A Robust Measure on FDFAs Following Duo-Normalized Acceptance

Dana Fisman

The Department of Computer Science, Ben-Gurion University of the Negev, Beer-Sheva, Israel

Emmanuel Goldberg

The Department of Computer Science, Ben-Gurion University of the Negev, Beer-Sheva, Israel

Oded Zimerman

The Department of Computer Science, Ben-Gurion University of the Negev, Beer-Sheva, Israel

Abstract

Families of DFAs (FDFAs) are a computational model recognizing ω-regular languages. They were introduced in the quest of finding a Myhill-Nerode theorem for ω-regular languages and obtaining learning algorithms. FDFAs have been shown to have good qualities in terms of the resources required for computing Boolean operations on them (complementation, union, and intersection) and answering decision problems (emptiness and equivalence); all can be done in non-deterministic logarithmic space. In this paper we study FDFAs with a new type of acceptance condition, *duo-normalization*, that generalizes the traditional *normalization* acceptance type. We show that duo-normalized FDFAs are advantageous to normalized FDFAs in terms of succinctness as they can be exponentially smaller. Fortunately this added succinctness doesn't come at the cost of increasing the complexity of Boolean operations and decision problems — they can still be preformed in NLOGSPACE.

An important measure of the complexity of an ω-regular language is its position in the Wagner hierarchy (aka the Rabin Index). It is based on the inclusion measure of Muller automata, and for the common ω-automata there exist algorithms computing their position. We develop a similarly robust measure for duo-normalized (and normalized) FDFAs, which we term the *diameter measure*. We show that the diameter measure corresponds one-to-one to the position in the Wagner hierarchy. We show that computing it for duo-normalized FDFAs is PSPACE-complete, while it can be done in NLOGSPACE for traditional FDFAs.

2012 ACM Subject Classification Theory of computation; Theory of computation → Formal languages and automata theory; Theory of computation → Automata over infinite objects

Keywords and phrases Omega-Regular Languages, Families of DFAs, Complexity Measure, Wagner Hierarchy, Rabin Index

Digital Object Identifier 10.4230/LIPIcs.MFCS.2024.53

Related Version *Full Version*: https://arxiv.org/abs/2310.16022 [12]

Funding *Oded Zimerman*: Supported by ISF grant 2507/21.

1 Introduction

Regular languages of finite words possess a natural canonical representation — the unique minimal DFA. The essence of the representation lies in a right congruence relation for a language L saying that two words x and y are *equivalent*, denoted $x \sim_L y$, if and only if $xz \in L \iff yz \in L$ for every finite word $z \in \Sigma^*$. The famous Myhill-Nerode theorem [20, 21] relates the equivalence classes of $\sim_L$ to the set of words reaching a state of the minimal DFA.

For regular languages of infinite words the situation is more complex. First, there is no unique minimal automaton for any of the common ω-automata acceptance conditions (Büchi, Muller, Rabin, Streett and parity). Second, one can indeed define two finite words x and y to be *equivalent* with respect to an ω-regular language L, denoted $x \sim_L y$, if $xz \in L \iff yz \in L$

49th International Symposium on Mathematical Foundations of Computer Science (MFCS 2024).
Editors: Rastislav Královič and Antonín Kučera; Article No. 53; pp. 53:1–53:17
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

for every infinite word $z \in \Sigma^\omega$. However, there is no one-to-one correspondence between these equivalence classes and a minimal ω-automaton for L. Consider for instance the language L_1 stipulating that *aab* occurs infinitely often. The right congruence relation $\sim_{L_1}$ has only one equivalence class, yet clearly an automaton for L_1 needs more than one state.

A quest for a characterization of an ω-regular language L, relating equivalence classes of a semantic definition of L to states of an automaton for L, has led to the development of families of right concurrences (FORCs) [19] and families of DFAs (FDFAs) [14, 2]. Several canonical FDFAs were introduced over the years, the periodic FDFA [7], the syntactic FDFA [19], the recurrent FDFA [2], and the limit FDFA [17]. All these representations have a one-to-one correspondence between the equivalence classes of semantic right congruence relations and the states of the respective automata. This is very satisfying in the sense that they induce a semantic canonical representation, ie one that is agnostic to a particular automaton; and this is a beneficial property when it comes to learning [2, 18]. FDFAs have additional good qualities — computing Boolean operations on them (complementation, union, and intersection) and answering decision problems (emptiness and equivalence) can all be done cheaply, in non-deterministic logarithmic space [1].

Loosely speaking, an FDFA is composed of a *leading automaton* $\mathcal{Q}$ and a family of *progress DFAs* $\{\mathcal{P}_q\}$, one for each state q of $\mathcal{Q}$. FDFAs consider only ultimately periodic words, ie words of the form $u(v)^\omega$ for $u \in \Sigma^*$ and $v \in \Sigma^+$. Since two ω-regular languages recognize the same language if and only if they agree on the set of ultimately periodic words [6, 7], this is not really a limitation. Exact acceptance of an ultimately periodic word (u, v) representing uv^ω is determined by checking acceptance of v in the progress DFA corresponding to the state reached in the leading automaton by reading u. Normalized acceptance is done by first *normalizing* the word wrt the leading automaton — this means considering a decomposition (uv^i, v^j) of uv^ω such that v^j loops on the state of the leading automaton reached by reading uv^i. This normalization was introduced as it leads to an exponential save in the number of states [2]. In this paper we consider a new acceptance condition for FDFAs, which we term *duo-normalization*, which considers decompositions (uv^i, v^j) where in addition v^j closes a loop on the state it arrives at in the respective progress DFA. We term FDFAs with this new type of acceptance *duo-normalized* FDFAs. The notion of duo-normalization has appeared in the literature before [27, 9, 1, 5] and was suggested as an acceptance condition for FDFAs in the future work of [5].

We show that duo-normalized FDFAs also enjoy the good qualities of computing Boolean operations and answering decision problems in non-deterministic logarithmic space. In terms of succinctness we show that they can be exponentially smaller than normalized FDFAs.

We are also interested in the problem of finding their position in the *Wagner hierarchy*, a hierarchy reflecting the complexity of ω-regular properties, that often correlates to the complexity of algorithms on ω-regular languages and games [29, 13, 3]. It is noted in [10, Sec. 5] that while for ω-automata there are algorithms for computing their position in the Wagner hierarchy, there is no clear way to relate the structure of a particular FDFA to its Wagner position.

In [26] Wagner defined a complexity measure on Muller automata: the *inclusion measure*. Wagner showed that the inclusion measure is robust in the sense that any two Muller automata for the same language (minimal or not) have the same inclusion measure. This is thus a semantic property of the language. Since the inclusion measure is unbounded it induces an infinite hierarchy. The position on the Wagner hierarchy has been shown to be tightly related to the minimal number of colors required in a parity automaton, and the minimal number of pairs required in a Rabin/Street automaton. Deterministic Büchi and coBüchi (which are less expressive than deterministic Muller/Rabin/Streett/parity automata, that are capable of recognizing all the ω-regular languages) lie in the bottom levels of the hierarchy. Given a deterministic ω-automaton (Büchi, coBüch, Muller, Rabin, Streett, or parity), its position in the Wagner hierarchy can be computed in polynomial time [28, 8, 22].

We develop a syntactic notion of a measure on FDFAs, that we term *the diameter measure*. Loosely speaking it relates to chains of prefixes $v_1 \prec v_2 \prec \ldots \prec v_k$ such that $u(v_i)^\omega \in L$ iff $u(v_{i+1})^\omega \notin L$, and moreover, each of the words v_i is *persistent* in the progress DFA of some $u \in \Sigma^*$. The precise definition of the term *persistent* and *persistent chains* is deferred to § 4. We show there that this measure is robust in the sense that computing it on two FDFAs for the same language will give the same result. The proof is by relating it to the position on the Wagner hierarchy. We show that computing the Wagner position of a duo-normalized FDFA can be done in PSPACE and it is PSPACE-complete, whereas for normalized FDFAs this computation can be done in NLOGSPACE. So this is one place where the added succinctness of duo-normalized FDFAs comes at a price.

The rest of the paper is organized as follows. We give some basic definitions and explain the Wagner hierarchy in § 2. We introduce duo-normalized FDFAs in § 3 where we show that it is not more expensive to compute the Boolean operations on them, or to answer emptiness and equivalence about them. § 4 is devoted to defining the *diameter measure* and proving that its computation is PSPACE-complete. § 5 provides several succinctness results relating duo-normalized FDFAs and normalized FDFAs, including results regarding succinctness of previously studied canonical FDFAs and the *Colorful FDFA*, a canonical duo-normalized FDFA. We conclude with a short discussion in § 6. Due to space limitations, proofs are deferred to the appendix of the full version [12].

2 Preliminaries

We use $[i..j]$ for the set $\{i, i+1, \ldots, j\}$. A (complete deterministic) *automaton structure* is a tuple $\mathcal{A} = (\Sigma, Q, q_0, \delta)$ consisting of an alphabet Σ, a finite set Q of states, an initial state q_0, and a complete transition function $\delta : Q \times \Sigma \to Q$. A run of an automaton on a finite word $v = a_1 a_2 \cdots a_n$ is a sequence of states $q_0, q_1, \ldots, q_n$, starting with the initial state, such that for each $i \geq 0$, $q_{i+1} = \delta(q_i, a_i)$. A run on an infinite word is defined similarly and results in an infinite sequence of states. Let $\mathcal{A} = (\Sigma, Q, q_0, \delta)$ be an automaton structure. We say that a word $w \in \Sigma^*$ reaches state q if the run of $\mathcal{A}$ on w ends in q, and use $\mathcal{A}(w)$ to denote q.

