ON THE DISSIMILARITY OPERATION ON FINITE LANGUAGES

Cezar Câmpeanu[(A)] Nelma Moreira[(B)]
Rogério Reis[(B)]

[(A)]School of Mathematical and Computational Sciences
The University of Prince Edward Island, Charlottetown, PE, Canada
ccampeanu@upei.ca

[(B)] CMUP & DCC, Faculdade de Ciências da Universidade do Porto, Porto, Portugal
{nam,rvr}@dcc.fc.up.pt

Abstract
The distinguishability language of a regular language L is the set of words distinguishing between pairs of words under the Myhill-Nerode equivalence induced by L, i.e., between pairs of distinct left quotients of L. The similarity relation induced by a language L is a similarity relation inspired by the Myhill-Nerode equivalence and it was used to obtain compact representation of automata for a finite language L, i.e., deterministic finite cover automata, which are deterministic finite automata accepting all the words of L and possibly some other words that are longer than any word of L. The dissimilarity language of a finite language L is defined as the set of words that separate a pair of words which are not similar w.r.t. to a (finite) language L. In this paper we extend the study of distinguishability operation on regular languages to l-dissimilarity, for $l \in \mathbb{N}$, and the dissimilarity operation on finite languages. We examine their properties, the state complexity, and relations that can be established between these operations.

1. Introduction

The distinguishability language of a regular language L is the set of words distinguishing between pairs of words under the Myhill-Nerode equivalence induced by L, i.e., between pairs of distinct left quotients of L. The distinguishability operation was introduced by Câmpeanu *et al.* [5], where they proved that this operation is suffix-closed, it has a fixed point under iteration, and its state complexity is $2^n - n$, where n is the state complexity of L. This later bound can be reached by using the universal witness of Brzozowski [1]. The number of elements of the set of minimal words that can distinguish all left quotients is at most $n - 1$ and this bound is reached. The computational complexity of several decision problems for the distinguishability

[(B)]Authors partially funded by CMUP (UID/MAT/00144/2013), which is funded by FCT (Portugal) with national (MCTES) and European structural funds through the programs FEDER, under the partnership agreement PT2020.

language was studied by Holzer and Jakobi [11], and they give a complete characterization of the synchronizing languages in terms of fixed points of the distinguishability operator, i.e., all synchronizing languages are fixed points of the distinguishability operator. Thus, the new operation of distinguishability, not only that preserves regularity, but it allows us to characterize classes of languages as a fixed points of an algebraic operation.

The similarity relation induced by a language L is inspired also by the Myhill-Nerode equivalence, although it is not an equivalence relation [8, 9, 10, 13]. Câmpeanu *et al.* [8] used this similarity relation to obtain a more compact cover automaton for a finite language L, i.e., a deterministic finite automaton accepting all the words of L and possibly some other words that are longer than any word of L. The number of states of minimal cover automata can be smaller than the state complexity of L, never exceeding this bound. These automata and their minimization algorithms were studied by various authors [3, 6, 7, 12, 14]. The dissimilarity language of a finite language L is defined as the set of words that separate some pair of words that are not similar w.r.t. to a finite language L. It is just natural to ask if dissimilarity operation shares the same properties with distinguishability operation. We study what would be the relation between these two operations and what are the properties of the dissimilarity operation. We investigate if some of the properties of distinguishability operation will also hold for dissimilarity (closures, fix-points under iteration, state complexity etc).

We introduce the dissimilarity operation and the set of words that distinguishes dissimilar words in Section 3., and study general properties in Section 4.. We determine upper-bounds and lower-bounds for the state complexity of dissimilarity and distinguishability on finite languages in Section 5.. Distinguishability coincides with suffix closure on finite languages, thus we also obtain the state complexity for this closure. We analyze the relation between the minimal DFA, dissimilarity operations, distinguishability operations and a minimal DFCA, and give several examples for the behaviour of the dissimilarity operation in Section 6.. Conclusions and further research directions are presented in Section 7..

2. Notations and Definitions

We denote the size of a set T by $|T|$. An *alphabet* Σ is a finite non-empty set, and the free monoid generated by Σ is $\Sigma^\star$. A *language* is any set $L \subseteq \Sigma^\star$.

For a word $w = a_1 \cdots a_n \in \Sigma^\star$, $a_i \in \Sigma$, $1 \leq i \leq n$, $n = |w|$ is the *length* of w. For the case where $n = 0$, we denote the resulting empty word by ε. The set of all words with length at most l ($l \geq 0$) is denoted by $\Sigma^{\leq l}$. If $w = ux$ for some $u, x \in \Sigma^\star$, then x is called a *suffix* of w and u is a *prefix* of w. Let $\mathsf{suff}(L)$, $\mathsf{pref}(L)$ denote the set of all suffixes, respectively prefixes, of a language L.

