, A, \delta, s_0, \{(E_1, F_1), \ldots, (E_r, F_r)\})$ be a nondeterministic Streett automaton. The upper bound follows from the equivalence

$$\mathcal{M} \in (\mathrm{C}_1^1)_{\mathrm{DS}} \Leftrightarrow \forall (S' \subseteq S)\big(S' \text{ loop in } \mathcal{M} \to \bigvee_{i=1}^{r} (S' \cap E_i = \emptyset \wedge S' \cap F_i \neq \emptyset)\big)$$

which is clearly a co-NP-condition.

For the P-hardness observe that even $(\mathrm{C}_1^1)_{\mathrm{DS}}$ is P-hard (Theorem 8.3).

2. Upper bound. An exponential time algorithm can convert the nondeterministic Streett automaton $\mathcal{M} = (S, A, \delta, s_0, \{(E_1, F_1), \ldots, (E_r, F_r)\})$ into the equivalent nondeterministic Muller automaton $\mathcal{M}' = (S, A, \delta, s_0, \{S' \subseteq S \mid \bigwedge_{i=1}^{r} (S' \cap E_i \neq \emptyset \vee S' \cap F_i = \emptyset)\})$. Now apply Theorem 9.1.2.

The PSPACE-hardness follows from the PSPACE-hardness of the corresponding problems for Mostowski automata (Theorem 9.1.2) because of Theorem 3.5 and Proposition 3.4. □

**Acknowledgement.** The authors are grateful to Ludwig Staiger for his impressive contribution to the theory of $\omega$-languages and for many helpful discussions.

# References

[KPB95] S. Krishnan, A. Puri and R. Brayton. Structural complexity of $\omega$-automata. *Lecture Notes in Computer Science*, v. 915, Springer: Berlin 1995, p. 143–156.

[La69] L.H. Landweber. Decision problems for $\omega$-automata. *Mathematical Systems Theory*, 4 (1969), 376–384.

[MS72] A. Meyer and L.J. Stockmeyer. *The equivalence problem for regular expressions with squaring requires exponential time.* Proc. 13th IEEE Symp. on Switching and Automata Theory (1972), 125-129

[Sa88] S. Safra. On the complexity of $\omega$-automata. *IEEE FOCS* 1988, 319–327.

[SVW87] A.P. Sistla, M.Y. Vardi and P. Wolper. The complementation problem for Büchi automata with applications to temporal logic. *Theoretical Computer Science* 49 (1987), 217–237.

[Sta97] L. Staiger. $\omega$-Languages. In: *Handbook of Formal Languages v. 3*, Springer, Berlin, 1997, 339–387.

[SW74] L. Staiger and K. Wagner. Automatentheoretische und automatenfreie Characterisierungen topologischer Klassen regulärer Folgenmengen. *Elektronische Informationsverarbeitung und Kybernetik* 10 No 7 (1974), 379–392.

[Th90] W. Thomas. Automata on infinite objects. *Handbook of Theoretical Computer Science*, v. B (1990), p.133–191.

[Th96] W. Thomas. Languages, automata and logic. *Handbook of Formal Language theory*, v. B (1996), p.133–191.

[Wa72] W. Wadge. Degrees of complexity of subsets of the Baire space. *Notices AMS*, 1972, A-714.

[Wa84] W. Wadge. *Reducibility and determinateness in the Baire space.* PhD thesis, University of California, Berkely, 1984.

[Wag79] K. Wagner. On $\omega$-regular sets. *Information and Control*, 43 (1979), p. 123–177.

[WW85] G. Wechsung and K.W. Wagner. On the Boolean closure of NP, *Manuskript.* Extended abstract as: G. Wechsung, On the Boolean closure of NP, *Proceedings of the 1985 Int. Conf. on Fundamentals of Computation theory*, v.199 of *Lecture Notes in Computer Science*, p.485–493. Springer-Verlag, 1985.

[WY95] T. Wilke and H. Yoo. Computing the Wadge degree, the Lipschitz degree, and the Rabin index of a regular language of infinite words in polynomial time. *Lecture Notes in Computer Science*, v. 915, Springer: Berlin 1995, p. 288–302.