By augmenting an automaton structure with an acceptance condition α, obtaining a tuple $(\Sigma, Q, q_0, \delta, \alpha)$, we get an *automaton*, a machine that accepts some words and rejects others. An automaton accepts a word if the run on that word is accepting. For finite words the acceptance condition is a set $F \subseteq Q$ and a run on a word v is accepting if it ends in an accepting state, ie a state $q \in F$. For infinite words, there are various acceptance conditions in the literature. The common ones are Büchi, coBüchi, Muller, Rabin, Streett and parity. They are all defined with respect to the set of states visited infinitely often during a run. For a run $\rho = q_0 q_1 q_2 \ldots$ we define $\mathit{inf}(\rho) = \{q \in Q \mid \forall i {\in} \mathbb{N}.\ \exists j {>} i.\ q_j = q\}$. We focus here on the most common types — Büchi, coBüchi, Muller and parity.

- A *Büchi* (resp. *coBüchi*) acceptance condition is a set $F \subseteq Q$. A run of a Büchi (resp. coBüchi) automaton is accepting if it visits F infinitely (resp. finitely) often. That is, if $\mathit{inf}(\rho) \cap F \neq \emptyset$ (resp. $\mathit{inf}(\rho) \cap F = \emptyset$).
- A *parity* acceptance condition is a mapping $\kappa : Q \to \{0, 1, \ldots, k\}$ of a state to a number (referred to as a *color*). For a subset $Q' \subseteq Q$, we use $\kappa(Q')$ for the set $\{\kappa(q) \mid q \in Q'\}$. A run ρ of a parity automaton is accepting if the **minimal** color in $\kappa(\mathit{inf}(\rho))$ is **even**.

- A *Muller* acceptance condition is a set $\alpha = \{F_1, \ldots, F_k\}$ where $F_i \subseteq Q$ for all $1 \leq i \leq k$. A run ρ of a Muller automaton is accepting if $\mathit{inf}(\rho) \in \alpha$. That is, if the set of states visited infinitely often by the run ρ is exactly one of the sets F_i specified in α.

We use DBA, DCA, DPA, and DMA as acronyms for deterministic (complete) Büchi, coBüchi, parity, and Muller automata, respectively. We use $[\![\mathcal{A}]\!]$ to denote the set of words accepted by a given automaton $\mathcal{A}$. Two automata $\mathcal{A}$ and $\mathcal{B}$ are *equivalent* if $[\![\mathcal{A}]\!]=[\![\mathcal{B}]\!]$. Let $\mathcal{A} = (\Sigma, Q, q_0, \delta, F)$ be a DFA and $q \in Q$. We use $\mathcal{A}_{[q}$ for $(\Sigma, Q, q, \delta, F)$, namely a DFA for the words exiting state q of $\mathcal{A}$.

The *syntactic right congruence relation* for an ω-language L relates two finite words x and y if there is no infinite suffix z differentiating them, that is, for $x, y \in \Sigma^*$, $x \sim_L y$ if $\forall z \in \Sigma^\omega (xz \in L \iff yz \in L)$. We use $[u]_{\sim L}$ (or simply $[u]$) for the equivalence class of u induced by $\sim_L$. A right congruence $\sim$ can be naturally associated with an automaton structure (Σ, Q, q_0, δ) as follows: the set of states Q are the equivalence classes of $\sim$. The initial state q_0 is the equivalence class $[\epsilon]$. The transition function δ is defined by $\delta([u], \sigma) = [u\sigma]$. We use $\mathcal{A}[\sim]$ to denote the automaton structure induced by $\sim$.

2.1 The Wagner Hierarchy

Let $\mathcal{M} = (\Sigma, Q, q_0, \delta, \alpha)$ be a complete deterministic Muller automaton, where all states are reachable. We use the term *strongly connected component (SCC)* for a set of states $S \subseteq Q$ such that there is a non-empty path between every pair of states in S. Thus, if S is a singleton $\{q\}$ we require a self-loop on q for S to be an SCC. We use the term *MSCC* for a *maximal SCC*, that is, an SCC S such that no set $S' \supset S$ is an SCC. We say that an SCC $S \subseteq Q$ is *accepting* iff $S \in \alpha$. Otherwise we say that S is *rejecting*. We define the *positive inclusion measure* of $\mathcal{M}$, denoted $|\mathcal{M}|^+_\subset$, to be the maximal length of an inclusion chain $S_1 \subset S_2 \subset S_3 \subset \cdots \subset S_k$ of SCCs with alternating acceptance where S_1 is accepting. (Therefore for each $1 \leq i \leq k$ if i is odd then S_i is accepting, and if it is even then S_i is rejecting.) Likewise, we define the *negative inclusion measure* of $\mathcal{M}$, denoted $|\mathcal{M}|^-_\subset$, to be the maximal length of an inclusion chain where the first SCC is rejecting. Note that for any $\mathcal{M}$ the difference between $|\mathcal{M}|^+_\subset$ and $|\mathcal{M}|^-_\subset$ may be at most one, since by omitting the first element of a chain we remain with a chain shorter by one, and of the opposite sign. We use $L_{\infty aa \wedge \neg\infty bb}$ in Ex. 2.2 to illustrate the concepts explained throughout this section.

Wagner [26] showed that this measure is robust in the sense that any two DMAs that recognize the same language have the same positive and negative inclusion measures.

▶ **Theorem 2.1** (Robustness of the inclusion measures [26]). *Let $\mathcal{M}_1$, $\mathcal{M}_2$ be two DMAs where $[\![\mathcal{M}_1]\!] = [\![\mathcal{M}_2]\!]$. For $i \in \{1, 2\}$, let $|\mathcal{M}_i|^+_\subset = p_i$ and $|\mathcal{M}_i|^-_\subset = n_i$. Then $p_1 = p_2$ and $n_1 = n_2$.*

Since this measure is robust and since one can construct DMAs with arbitrarily long inclusion chains, the inclusion measure yields an infinite hierarchy of ω-regular languages. Formally, the classes of the Wagner hierarchy are defined as follows for a positive integer k:

$$\begin{aligned}
\mathbb{DM}^+_k &= \{L \mid \exists \text{ DMA } \mathcal{M} \text{ s.t. } [\![\mathcal{M}]\!] = L \text{ and } |\mathcal{M}|^+_\subset \leq k \text{ and } |\mathcal{M}|^-_\subset < k\} \\
\mathbb{DM}^-_k &= \{L \mid \exists \text{ DMA } \mathcal{M} \text{ s.t. } [\![\mathcal{M}]\!] = L \text{ and } |\mathcal{M}|^+_\subset < k \text{ and } |\mathcal{M}|^-_\subset \leq k\} \\
\mathbb{DM}^\pm_k &= \{L \mid \exists \text{ DMA } \mathcal{M} \text{ s.t. } [\![\mathcal{M}]\!] = L \text{ and } |\mathcal{M}|^+_\subset \leq k \text{ and } |\mathcal{M}|^-_\subset \leq k\}
\end{aligned}$$

The hierarchy is depicted in Fig. 3.1 (left). Note that if $\mathcal{A}$ is an ω-automaton, for any of the ω-automata types, then it can be recognized by a Muller automaton on the same structure. Transforming a Büchi $\mathcal{B}$ automaton with accepting states F to a Muller automaton $\mathcal{M}_\mathcal{B}$ yields an acceptance condition $\alpha_\mathcal{B} = \{F' \mid F' \cap F \neq \emptyset\}$. Note that for any $F' \in \alpha_\mathcal{B}$ and $F'' \supseteq F'$ it holds that $F'' \in \alpha_\mathcal{B}$. Therefore $|\mathcal{M}_\mathcal{B}|^+_\subset = 1$ and $|\mathcal{M}_\mathcal{B}|^-_\subset = 2$ (unless $[\![\mathcal{B}]\!] = \Sigma^\omega$ or

$[\![\mathcal{B}]\!] = \emptyset$). Hence all languages recognized by a DBA are in $\mathbb{DM}_2^-$. Dually, one can see that all languages recognized by a DCA are in $\mathbb{DM}_2^+$. It can be shown that a parity automaton for a language in $\mathbb{DM}_k^-$ can suffice with colors $\{1, \ldots, k\}$ if k is odd and $\{0, \ldots, k-1\}$ if it is even [22]. Likewise, a DPA for a language in $\mathbb{DM}_k^+$ can suffice with colors $\{0, \ldots, k-1\}$ if k is odd and with $\{1, \ldots, k\}$ otherwise. A DPA in $\mathbb{DM}_k^\pm$ requires $k+1$ colors starting with 0.