If we consider an order over Σ, then the quasi-lexicographical order on $\Sigma^\star$ is defined as follows: $w \preceq_l w'$ if $|w| < |w'|$, or if $|w| = |w'|$ and w lexicographically precedes w'.

A *deterministic finite automaton* (DFA) is a quintuple $\mathcal{A} = \langle Q, \Sigma, q_0, \delta, F \rangle$, where Q is a

finite set of states, Σ is the alphabet, $q_0 \in Q$ is the initial state, $F \subseteq Q$ is the set of final states, and $\delta : Q \times \Sigma \longrightarrow Q$ is the transition function. A *reduced* DFA is a DFA with all states reachable from the initial state (*accessible*), and such that all states can reach a final state (*useful*), with the possible exception of one that is a *sink* state or *dead* state (here named Ω), i.e., a non-final state where all output transitions are self loops. The transition function δ can be naturally extended to words. The language recognized by a DFA $\mathcal{A}$ is $\mathcal{L}(\mathcal{A}) = \{w \in \Sigma^\star \mid \delta(q_0, w) \in F\}$. We denote by L_q and R_q the *left* and *right* languages of q, respectively, i.e., $L_q = \{w \mid \delta(q_0, w) = q\}$, and $R_q = \{w \mid \delta(q, w) \in F\}$.

The minimal word in quasi-lexicographical order that reaches state $q \in Q$ is $x_{\mathcal{A}}(q)$. The length of $x_{\mathcal{A}}(q)$ is the *level* of $q \in Q$, i.e., $\mathrm{level}(q) = \min\{|w| \mid \delta(q_0, w) = q\}$. A regular language L induces on $\Sigma^\star$ the Myhill-Nerode equivalence relation: $x \equiv_L y$ if, for all $w \in \Sigma^\star$, we have that $xw \in L$ if and only if $yw \in L$. The *left quotient*, or simply *quotient*, of a regular language L by a word w is the language $w^{-1}L = \{x \mid wx \in L\}$. A quotient corresponds to an equivalence class of $\equiv_L$, i.e., two words are equivalent if and only if their quotients are the same. If a language L is regular, the number of distinct left quotients is finite, and it is exactly the number of states in the *minimal* DFA recognizing L. This number is called the *state complexity* of L, and is denoted by $sc(L)$. In a minimal DFA, for each $q \in Q$, R_q is exactly a quotient. If some quotient of a language L is $\emptyset$, the minimal DFA of L has a dead state. The state complexity of an operation on regular languages is the worst-case state complexity of a language resulting from that operation, as a function of the state complexities of the operands.

Let L be a finite language. The *rank* of L is the length of the longest word in L, $l = rank(L)$. Let $\mathcal{A} = \langle Q, \Sigma, q_0, \delta, F\rangle$ be a DFA recognizing L and possibly other words of length (strictly) greater than l. Then $\mathcal{A}$ is a *deterministic finite cover automaton* (DFCA) for L. A *minimal* DFCA of a language L is a DFCA of L with minimal number of states. We call this number the *cover state complexity* of L and denote it by $csc(L)$. For an arbitrary language L and $l \in \mathbb{N}$, the language $L \cap \Sigma^{\leq l}$ is always finite, and a DFCA for $L \cap \Sigma^{\leq l}$ is called l-DFCA for L.

We now recall the definition of distinguishability and minimal distinguishable words introduced by Câmpeanu *et al.* [5]. Given $x, y \in \Sigma^\star$, the language that distinguishes x from y w.r.t. L is

$$\mathsf{D}_L(x, y) = \{w \mid xw \in L \not\Leftrightarrow yw \in L\}\,. \tag{1}$$

Naturally, we define the *distinguishability language* of L by

$$\mathsf{D}(L) = \{w \mid \exists x, y \in \Sigma^\star\ (xw \in L \text{ and } yw \notin L)\}. \tag{2}$$

It is immediate that

$$\mathsf{D}(L) = \bigcup_{x,y \in \Sigma^\star} \mathsf{D}_L(x, y) = \mathsf{suff}(L) \cap \mathsf{suff}(\overline{L}),$$

i.e., $\mathsf{D}(L)$ is suffix closed. In particular, if L is finite, then $\mathsf{D}(L) = \mathsf{suff}(L)$. It was also shown that the iteration of D always reaches a fixed point, i.e., for any regular language L, $\mathsf{D}^3(L) = \mathsf{D}^2(L)$. Moreover, $\mathsf{D}(L) = L$ if and only if L is suffix-closed and has $\emptyset$ as one of its quotients. Holzer and Jakobi noted that if $\mathsf{D}^2(L) = \mathsf{D}(L)$, then L is accepted by a synchronizing

DFA $\mathcal{A}$, i.e., such that there exists a word $w \in \Sigma^\star$ which leaves the automaton in one particular state independently of the starting state.