▶ **Example 2.2.** Fig. 3.1 (middle) shows a Muller automaton $\mathcal{M}$ for the language $L_{\infty aa \wedge \neg\infty bb}$. The inclusion chain $\{q_1, q_2\} \subset \{q_1, q_2, q_3\} \subset \{q_1, q_2, q_3, q_4\}$ is a negative inclusion chain of length 3 (since $\{q_1, q_2\}$ is rejecting, $\{q_1, q_2, q_3\}$ is accepting, and $\{q_1, q_2, q_3, q_4\}$ is rejecting). There are no negative inclusion chains of length 4, and there are no positive inclusion chains of length 3. (Note that $\{q_3\} \subset \{q_2, q_3\} \subset \{q_1, q_2, q_3\}$ is not an inclusion chain since $\{q_2, q_3\}$ is not an SCC.) We can thus conclude that $L_{\infty aa \wedge \neg\infty bb} \in \mathbb{DM}_3^-$. Consider the parity automaton $\mathcal{P}$ for $L_{\infty aa \wedge \neg\infty bb}$ defined on the same structure as $\mathcal{M}$. It uses the three colors $\{1, 2, 3\}$ in accordance with our conclusion that $L_{\infty aa \wedge \neg\infty bb} \in \mathbb{DM}_3^-$.

3 FDFAs with duo-normalized acceptance condition

As already mentioned, none of the common ω-automata has a unique minimal automaton, and the number of states in the minimal automaton may be bigger than the number of equivalence classes in $\sim_L$. For example, $L_2 = (\Sigma^* abc)^\omega$ has one equivalence class under $\sim_{L_2}$, since for any finite word x, an infinite extension xw for $w \in \Sigma^\omega$ is in the language iff $w \in L_2$.

The quest for finding a correspondence between equivalence classes of the language and an automaton model led to the development of *Families of Right Congruences* (FORCs) [19] and *Families of DFAs* (FDFAs) [2]. These definitions consider only ultimately periodic words (ie words of the form uv^ω) building on the well-known result that two ω-regular languages are equivalent if and only if they agree on the set of ultimately periodic words [6, 7]. We also consider only such words, and represent them as pairs (u, v) for $u \in \Sigma^*$ and $v \in \Sigma^+$.

Several canonical FDFAs were introduced over the years, the periodic FDFA [7], the syntactic FDFA [19], the recurrent FDFA [2], and the limit FDFA [17]. We do not go into the details of their definition but summarize the succinctness relations among them. It was shown in [2] that the syntactic and recurrent FDFAs can be exponentially more succinct than the periodic FDFA, while the translations in the other direction are at most polynomial. Further, the recurrent FDFA is never bigger, and can be quadratically more succinct, than the syntactic FDFA [2]. Limit FDFAs are the duals of recurrent FDFAs and similarly can also be at most quadratically bigger than the syntactic; and there are examples of quadratic blowups in the transformation from the recurrent to the limit and vice versa [17].

The gain in succinctness in going from the syntactic to the recurrent (or limit) FDFAs originates from removing the requirement that $x \approx^u y$ implies that $ux \sim uy$, which comes from the definition of a FORC.[1] The gain in succinctness of the syntactic/recurrent/limit FDFAs compared to the periodic FDFA is due to the use of a different type of acceptance condition.

An FDFA is a pair $\mathcal{F} = (\mathcal{Q}, \{\mathcal{P}_q\}_{q \in Q})$ consisting of a *leading* automaton structure $\mathcal{Q}$ and of a *progress* DFA $\mathcal{P}_q$ for each state q of $\mathcal{Q}$. There are a few ways to define acceptance on FDFAs. They differ on what decompositions (u, v) of an infinite word w are considered. We

[1] A FORC is a pair $\mathcal{F} = (\sim, \{\approx^u\})$ where $\sim$ is a right congruence, $\approx^u$ is a right congruence for every equivalence class u of $\sim$, and it satisfies that $x \approx^u y$ implies $ux \sim uy$. An ω-language L is recognized by $\mathcal{F}$ if it can be written as a union of sets of the form $[u]([v]_u)^\omega$ s.t. $uv \sim_L u$. Every FORC corresponds to an FDFA, but the other direction many not hold. This is since there is no requirement on the relation between the progress DFAs and the leading automaton in an FDFA, while there is in a FORC.

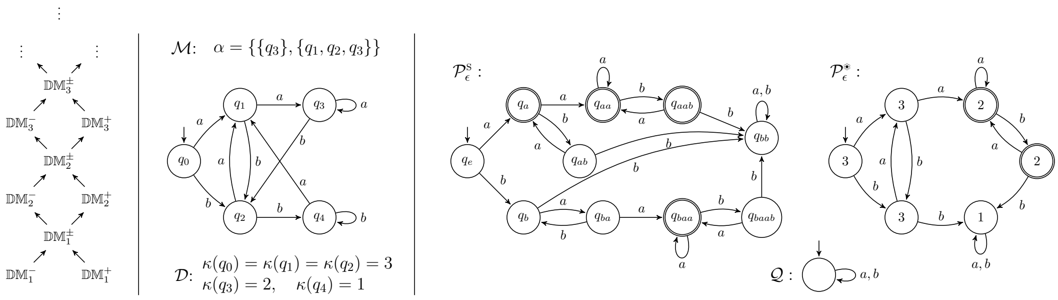

Figure 3.1 Left: The Wagner hierarchy. **Middle:** A DMA $\mathcal{M}$ and a DPA $\mathcal{D}$ for the language $L_{\infty aa \wedge \neg\infty bb}$. **Right:** Two FDFAs $\mathcal{F}^{\text{S}} = (\mathcal{Q}, \{\mathcal{P}^{\text{S}}_{\epsilon}\})$ and $\mathcal{F}^{\circledast} = (\mathcal{Q}, \{\mathcal{P}^{\circledast}_{\epsilon}\})$ for the language $L_{\infty aa \wedge \neg\infty bb}$ where $\mathcal{Q}$ is a one-state leading automaton. $\mathcal{F}^{\text{S}}$ uses normalized acceptance, $\mathcal{F}^{\circledast}$ uses duo-normalized acceptance.

provide a definition for such decompositions in Def. 3.1. Once an α-decomposition is defined, an ω-word w is accepted by an FDFA using α-acceptance if there exists an α-decomposition (u, v) of w which is accepted. That is, the word v is accepted by $\mathcal{P}_{\mathcal{Q}(u)}$ where $\mathcal{P}_{\mathcal{Q}(u)}$ is the progress DFA corresponding to the state $\mathcal{Q}(u)$ reached by the leading automaton after reading u. We henceforth use $\mathcal{P}_u$ for $\mathcal{P}_{\mathcal{Q}(u)}$.

In exact acceptance, that is used in the periodic FDFA, any decomposition of the ω-word into an ultimately periodic word is considered. In normalized acceptance, used by the other three canonical FDFAs, only decompositions (u, v) in which the periodic part v loops in the leading automaton (ie $\mathcal{Q}(u) = \mathcal{Q}(uv)$) are considered.

As shown in [2] this acceptance condition, termed *normalization*, can yield an exponential save in the number of states. The intuition is that some periods are easier to verify as good periods if one considers some repetitions of them. For instance, in the language $(121 + 212)^{\omega}$ it is harder to figure out that $(\epsilon, 12)$ should be accepted than it is for $(\epsilon, 121{\cdot}212)$ though both represent the same ω-word.

For similar reasons, one may wonder if considering only decompositions that also close a loop in the progress automaton might lead to an exponential save as well. In the following we define FDFAs with such an acceptance condition, which we term *duo-normalization*. The notion of duo-normalization has appeared in the literature before. In particular, it resembles the notion of a linked-pair in ω-semigroups and Wilke-algebras [27, 9], it is used in [1, Proof of Thm. 5.8] and it is termed *idempotent* in [5].

▶ **Definition 3.1** (ω-words decomposition wrt an FDFA)**.** *Let $u \in \Sigma^*$, $v \in \Sigma^+$ and $w \in \Sigma^{\omega}$. Let $\mathcal{F} = (\mathcal{Q}, \{\mathcal{P}_q\}_{q \in Q})$ be an FDFA.*

- *(u, v) is a decomposition of w if $uv^{\omega} = w$.*
- *A decomposition (u, v) is* normalized *if $\mathcal{Q}(u) = \mathcal{Q}(uv)$.*
- *A normalized decomposition is* duo-normalized *if $\mathcal{P}_u(v) = \mathcal{P}_u(vv)$.*

▶ **Definition 3.2** (Exact, Normalized, and Duo-Normalized acceptance)**.** *Let $\mathcal{F} = (\mathcal{Q}, \{\mathcal{P}_q\}_{q \in Q})$ be an FDFA, $u \in \Sigma^*$, $v \in \Sigma^+$. We define three types of acceptance conditions: We say that $(u, v) \in [\![\mathcal{F}]\!]$ using* exact acceptance *if $v \in [\![\mathcal{P}_u]\!]$. We say that $(u, v) \in [\![\mathcal{F}]\!]$ using* normalized *(resp. duo-normalized) acceptance if there exists a normalized (resp. duo-normalized) decomposition (x, y) of uv^{ω} such that $y \in [\![\mathcal{P}_x]\!]$.*

An FDFA $\mathcal{F}$ is said to be α-*saturated* if for every ultimately periodic word w, all its α-decompositions agree on membership in $\mathcal{F}$. Assuming saturation, and an efficient α-normalization process (as suggested by Claim 3.3) we can alternatively define α-acceptance as in [2] using the efficient procedure that given any (u, v) returns a particular (x, y) that is α-normalized and satisfies $uv^\omega = xy^\omega$. Henceforth, all FDFAs are presumed to be saturated.

▷ Claim 3.3. Let $x \in \Sigma^*$ and $y \in \Sigma^+$. The word xy^ω has an α-decomposition of the form (xy^i, y^j) for i and j quadratic in the size of $\mathcal{F}$ for all $\alpha \in \{\text{exact, normalized, duo-normalized}\}$.

Clearly, if $\mathcal{F}$ is saturated using exact acceptance and it recognizes L then it is also saturated and recognizes L when using normalized acceptance instead. The same is true when going from normalized acceptance to duo-normalized acceptance.

▶ **Corollary 3.4.** *Duo-normalized FDFAs recognize all ω-regular languages.*

▶ **Example 3.5.** Fig. 4.1 (left) shows two FDFAs. The FDFA $\mathcal{F}_1 = (\mathcal{Q}, \{\mathcal{P}_\epsilon, \mathcal{P}_b\})$ has a leading automaton with two states $[\epsilon]$ and $[b]$, and the corresponding progress automata are $\mathcal{P}_\epsilon$ and $\mathcal{P}_b$. Consider the ultimately periodic word a^ω; since (ϵ, a) is a normalized decomposition of a^ω and $a \in [\![\mathcal{P}_\epsilon]\!]$, the word a^ω is accepted by $\mathcal{F}_1$ using normalized acceptance. The FDFA $\mathcal{F}_2 = (\mathcal{Q}, \{\mathcal{P}'_\epsilon, \mathcal{P}_b\})$ uses duo-normalization and $\mathcal{P}'_\epsilon$ as the progress DFA for $[\epsilon]$. The pair (ϵ, aa) is a duo-normalized decomposition of a^ω wrt $\mathcal{F}_2$ and it holds that $a \in [\![\mathcal{P}'_\epsilon]\!]$ thus the word a^ω is accepted by $\mathcal{F}_2$. Observe that the normalized (rather than duo-normalized) decomposition (ϵ, a) is not accepted by $\mathcal{P}'_\epsilon$. In this example the FDFA using duo-normalization has more states. Later on we provide an example where an FDFA using duo-normalization has fewer states, and even exponentially fewer.

▶ **Theorem 3.6.** *The following holds for saturated FDFAs using duo-normalized acceptance: complementation can be computed in constant space; intersection, union and membership can be computed in logarithmic space; emptiness, universality, containment and equivalence can be computed in non-deterministic logarithmic space.*

From now on, unless stated otherwise, we work with duo-normalization. That is, when we say $(u, v) \in [\![\mathcal{F}]\!]$ or $w \in [\![\mathcal{F}]\!]$ we mean according to duo-normalized acceptance condition.

As we later show duo-normalized FDFAs can be exponentially more succinct than all previously defined FDFAs. This is essentially because it considers fewer or more specific decompositions. One might wonder if considering even more specific decompositions will lead to more succinct FDFAs, and can this still be done in the same complexity as for normalized and duo-normalized FDFAs. We come back to this point in the next section, see Prop. 4.3.

4 The Diameter Measure — A Robust Measure on FDFAs

In the following section we define a measure on FDFAs that is tightly related to the inclusion measure of the Wagner hierarchy. The defined measure is robust among FDFAs in the same way that the inclusion measure is robust among DMAs. That is, every pair of FDFAs $\mathcal{F}_1$ and $\mathcal{F}_2$ recognizing the same language agree on this measure.

To devise this measure we would like to understand what the inclusion measure entails on an FDFA for the language. If a DMA has an inclusion chain $S_1 \subset S_2 \subset S_3$ then there is a state q_u in S_1 reachable by some word u from which there are words v_1, v_2, v_3 looping on q_u while traversing the states of S_1, S_2 and S_3, respectively (all and only these states). We term this state a *pivot* state. Assume S_1 is rejecting, then $u(v_1)^\omega \notin L$, $u(v_2)^\omega \in L$ and

$u(v_3)^\omega \notin L$. Since they all loop back to q_u, we have that $u(v_1v_2)^\omega$ also loops in S_2 and is thus accepted and $u(v_1v_2v_3)^\omega$ also loops in S_3 and is thus rejected. Since $v_1 \prec v_1v_2 \prec v_1v_2v_3$ (where $\prec$ denotes the prefix relation) tracing the run on $v_1v_2v_3$ in a progress DFA for u we expect the state reached after v_1 to be rejecting, the one after v_1v_2 to be accepting and the one after $v_1v_2v_3$ to be rejecting. To be precise, we should expect this only if the words v_1, v_1v_2 and $v_1v_2v_3$ are α-normalized, where α is the normalization used by the FDFA.

Let's inspect this on our running example $L_{\infty aa \wedge \neg\infty bb}$ with the DMA in Fig. 3.1 (middle) and the inclusion chain $S_1 = \{q_1, q_2\}$, $S_2 = \{q_1, q_2, q_3\}$, and $S_3 = \{q_1, q_2, q_3, q_4\}$. We can choose q_1 for the pivot state q_u of S_1 and the words $v_1 = ba$, $v_2 = aba$ and $v_3 = abba$, that loop respectively in S_1, S_2 and S_3. Fig. 3.1 (right) provides two FDFAs for this language. The FDFA $\mathcal{F}^{\mathrm{s}}$ uses normalization and $\mathcal{F}^{\circledast}$ uses duo-normalization. Looking at $\mathcal{P}^{\mathrm{s}}_{\epsilon}$, we see that v_1, v_1v_2 and $v_1v_2v_3$ are normalized and the states reached after reading them (q_{ba}, q_{baa}, q_{bb}) are rejecting or accepting as expected. The same is true for $\mathcal{P}^{\circledast}_{\epsilon}$.

Should we entail from this discussion and example that the maximal number of alternations between rejecting and accepting states along any path in an FDFA for a language in $\mathbb{DM}^-_k$ is at most $k-1$? This is true in the progress DFA $\mathcal{P}^{\circledast}_{\epsilon}$, but the progress DFA $\mathcal{P}^{\mathrm{s}}_{\epsilon}$ clearly refutes it, since it has strongly connected accepting and rejecting states (eg, q_a and q_{ab}) and so we can create paths with an unbounded number of alternations between them.

Take such a path with say $k+1$ prefixes $z_1 \prec z_2 \prec \ldots \prec z_{k+1}$ alternating between accepting and rejecting states. Are the words z_i normalized? They can be. Take for instance $z_1 = a$, $z_2 = ab$, and so on ($z_{k+1} = z_k \cdot b$ if k is even and $z_{k+1} = z_k \cdot a$ otherwise).

Can they all be duo-normalized? They can be as we show in Fig. 4.1 (right). It shows a progress DFA $\mathcal{P}_\epsilon$ for an FDFA using duo-normalization and a one-state leading automaton. The FDFA recognizes the language ∞aa. The words $a \prec ab \prec abaa$ are all duo-normalized and reach alternating accepting/rejecting states though the language is in $\mathbb{DM}^-_2$. To fix this issue we introduce the notion of a *persistent decomposition*.

▶ **Definition 4.1** (persistent decomposition wrt an FDFA). *Let $u \in \Sigma^*$, $v \in \Sigma^+$. Let $\mathcal{F} = (\mathcal{Q}, \{\mathcal{P}_q\}_{q\in Q})$ be an FDFA. A duo-normalized decomposition (u, v) is* persistent *if for every $z \in \Sigma^*$ there exists $i > 1$ such that $\mathcal{P}_u(zv) = \mathcal{P}_u(zv^i)$.*

As in Claim 3.3 there exist i and j quadratic in the size of $\mathcal{F}$ such that (uv^i, v^j) is persistent.

▷ Claim 4.2. For every $x \in \Sigma^*$ and $y \in \Sigma^+$ the word xy^ω has a persistent decomposition of the form (xy^i, y^j) where i and j are of size quadratic in $\mathcal{F}$.

One might try to define an additional acceptance condition using the persistent decomposition as was done above, but as stated by the following claim this is futile.

▶ **Proposition 4.3.** *Every FDFA defined with persistent acceptance recognizes the same language when defined with duo-normalized acceptance instead.*

Back to the issue of finding the position on the Wanger hierarchy, we can look for chains of prefixes with alternating acceptance such that each prefix in the chain is persistent.

▶ **Definition 4.4** (Persistent Chain). *Let $\mathcal{F}$ be an FDFA and $u \in \Sigma^*$. We say that $v_1 \prec v_2 \prec \ldots \prec v_k$ for $v_i \in \Sigma^*$ is a u-*persistent chain *of length k wrt $\mathcal{F}$ if (u, v_i) is persistent for every $1 \leq i \leq k$ and $(u, v_{i+1}) \in [\![\mathcal{F}]\!]$ iff $(u, v_i) \notin [\![\mathcal{F}]\!]$ for every $1 \leq i < k$. We say the chain is* positive *(resp.* negative*) if $(u, v_1) \in [\![\mathcal{F}]\!]$ (resp. $(u, v_1) \notin [\![\mathcal{F}]\!]$). We use simply* persistent chain *when u is clear from the context.*

We can now state the measure on FDFAs that relates them to the Wagner hierarchy.