If $x, y \in \Sigma^\star$ and $x \not\equiv_L y$, we define

$$\underline{\mathsf{D}}_L(x,y) = \min\{w \mid w \in \mathsf{D}_L(x,y)\}, \tag{3}$$

where minimum is considered with respect to the quasi-lexicographical order. In case $x \equiv_L y$, $\underline{\mathsf{D}}_L(x,y)$ is undefined. We can observe that if $x \not\equiv_L y$, $\underline{\mathsf{D}}_L(x,y) = \min(x^{-1}L\Delta y^{-1}L)$, where Δ is the symmetric difference of operands. The set of minimal words distinguishing quotients of a language L is

$$\underline{\mathsf{D}}(L) = \{\underline{\mathsf{D}}_L(x,y) \mid x, y \in \Sigma^*, x \not\equiv_L y\}. \tag{4}$$

The language $\underline{\mathsf{D}}(L)$ is also suffix-closed and $|\underline{\mathsf{D}}(L)| \leq n-1$, where $n \geq 2$ is the state complexity of L.

3. The Dissimilarity Operation

According to Dwork and Stockmeyer [10], every language $L \subseteq \Sigma^\star$ and number $l \in \mathbb{N}$ induces the *l-similarity relation* $\sim_{L,l}$ defined as follows: for $x, y \in \Sigma^\star$, if $|x| \leq l$ and $|y| \leq l$, then

$$x \sim_{L,l} y \text{ iff for all } w \in \Sigma^{\leq l-\max\{|x|,|y|\}},\ xw \in L \Leftrightarrow yw \in L. \tag{5}$$

Two words $x, y \in \Sigma^\star$ are *l-dissimilar*, and which we denote by $x \not\sim_{L,l} y$, if $|x|, |y| \leq l$ and $x \sim_{L,l} y$ does not hold. It is clear that the l-similarity relation $\sim_{L,l}$ generalizes the Myhill-Nerode equivalence relation. The relation $\sim_{L,l}$ is reflexive, symmetric, but not transitive. However, $\sim_{L,l}$ is semi-transitive, i.e., if $|x| \leq |y| \leq |z|$ if $x \sim_{L,l} y$ and $y \sim_{L,l} z$ (resp. $x \sim_{L,l} z$), then $x \sim_{L,l} z$ (resp. $y \sim_{L,l} z$).

The maximum number of dissimilar words with respect to the relation $\sim_{L,l}$ is denoted by $N_L(l)$, [10]. In case of a finite language, if l is the length of the longest word in L, l-similarity is called just the *similarity* induced by L and we omit the subscript l. For an arbitrary language L, l-similarity relation is the similarity relation induced by the finite language $L \cap \Sigma^{\leq l}$.

If two words are dissimilar with respect to the similarity relation induced by a finite language L with l as the length of the longest word in L, or by a language L and a constant $l \in \mathbb{N}$, then we can find at least one $w \in \Sigma^{\leq l-\max\{|x|,|y|\}}$, such that $xw \in L$ and $yw \notin L$, or $xw \notin L$ and $yw \in L$. Given $x, y \in \Sigma^\star$, with $|x|, |y| \leq l$, we define the *dissimilarity language* of x and y w.r.t $L \cap \Sigma^{\leq l}$ by

$$\widetilde{\mathsf{D}}_{L,l}(x,y) = \{w \in \Sigma^{\leq l-\max\{|x|,|y|\}} \mid xw \in L \not\Leftrightarrow yw \in L\}. \tag{6}$$

It is clear that we have $\widetilde{\mathsf{D}}_{L,l}(x,y) \neq \emptyset$ iff $x \not\sim_{L,l} y$.