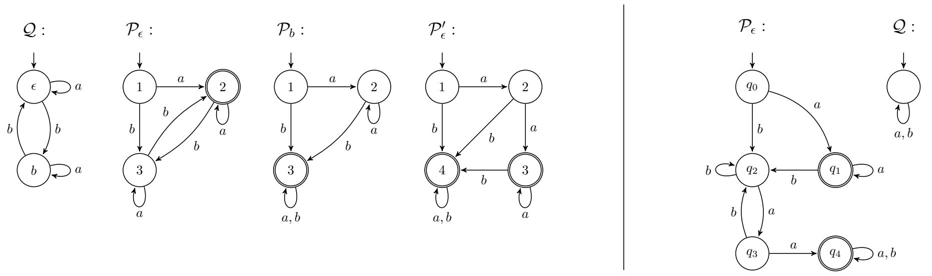

Figure 4.1 Left: Two FDFAs $\mathcal{F}_1 = (\mathcal{Q}, \{\mathcal{P}_\epsilon, \mathcal{P}_b\})$ and $\mathcal{F}_2 = (\mathcal{Q}, \{\mathcal{P}'_\epsilon, \mathcal{P}_b\})$ for the language $(\Sigma^*b)^\omega \cup (bb)^*a^\omega$ using normalized and duo-normalized acceptances, respectively. **Right:** The progress DFA $\mathcal{P}_\epsilon$ for an FDFA accepting ∞aa that uses duo-normalization and a one-state leading automaton.

▶ **Definition 4.5** (The Diameter Measure). *Let $\mathcal{F}$ be an FDFA and $u \in \Sigma^*$. We define the* positive *(resp.* negative*)* diameter measure *of the progress DFA $\mathcal{P}_u$, denoted $|\mathcal{P}_u|^+_\rightsquigarrow$ (resp. $|\mathcal{P}_u|^-_\rightsquigarrow$), as the maximal k for which there exists a* positive *(resp. negative) persistent chain of length k in $\mathcal{P}_u$. We define $|\mathcal{F}|^+_\rightsquigarrow$ as $\max\{|\mathcal{P}_q|^+_\rightsquigarrow : q \in Q\}$ and $|\mathcal{F}|^-_\rightsquigarrow$ as $\max\{|\mathcal{P}_q|^-_\rightsquigarrow : q \in Q\}$.*

We show that the diameter measure is robust among all FDFAs for the language by relating it to the Wagner hierarchy as formally stated below.

▶ **Theorem 4.6** (Correlation to Wagner's hierarchy). *Let $\mathcal{F}$ be an FDFA using any of the acceptance types $\alpha \in \{$exact, normalized, duo-normalized$\}$.*

- $[\![\mathcal{F}]\!] \in \mathbb{DM}^\pm_k$ *iff* $|\mathcal{F}|^+_\rightsquigarrow \leq k$ *and* $|\mathcal{F}|^-_\rightsquigarrow \leq k$
- $[\![\mathcal{F}]\!] \in \mathbb{DM}^+_k$ *iff* $|\mathcal{F}|^+_\rightsquigarrow \leq k$ *and* $|\mathcal{F}|^-_\rightsquigarrow < k$
- $[\![\mathcal{F}]\!] \in \mathbb{DM}^-_k$ *iff* $|\mathcal{F}|^-_\rightsquigarrow < k$ *and* $|\mathcal{F}|^-_\rightsquigarrow \leq k$

In the proof of Thm. 4.6, given a persistent chain $v_1 \prec v_2 \prec \ldots \prec v_k$ in $\mathcal{P}_u$ in some FDFA, we would like to find an inclusion chain of length k in some DMA recognizing the same language. The SCCs visited infinitely often by the words $u(v_1)^\omega, \ldots, u(v_k)^\omega$ might not correspond to an inclusion chain in the DMA. Roughly speaking, to obtain a persistent chain for which this does hold, we make sure every element of the chain has already reached its final SCC and traversed it. Using the following lemma we can create such a persistent chain.

▶ **Lemma 4.7** (Pumping Persistent Periods). *Let $\mathcal{A}$ be an automaton and let $v \in \Sigma^+$ be $\mathcal{A}$-persistent.*[2] *For every $n \in \mathbb{N}$ there exists $l \geq n$ such that for every extension $z \in \Sigma^*$, if vz is $\mathcal{A}$-persistent then $v^l z$ is also $\mathcal{A}$-persistent.*

Following [10] a word $v \in \Sigma^*$ is said to be a *suffix-invariant* of $u \in \Sigma^*$ (in short *u-invariant*) with respect to L if $u \sim_L uv$. That is, no suffix distinguishes between u and the word obtained by concatenating v to u.

Proof of Thm. 4.6. We prove the claim regarding the positive measure. The claim regarding the negative measure is proven symmetrically. We show that

1. $|L|^+_\subset \geq k$ implies $|\mathcal{F}|^+_\rightsquigarrow \geq k$ and
2. $|\mathcal{F}|^+_\rightsquigarrow \geq k$ implies $|L|^+_\subset \geq k$.

The two claims together entail that $|\mathcal{F}|^+_\rightsquigarrow = |L|^+_\subset$.

[2] We say that v is $\mathcal{A}$-persistent if $\mathcal{A}(v) = \mathcal{A}(vv)$ and for every $z \in \Sigma^*$ there exists an $i > 1$ such that $\mathcal{A}(zv) = \mathcal{A}(zv^i)$.

1. We start by showing that if $|L|^+_\subset \geq k$ then $|\mathcal{F}|^+_\rightsquigarrow \geq k$. Let $\mathcal{M}$ be a DMA for L. From $|L|^+_\subset \geq k$ we know that there exists an MSCC of $\mathcal{M}$ subsuming SCCs $S_1, S_2, \ldots, S_k$ such that $S_1 \subsetneq S_2 \subsetneq \ldots \subsetneq S_k$ and S_i is an accepting component if and only if i is odd. Pick a state s in S_1. For $1 \leq i \leq k$ let v_i be a word that loops on s while traversing all the states of S_i and no other states. Let u be a word reaching s from the initial state. Consider the progress DFA of u, $\mathcal{P}_u$. By Claim 4.2 there exists l_1 such that $y_1 = (v_1)^{l_1}$ is $\mathcal{P}_u$-persistent. Similarly, there exists l_2 such that $y_2 = (y_1 v_2)^{l_2}$ is $\mathcal{P}_u$-persistent. In the same way we define $y_i = (y_{i-1} v_i)^{l_i}$ for all $i \in [2..k]$. Consider the words $w_i = u(y_i)^\omega$ for $i \in [1..k]$. Since the set of states visited infinitely often when reading w_i is exactly S_i, it follows that w_i is in L if and only if i is odd. Since all the infixes y_i loop back to s it follows that all the y_i's are u-invariants and thus the (u, y_i) are persistent in $\mathcal{F}$. Since $y_1 \prec y_2 \prec \ldots \prec y_k$ we have found a positive alternating persistent chain in $\mathcal{P}_u$ of length k. Hence, $|\mathcal{P}_u|^+_\rightsquigarrow \geq k$, which in turn gives that $|\mathcal{F}|^+_\rightsquigarrow \geq k$.
2. Next we show that if $|\mathcal{F}|^+_\rightsquigarrow \geq k$ then $|L|^+_\subset \geq k$. Let u be such that $|\mathcal{P}_u|^+_\rightsquigarrow \geq k$. Then there exists a persistent chain $v_1 \prec v_1 v_2 \prec \ldots \prec v_1 v_2 \cdots v_k$ of length k in $\mathcal{P}_u$, starting with an accepting state. For $i \in [1..k]$ let q_i be the state reached after reading $v_1 v_2 \cdots v_i$. Note that q_i is accepting iff i is odd.
   Let $\mathcal{M}$ be a DMA for L and let n be its number of states. Let $l_1 \geq n$ be the number promised by Lemma 4.7 for v_1. Consider $(v_1)^{l_1} v_2$. As v_1 and $v_1 v_2$ are both $\mathcal{P}_u$-persistent, it follows from the lemma that $(v_1)^{l_1} v_2$ is $\mathcal{P}_u$-persistent as well. Since v_1 is $\mathcal{P}_u$-persistent it loops on q_1 and it holds that $(v_1)^{l_1} v_2$ reaches and loops on q_2. Continuing in the same manner, let $y_1 = v_1$ and $y_i = (y_{i-1})^{l_{i-1}} \cdot v_i$ for $i \in [2..k]$ where $l_{i-1} \geq n$ is the number from Lemma 4.7 for y_{i-1}. Then y_i is $\mathcal{P}_u$-persistent, reaching and looping on q_i. Moreover $u(v_1 v_2 \cdots v_i)^\omega \in L$ iff $u(y_i)^\omega \in L$ iff $x(y_i)^\omega \in L$ for any $x \sim_L u$. Let $x_i = u \cdot y_k^n \cdot y_{k-1}^n \cdots y_i^n$ for $i \in [1..k]$. As the v_i's are u-invariant it holds that the x_i's are $\sim_L u$. For $i \in [1..k]$ let $w_i = x_i(y_i)^\omega$. Thus, w_i is in L iff i is odd. For every such i, consider the run of $\mathcal{M}$ on w_i, and let $\mathit{inf}(w_i) = S_i$ be the states of the SCC that $\mathcal{M}$ eventually traverses in. Since n bounds the number of states of $\mathcal{M}$ it follows that after reading x_i the automaton already traversed the SCC S_i and reading y_i again, it will stay in S_i. Because $x_{i-1} = x_i \cdot y_{i-1}^n$ and $y_{i-1}^n \prec y_i$ it holds that $S_1 \subseteq S_2 \subseteq \ldots \subseteq S_k$.
   From the acceptance of the words w_i we conclude S_i is accepting iff i is odd. Therefore the inclusions are strict, namely $S_1 \subsetneq S_2 \subsetneq \ldots \subsetneq S_k$. This proves $|L|^+_\subset \geq k$. ◀

With Thm. 4.6 in place we conclude the robustness of the diameter measure.