Naturally, we define the *dissimilarity language* of $L \cap \Sigma^{\leq l}$ by

$$\widetilde{\mathsf{D}}(L,l) = \{w \in \Sigma^\star \mid \exists x, y \in \Sigma^\star (xw \in L \wedge yw \notin L \wedge w \in \Sigma^{\leq l-max\{|x|,|y|\}})\}. \tag{7}$$

It is immediate that $\widetilde{\mathsf{D}}(L,l) = \bigcup_{x,y\in\Sigma^\star} \widetilde{\mathsf{D}}_{L,l}(x,y)$. If L is finite, we omit the argument l. Thus, in general, we have

$\widetilde{\mathsf{D}}(L,l) = \widetilde{\mathsf{D}}(L\cap\Sigma^{\leq l})$. Because $\widetilde{\mathsf{D}}_{L,l}(x,y) \subseteq \mathsf{D}_L(x,y)$, it follows that $\widetilde{\mathsf{D}}(L,l) \subseteq \mathsf{D}(L)$. If $x,y\in\Sigma^\star$ w.r.t. $L\cap\Sigma^{\leq l}$ we define

$$\underline{\widetilde{\mathsf{D}}}_{L,l}(x,y) = \min\{w\in\Sigma^{\leq l-\max\{|x|,|y|\}} \mid w\in \widetilde{\mathsf{D}}_{L,l}(x,y)\}, \tag{8}$$

where the minimum is taken according to the quasi-lexicographical order[1]. In case $x\sim_{L,l} y$, $\underline{\widetilde{\mathsf{D}}}_{L,l}(x,y)$ is undefined[2]. The set of minimal words that distinguishes dissimilar words w.r.t. L,l is

$$\underline{\widetilde{\mathsf{D}}}\,(L,l) = \bigcup_{\substack{x,y\in\Sigma^*\\ x\not\sim_{L,l} y}} \underline{\widetilde{\mathsf{D}}}_{L,l}(x,y). \tag{9}$$

Given a DFA $\mathcal{A}=\langle Q,\Sigma,q_0,\delta,F\rangle$ and $l\in\mathbb{N}$, let $\sim_{\mathcal{A},l}$ be the relation on Q defined by $p\sim_{\mathcal{A},l} q$ if for every word $w\in\Sigma^{\leq l-\max\{\mathrm{level}(p),\mathrm{level}(q)\}}$, $\delta(p,w)\in F \Leftrightarrow \delta(q,w)\in F$. Then, for the automaton $\mathcal{A}=\langle Q,\Sigma,q_0,\delta,F\rangle$ and $p,q\in Q$, we define

$$\widetilde{\mathsf{D}}_{\mathcal{A},l}(p,q) = \{w\in\Sigma^{\leq l-\max\{\mathrm{level}(p),\mathrm{level}(q)\}} \mid \delta(p,w)\in F \not\Leftrightarrow \delta(q,w)\in F\},\ \text{and}$$
$$\widetilde{\mathsf{D}}(\mathcal{A},l) = \{w \mid \exists p,q\in Q\ (\delta(p,w)\in F\wedge\delta(q,w)\notin F\wedge w\in\Sigma^{\leq l-\max\{\mathrm{level}(p),\mathrm{level}(q)\}})\}.$$

Finally, we can also define the set of minimal words that distinguish $p,q\in Q$ with $p\not\sim_{\mathcal{A},l} q$:

$$\underline{\widetilde{\mathsf{D}}}_{\mathcal{A},l}(p,q) = \min\{w\in\Sigma^{\leq l-\max\{\mathrm{level}(p),\mathrm{level}(q)\}} \mid w\in \widetilde{\mathsf{D}}_{\mathcal{A},l}(p,q)\},\ \text{and}$$
$$\underline{\widetilde{\mathsf{D}}}\,(\mathcal{A},l) = \bigcup_{\substack{p,q\in Q\\ p\not\sim_{\mathcal{A},l} q}} \underline{\widetilde{\mathsf{D}}}_{\mathcal{A},l}(p,q).$$

A simple calculation shows that $\widetilde{\mathsf{D}}_{\mathcal{A},l}(p,q) = \widetilde{\mathsf{D}}_{\mathcal{L}(\mathcal{A}),l}(x_{\mathcal{A}}(p),x_{\mathcal{A}}(q))$ and $\underline{\widetilde{\mathsf{D}}}_{\mathcal{A},l}(p,q) = \underline{\widetilde{\mathsf{D}}}_{\mathcal{L}(\mathcal{A}),l}(x_{\mathcal{A}}(p),x_{\mathcal{A}}(q))$.