▶ **Corollary 4.8.** *Let $\mathcal{F}_1$ and $\mathcal{F}_2$ be two FDFAs recognizing the same language. Then $|\mathcal{F}_1|^+_\rightsquigarrow = |\mathcal{F}_2|^+_\rightsquigarrow$ and $|\mathcal{F}_1|^-_\rightsquigarrow = |\mathcal{F}_2|^-_\rightsquigarrow$.*

Next we show that given an FDFA we can compute its diameter measure in polynomial space. The proof shows that with each progress DFA $\mathcal{P}_u$ we can associate a DFA $\mathbb{P}_u$, which we term the *persistent DFA*. Broadly speaking, $\mathbb{P}_u$ is the product of the leading automaton and copies of the progress automaton, starting from each of its states. The states of $\mathbb{P}_u$ can be classified into *significant* and *insignificant* where significant states are those that are reached by persistent words and only such words. The classification can easily be done by inspecting the "state vector". The persistent DFA need not be built, instead a persistent chain can be non-deterministically guessed and verified in polynomial space.

▶ **Theorem 4.9.** *The diameter measure of a duo-normalized FDFA can be computed in PSPACE.*

Since normalized and exact FDFAs are also duo-normalized, this upper bound holds for them as well. However, in Prop. 4.12 we show that for normalized (and exact) FDFAs the diameter measure can be computed in NLOGSPACE. For the case of duo-normalized FDFAs, we provide a matching lower bound.

▶ **Theorem 4.10.** *The problem of determining whether the diameter measure of a duo-normalized FDFA is at least k is PSPACE-hard.*

Proof sketch. The proof uses a reduction from non-emptiness of intersection of DFAs, which is known to be PSPACE-hard [15]. Let $D_1, D_2, \ldots, D_k$ be k DFAs over Σ. We construct an FDFA with a one-state leading automaton and a progress DFA $\mathcal{P}$ over $\Sigma' = \Sigma \cup \{1, \ldots, k\} \cup \{\sharp\}$ as depicted in Fig. 5.1 (left), see the full version for a complete description. To see that the FDFA is saturated, we show in the full version that not only every two duo-normalized decompositions (u, v) and (u', v') of the same ultimately periodic word w agree on acceptance, but their periods also traverse the same MSCC in $\mathcal{P}$.

We claim that if there is a word $v \in \Sigma$ in the intersection of all D_i's, then $\sharp v \prec \sharp v1 \prec \sharp v12 \prec \sharp v123 \prec \ldots \prec \sharp v12 \cdots k \prec \sharp v12 \cdots k\sharp\sharp$ is a persistent chain in $\mathcal{P}$ of length $k+2$. Let $y_0 = \sharp v$, $y_i = \sharp v12 \cdots i$ for $i \in [1..k]$, and $y_{k+1} = y_k\sharp\sharp$. Then y_i reaches s_i^i, and reading y_i from s_i^i reaches s_i^i again. Thus y_i is duo-normalized and y_i is accepted iff i is even. To see that y_i is persistent it remains to show that from any state q reading y_i and reading $(y_i)^2$ the automaton reaches the same state q' for some q'. Observe that reading y_i from any state s_j^j for $j \leq i$ will still reach s_i^i. If y_i is read from s_j^j for $i < j \leq k+1$ or any other state (as there are no other outgoing $\sharp$-transitions) it will reach s_{k+1}^{k+1} and stay there forever. Thus y_i is persistent for any i. Hence, $y_0 \prec y_1 \prec y_2 \prec \ldots \prec y_k \prec y_{k+1}$ is a persistent chain in $\mathcal{P}$ of length $k+2$.

For the other direction, we claim that if there exists a persistent chain of length $k+2$ in $\mathcal{P}$, then the intersection of all D_i's is non-empty. Let $w_0 \prec w_2 \prec \ldots \prec w_{k+1}$ be such a chain. First, we note that from the structure of $\mathcal{P}$, since the MSCCs corresponding to the D_i's are of alternating acceptance, it follows that w_0 reaches s_0^0, w_{k+1} reaches s_{k+1}^{k+1} and all other w_i reach some state in the i-th MSCC. Second, if reading w_i reaches a state in the i-th MSCC, then since w_i is duo-normalized, reading w_i for the second time must loop back to the same state. For $i \in [1..k]$ this can only occur if w_i is a rotation of $v_1 12 \cdots i\sharp v_2 12 \cdots i\sharp \cdots v_m 12 \cdots i\sharp$ for some $v_1, \ldots, v_m \in D_i$. This is since w_i must contain the letter 1 and it can only be read from final states of D_i. Note that there can't be an infix of any w_i for $1 \leq i \leq k$ where the letter $\sharp$ appears after a letter lower than i, as this would lead from the i-th MSCC to s_{k+1}^{k+1}. As the w_i's form a chain, w_1 is a prefix of all the w_i's for $1 < i \leq k$. It follows that w_1 is exactly of the form $\sharp v1$, and as v is common to all the following w_i's, v is in the intersection of all the D_i's. ◀

The following proposition shows that there exists FDFAs where the smallest elements of a persistent chain are inevitably of exponential length. The idea is to take the construction of the FDFA in the proof of Thm. 4.10 and let the DFA D_i count modulo the i-th prime.

▶ **Proposition 4.11.** *There exists a family of languages $\{L_n\}$ with an FDFA with number of states polynomial in n where any persistent chain of maximal length must include a word of length exponential in n.*

Computing the Wagner position on normalized (rather than duo-normalized) FDFAs, can be done more efficiently, specifically in NLOGSPACE.

▶ **Proposition 4.12.** *The position in the Wagner hierarchy of an FDFA using normalized acceptance can be computed in NLOGSPACE.*

For the sake of the proof we define two sub-classes of normalized FDFAs. The smallest one is a generalization studied in [5] of the syntactic FDFA for a language L [19].

▶ **Definition 4.13** ($\sim$-syntactic FDFA). *Let $L \subseteq \Sigma^\omega$ and let $\sim$ be a right congruence refining the syntactic right congruence $\sim_L$. The* $\sim$-syntactic FDFA*, denoted $(\mathcal{Q}^\sim, \mathcal{P}^\sim_u)$, is defined as follows. The leading automaton $\mathcal{Q}^\sim$ is $\mathcal{A}[\sim]$, and the progress automaton $\mathcal{P}^\sim_u$ is $\mathcal{A}[\approx_u]$ where $x \approx_u y$ if (a) $ux \sim uy$ and (b) for every $z \in \Sigma^*$ it holds that $uxz \sim u$ implies $u(xz)^\omega \in L$ iff $u(yz)^\omega \in L$.*[3]

It is shown in [5, Lemma 21] that if x is duo-normalized wrt $\sim$ and $\approx_u$ then for every $y \approx_u x$ we have that y is also duo-normalized. We can thus refer to a state as being duo-normalized. It is then showed that two duo-normalized states in the same SCC of a $\sim$-syntactic progress DFA agree on acceptance [5, Lemma 23]. These properties allow defining a polynomial procedure that associates with every state of a *$\sim$-syntactic FDFA* a color that tightly correlates to position on the Wagner hierarchy [5].

It follows that on *$\sim$-syntactic FDFAs* the Wagner position can be determined in polynomial time. The proof can be generalized to any FDFA using normalization in which the right congruence $x \approx_u y$ implies $x \sim_L y$. We call such FDFAs *projective FDFAs*.

▶ **Definition 4.14.** *An FDFA $(\mathcal{A}[\sim], \{\mathcal{A}[\approx_u]\})$ is termed* projective *if for every progress DFA, the respective right congruence $\approx_u$ satisfies that $x \approx_u y$ implies $x \sim_L y$.*

Fig. 5.1 (left) depicts the inclusions among these classes. (In the meantime ignore the text in blue, and the arrow; we will come back to this in the next section.) Since any FDFA using normalization can be transformed with a quadratic blowup into a *projective FDFA* (by multiplying the progress DFAs by the leading automaton) we have that the Wagner position on normalized FDFAs can be computed in polynomial time. In the full version of the paper we show that it can also be computed in NLOGSPACE.

5 Succinctness Results

We turn to provide some succinctness results regarding FDFAs with duo-normalized acceptance. The results compare duo-normalized FDFAs with normalized FDFAs, as well as canonical FDFAs using these acceptance conditions. We already mentioned four canonical FDFAs: the periodic FDFA [7] that uses exact acceptance; and the syntactic [19], recurrent [2] and limit FDFAs [17] that use normalized acceptance. A canonical FDFA that uses duo-normalization can be extracted from notions of [5]. We term it the *Colorful FDFA*, since it relies on the notion of natural colors [10].

Loosely speaking, [10] shows that given an ω-regular language L one can associate with every word $w \in \Sigma^\omega$ a natural color. If w is given color k wrt L, then there is no parity automaton for L that would visit a color lower than k infinitely often when reading w. Consider again the language $L_{\infty aa \wedge \neg\infty bb}$ requiring infinitely many aa and finitely many bb for which a DPA is given in Fig. 3.1 (middle). The colors of $(ab)^\omega$, $(a)^\omega$, $(aab)^\omega$, $(aabb)^\omega$ and $(b)^\omega$ are 3, 2, 2, 1 and 1, resp. The intuition is that the color is 1 if bb occurs infinitely often, it is 2 if aa occurs infinitely often but bb occurs only finitely often, and it is 3 if neither aa nor bb occur infinitely often. We use $\mathsf{Color}(w)$ for the natural color of w. Note that the language L is implicit in the notation.

A related definition provided in [5, Def 2.] can be viewed as giving colors for finite words v wrt to an ω-regular language L and an equivalence class $[u]$. We use $\mathsf{Color}_u(v)$ to denote the color of the finite word $v \in \Sigma^+$ wrt a finite word $u \in \Sigma^*$. The definition satisfies that

[3] *Canonical FORC* is the term used in [5] for the FORC underlying the $\sim$-syntactic FDFA.

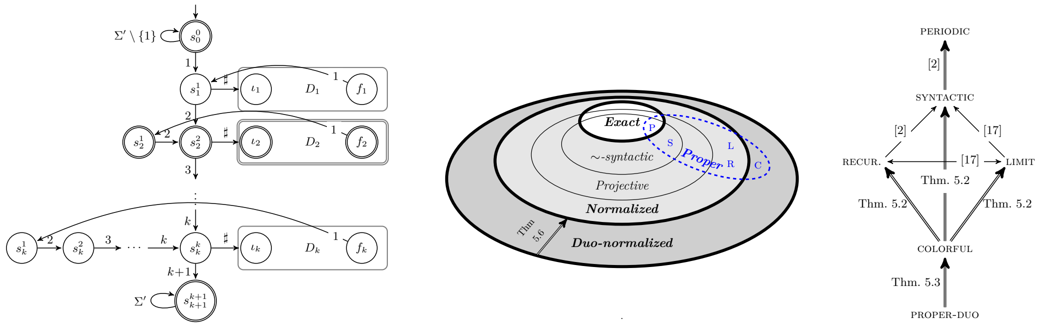

Figure 5.1 Left: The progress DFA $\mathcal{P}$ used in the proof of Thm. 4.10. Its alphabet is $\Sigma' = \Sigma \cup \{1, \ldots, k, k{+}1\} \cup \{\sharp\}$ where Σ is the alphabet of the DFAs $D_1, \ldots, D_k$. Transitions to the sink state s^{k+1}_{k+1} are omitted. The figure assumes k is odd; for $j \in [1..k]$, ι_j is initial and f_j is accepting. **Middle:** The inclusions among these classes of FDFAs (in black), as well as the placement of the canonical FDFAs in these classes (in blue). The letters P,S,R,L,C abbreviate PERIODIC, SYNTACTIC, RECURRENT, LIMIT, and COLORFUL, resp. **Right:** Picture summarizing succinctness results on proper FDFAs. A double-line (resp. one-line) arrow from C to D indicates that C can be exponentially (resp. quadratically) more succinct than D.

$\mathsf{Color}_u(v)$ returns $\max\{\mathsf{Color}(u(vz)^\omega) \mid z \in \Sigma^*\}$. Note that it is possible that $u(v')^\omega = u(v'')^\omega$ for some $u \in \Sigma^*$ and $v', v'' \in \Sigma^+$, though $\mathsf{Color}_u(v') \neq \mathsf{Color}_u(v'')$. Indeed in the example of $L_{\infty aa \wedge \neg\infty bb}$ we have $(a)^\omega = (aa)^\omega$, yet $\mathsf{Color}_\epsilon(a) = 3$ while $\mathsf{Color}_\epsilon(aa) = 2$.

The reason is that if the period contains a followed by some z the resulting color may be 3 or 2 or 1 while if the period contains aa followed by some z the color can be 2 or 1 but it cannot be 3 since aa surely occurs infinitely often.

These colors can be used to define equivalence classes $\approx^{\circledast}_u$ for each word $u \in \Sigma^*$ by differentiating between words x and y if there is a word z such that the respective extensions xz and yz disagree on the color (wrt u).[4] The Colorful FDFA uses the automaton for $\sim_L$ for the leading automaton as the other canonical FDFAs do. For the progress DFA for u it takes a DFA whose automaton structure is derived by the equivalence relation $\approx^{\circledast}_u$, and the accepting states are those with an even color. The acceptance type of the Colorful FDFA is duo-normalization.

▶ **Definition 5.1** (The Colorful FDFA). *Let $u, x, y \in \Sigma^*$. We define $x \approx^{\circledast}_u y$ if for every $z \in \Sigma^*$ we have $\mathsf{Color}_u(xz) = \mathsf{Color}_u(yz)$. The* colorful FDFA *for a language L, denoted $\mathcal{F}^{\circledast}(L)$, uses duo-normalized acceptance and consists of $(\mathcal{A}[\sim_L], \{\mathcal{P}^{\circledast}_u\})$ where $\mathcal{P}^{\circledast}_u$ is a DFA with the automaton structure $\mathcal{A}[\approx^{\circledast}_u]$ where state q_v is accepting if $\mathsf{color}_u(v)$ is even.*

The Colorful FDFA $\mathcal{F}^{\circledast} = (\mathcal{Q}, \{\mathcal{P}^{\circledast}_\epsilon\})$ for our running example $L_{\infty aa \wedge \neg\infty bb}$ is given in Fig. 3.1 (right).

We are now ready to discuss the succinctness results. Recall that the canonical FDFAs using normalized acceptance have been shown to be exponentially more succinct than the canonical model using exact acceptance [2]. We first show that using duo-normalization a similar succinctness gain is achieved, namely that the Colorful FDFA can be exponentially more succinct than all other canonical representations.

[4] These equivalence classes correspond to the precise family of weak priority mappings wrt to $\sim_L$ from [5, Def. 17].

▶ **Theorem 5.2.** *The Colorful FDFA can be exponentially more succinct than the syntactic/recurrent/limit FDFA.*

The proof uses the family of languages $\{L_n\}_{n\in\mathbb{N}}$ over $\Sigma = \{a, b, \langle, \rangle\}$ defined as follows.

$$L_n = \left\{ w \in \{a,b,\langle,\rangle\}^\omega \;\middle|\; \begin{array}{l} w \text{ has inf. many occurrences of } \langle a^k b^m \rangle \text{ for some } k \in [1..n] \\ \text{and } m \text{ that is divisible by the } k\text{-th prime.} \end{array} \right\}$$

The idea is that using duo-normalization the Colorful FDFA can look only for prefixes of the form $\langle a^k b^m \rangle$, whereas the syntactic/recurrent/limit FDFA must also answer correctly for prefixes of the form $b^m\rangle\langle a^k$ due to using normalized acceptance. Recognizing prefixes of the latter form is much harder.

Next we show that the Colorful FDFA is not the most succinct among the duo-normalized FDFAs. The essence of the proof is that the Colorful FDFA must keep track of all infixes of interest seen in order to maintain the real color of the word. On the other hand, a duo-normalized FDFA, similarly to a DBA, can choose an arbitrary order to look for such infixes. The proof uses the family $\{L'_n\}_{n\in\mathbb{N}}$ over $\Sigma = \{a_1, \ldots, a_n\}$ defined as follows.[5]

$$L'_n = \{w \in \{a_1, \ldots, a_n\}^\omega \mid \text{for all } i \in [1..n] \text{ the letter } a_i \text{ appears inf. often in } w\}$$

▶ **Theorem 5.3.** *Duo-normalized FDFAs can be exponentially more succinct than the Colorful FDFA.*

An FDFA in general can use any leading automaton $\mathcal{A}[\sim]$ for a right congruence $\sim$ that refines $\sim_L$. We note that all the canonical models (the periodic, syntactic, recurrent, limit and colorful) use $\mathcal{A}[\sim_L]$ for the leading automaton, we term such FDFAs, *proper*.

▶ **Definition 5.4** (Proper FDFAs). *An FDFA recognizing a language L is termed* proper *if its leading automaton is $\mathcal{A}[\sim_L]$.*

The FDFA used in the proof of Thm. 5.3 is proper. We can thus strengthen the claim as follows.

▶ **Corollary 5.5.** *Proper duo-normalized FDFAs can be exponentially more succinct than the Colorful FDFA.*

Surprisingly, Klarlund has shown that non-proper normalized FDFAs may be exponentially more succinct than proper normalized FDFAs [14]. One may thus wonder if non-proper normalized FDFAs can be as succinct as duo-normalized FDFAs. That is, if duo-normalization adds succinctness when considering non-proper FDFAs. The following theorem shows that duo-normalization does add succinctness even relative to non-proper normalized FDFAs (and even when limiting duo-normalized FDFAs to proper ones).

▶ **Theorem 5.6.** *(Proper) duo-normalized FDFAs can be exponentially more succinct than (not necessarily proper) normalized FDFAs.*

The proof uses the following family of languages $\{L''_n\}_{n\in\mathbb{N}}$ over $\Sigma = \Sigma_a \cup \Sigma_s$ where $\Sigma_a = \{a_1, \ldots, a_n\}$ and $\Sigma_s = \{s_1, \ldots, s_n\}$.

$$L''_n = \left\{ w \in (\Sigma^*\Sigma_a)^\omega \;\middle|\; \begin{array}{l} \text{Let } m = \max\{j \mid a_j \in \Sigma_a \cap \mathit{inf}(w)\}. \\ \text{Then } s_m \in \Sigma_s \text{ appears inf. often in } w. \end{array} \right\}.$$

[5] This family is used in [4] to show DBAs can be exponentially more succinct than combinations of DFAs.

The challenge in the language can be observed in periods where s_m occurs before a_m was seen (for m being the maximal index of an a_i letter in the period). The duo-normalized FDFA has the privilege of looking for duo-normalized decompositions in which s_m is observed after the maximal a_m is seen. As the normalized FDFA has to consider all prefixes, specifically prefixes for every subset of Σ_s, it must grow to an exponential size.

6 Discussion

We have shown that FDFAs with duo-normalized acceptance can be exponentially more succinct than FDFAs using (standard) normalization. At the same time the common operations procedures and decision problems on them can still be done in NLOGSPACE. Fig. 5.1 (right) summarizes the results regarding succinctness among the canonical FDFAs suggested thus far. It shows that the Colorful FDFA can be exponentially more succinct than all other canonical models. At the same time, a minimal duo-normalized FDFA can be exponentially more succinct than the Colorful FDFA.

The figure might raise the question whether a duo-normalized FDFA can be doubly-exponentially more succinct than the periodic FDFA (the least succinct canonical representation). However this cannot be since a duo-normalized FDFA can be translated into a non-deterministic Büchi automaton (NBA) using exactly the same procedure as the one transforming a normalized FDFA into an NBA [1]. The reason is that the construction actually looks for a duo-normalized decomposition (which by saturation exists).

▶ **Proposition 6.1.** *If L has a duo-normalized FDFA $\mathcal{F}$ then it has an NBA of size polynomial in the number of states of $\mathcal{F}$.*

Since an NBA can be converted into a periodic FDFA in an exponential blowup [7, 16] we get an overall exponential translation from duo-normalized FDFAs to the periodic FDFA, showing no doubly-exponential lower bound can be achieved. Since NBAs can be converted to DPAs with an exponential blow up [24, 23, 25, 11] and DPAs can be polynomially converted into non-proper FDFAs [1] we can conclude the following.

▶ **Corollary 6.2.** *The periodic FDFA and the minimal (non-proper) normalized FDFAs and DPAs of a language L are at most exponentially larger than a duo-normalized FDFA for L.*

We also answer a question posed by [10] regarding the relation of the structure of an FDFA to its position in the Wagner hierarchy. Specifically, we have provided a measure on FDFAs that corresponds to the Wagner hierarchy. We have shown that its computation is PSPACE-complete for duo-normalized FDFAs, and is in NLOGSPACE for normalized FDFAs. The measure is based on the notion of a persistent chain. Since the Wagner hierarchy correlates to the minimal color required by a parity automaton, we can define a notion of chains that relates to natural colors of [10, 5], and is thus a semantic notion (defined wrt to the language regardless of a particular acceptor for it). In the full version we show that the existence of one type of chain implies the existence of the other type of chain, and vice versa.

The notion of duo-normalization decomposition seems to be related to the notion of a *linked-pair* in ω-semigroups and Wilke-algebras [27, 9], in which (s, e) is a linked pair if $se = s$ and e is idempotent. It seems that there are two main differences; the first is that a linked pair relates to one relation $\sim$, while a duo-normalized decomposition relates to a pair of relations $(\sim, \approx)$ (one for the leading automaton and one for the respective progress DFA). The second is that the relation $\sim$ used in Wilke-algebras is a two-sided congruence, while the relations used by FDFAs are one-sided. Both differences suggest that duo-normalized FDFAs would be more succinct, but this deserves further study.

References

1 D. Angluin, U. Boker, and D. Fisman. Families of DFAs as acceptors of ω-regular languages. *Log. Methods Comput. Sci.*, 14(1), 2018.

2 D. Angluin and D. Fisman. Learning regular omega languages. *Theor. Comput. Sci.*, 650:57–72, 2016.

3 R. Bloem, B. Jobstmann, N. Piterman, A. Pnueli, and Y. Sa'ar. Synthesis of reactive(1) designs. *J. Comput. Syst. Sci.*, 78(3):911–938, 2012.

4 L. Bohn and C. Löding. Passive learning of deterministic Büchi automata by combinations of DFAs. In *49th International Colloquium on Automata, Languages, and Programming, ICALP 2022, July 4-8, 2022, Paris, France*, pages 114:1–114:20, 2022.

5 L. Bohn and C. Löding. Constructing deterministic parity automata from positive and negative examples. arXiv:2302.11043. *In print for TheoretiCS, accepted on: 2024-05-11*, 2024.

6 Büchi J. R. On a decision method in restricted second order arithmetic. In *Int. Congress on Logic, Method, and Philosophy of Science*, pages 1–12. Stanford University Press, 1962.

7 H. Calbrix, M. Nivat, and A. Podelski. Ultimately periodic words of rational w-languages. In *9th Inter. Conf. on Mathematical Foundations of Programming Semantics (MFPS)*, pages 554–566, 1993.

8 O. Carton and R. Maceiras. Computing the Rabin index of a parity automaton. *RAIRO Theor. Informatics Appl.*, 33(6):495–506, 1999.

9 O. Carton, D. Perrin, and J-E. Pin. Automata and semigroups recognizing infinite words. In *Logic and Automata: History and Perspectives [in Honor of Wolfgang Thomas]*, pages 133–168, 2008.

10 R. Ehlers and S. Schewe. Natural colors of infinite words. In *42nd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS*, pages 36:1–36:17, 2022.

11 D. Fisman and Y. Lustig. A modular approach for Büchi determinization. In *26th International Conference on Concurrency Theory, CONCUR 2015*, pages 368–382, 2015.

12 Dana Fisman, Emmanuel Goldberg, and Oded Zimerman. A robust measure on FDFAs following duo-normalized acceptance. *arXiv*, 2023. doi:10.48550/arXiv.2310.16022.

13 M. Jurdzinski. Small progress measures for solving parity games. In *STACS 2000, 17th Annual Symposium on Theoretical Aspects of Computer Science*, pages 290–301, 2000.

14 N. Klarlund. A homomorphism concepts for omega-regularity. In *Computer Science Logic, 8th International Workshop, CSL*, pages 471–485, 1994.

15 D. Kozen. Lower bounds for natural proof systems. In *18th Annual Symposium on Foundations of Computer Science*, pages 254–266, 1977.

16 D. Kuperberg, L. Pinault, and D. Pous. Coinductive algorithms for Büchi automata. *Fundam. Informaticae*, 180(4):351–373, 2021.

17 Y. Li, S. Schewe, and Q. Tang. A novel family of finite automata for recognizing and learning ømega-regular languages. In *Automated Technology for Verification and Analysis - 21st International Symposium, ATVA*, pages 53–73, 2023.

18 Y. Li, X. Sun, A. Turrini, Y-F. Chen, and J. Xu. ROLL 1.0: ω -regular language learning library. In *Tools and Algorithms for the Construction and Analysis of Systems - 25th International Conference, TACAS*, pages 365–371, 2019.

19 O. Maler and L. Staiger. On syntactic congruences for omega-languages. *Theor. Comput. Sci.*, 183(1):93–112, 1997.

20 J. Myhill. Finite automata and the representation of events. Technical report, Wright Patterson AFB, Ohio, 1957.

21 A. Nerode. Linear automaton transformations. In *Proceedings of the American Mathematical Society, 9(4)*, pages 541–544, 1958.

22 D. Perrin and J-E Pin. *Infinite words – Automata, semigroups, logic and games*, volume 141 of *Pure and applied mathematics series.* Elsevier Morgan Kaufmann, 2004.

23 N. Piterman. From nondeterministic Büchi and Streett automata to deterministic parity automata. In *21th IEEE Symposium on Logic in Computer Science LICS*, pages 255–264, 2006.

24 S. Safra. On the complexity of omega-automata. In *29th Annual Symposium on Foundations of Computer Science, White Plains*, pages 319–327, 1988.

25 S. Schewe. Büchi complementation made tight. In *26th International Symposium on Theoretical Aspects of Computer Science*, pages 661–672, 2009.

26 K. W. Wagner. A hierarchy of regular sequence sets. In *Mathematical Foundations of Computer Science 1975, 4th Symposium, MFCS*, pages 445–449, 1975.

27 T. Wilke. An Eilenberg theorem for infinity-languages. In *Automata, Languages and Programming, 18th International Colloquium, ICALP*, pages 588–599, 1991.

28 T. Wilke and H. Yoo. Computing the rabin index of a regular language of infinite words. *Inf. Comput.*, 130(1):61–70, 1996.

29 W. Zielonka. Infinite games on finitely coloured graphs with applications to automata on infinite trees. *Theor. Comput. Sci.*, 200(1-2):135–183, 1998.