Lemma 3.1 *For every finite language L and a DFCA $\mathcal{A}$ for L, we have that $\widetilde{\mathsf{D}}(\mathcal{A}) = \widetilde{\mathsf{D}}(L)$, and $\underline{\widetilde{\mathsf{D}}}\,(\mathcal{A}) = \underline{\widetilde{\mathsf{D}}}\,(L)$.*

Proof. We considerer only the first equality.

$$\begin{aligned}
\widetilde{\mathsf{D}}(\mathcal{A}) &= \bigcup_{\substack{p,q\in Q\\ p\not\sim_{\mathcal{A}} q}} \widetilde{\mathsf{D}}_{\mathcal{A}}(p,q)\\
&= \bigcup_{\substack{p,q\in Q\\ p\not\sim_{\mathcal{A}} q}} \left\{w\in\Sigma^{\leq l-\max\{\mathrm{level}(p),\mathrm{level}(q)\}} \mid \delta(p,w)\in F \not\Leftrightarrow \delta(q,w)\in F\right\}\\
&= \bigcup_{\substack{p,q\in Q\\ p\not\sim_{\mathcal{A}} q}} \left\{w\in\Sigma^{\leq l-\max\{|x_{\mathcal{A}}(p)|,|x_{\mathcal{A}}(q)|\}} \mid x_{\mathcal{A}}(p)w\in L \not\Leftrightarrow x_{\mathcal{A}}(q)w\in L\right\}
\end{aligned}$$

[1] $\underline{\widetilde{\mathsf{D}}}_{L,l}(x,y)$ is a singleton set or empty, because quasi-lexicographical order is a total order.

[2] This is the case when $\underline{\widetilde{\mathsf{D}}}_{L,l}(x,y)=\emptyset$.

$$
\begin{aligned}
&= \bigcup_{\substack{x,y\in\Sigma^\star \\ x\not\sim_L y}} \left\{w \in \Sigma^{\leq l-\max\{|x|,|y|\}} \mid xw \in L \nLeftrightarrow yw \in L\right\} \\
&= \bigcup_{\substack{x,y\in\Sigma^\star \\ x\not\sim_L y}} \widetilde{\mathsf{D}}_L(x,y) \\
&= \widetilde{\mathsf{D}}(L).
\end{aligned}
$$

□

Because the dissimilarity condition is stronger than distinguishability, if L is finite then we have that $\widetilde{\mathsf{D}}(L) \subseteq \mathsf{D}(L)$. Even more, because L is a finite language included in $\Sigma^{\leq l}$, $\widetilde{\mathsf{D}}(L)$ is finite and included in $\Sigma^{\leq l-1}$.

It is clear that the distinguishability operation and dissimilarity operation can differ on finite languages and it is worth studying the new operation. Moreover, the minimal version of the dissimilarity operation is different than the minimal version of the distinguishability operation. In Figures 1 and 2 we can see examples of languages where distinguishability and dissimilarity languages are different.

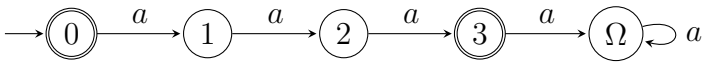

Figure 1: For the language $L = \{\varepsilon, a^3\} \subseteq \{a\}^\star$, we have that $a^3 \in \mathsf{D}(L) \setminus \widetilde{\mathsf{D}}(L)$ and $a^3 \in \underline{\mathsf{D}}(L) \setminus \widetilde{\underline{\mathsf{D}}}\,(L)$, since $\mathsf{D}(L) = \{\varepsilon, a, a^2, a^3\}$, $\widetilde{\mathsf{D}}(L) = \{\varepsilon, a, a^2\}$, $\underline{\mathsf{D}}(L) = \mathsf{D}(L)$, and $\widetilde{\underline{\mathsf{D}}}\,(L) = \{\varepsilon, a\}$.

We observe that if a word w distinguishes between two words x, y w.r.t. a finite language L, and $|x| < |y|$, we must have $w \in \Sigma^{\leq l-|y|} \subseteq \Sigma^{\leq l-1}$, i.e., $\widetilde{\mathsf{D}}(L) \subseteq \mathsf{D}(L) \cap \Sigma^{\leq l-1}$. This is the case in the examples of Figures 1 and 2, where even the equality holds. However, in general the equality is not true, i.e., $\mathsf{D}(L) \cap \Sigma^{\leq l-1} \neq \widetilde{\mathsf{D}}(L)$, because two equivalent words must be similar, but we may have similar words that are not equivalent. For example, for the language of Figure 3, we have that $\mathsf{D}(L) = L$, $l = 3$, and $\widetilde{\mathsf{D}}(L) \subsetneq \mathsf{D}(L) \cap \Sigma^{\leq 2}$. Finally, in Figure 4 we present a language L for which all the operations D, $\underline{\mathsf{D}}$, $\widetilde{\mathsf{D}}$, and $\widetilde{\underline{\mathsf{D}}}$ are distinct.

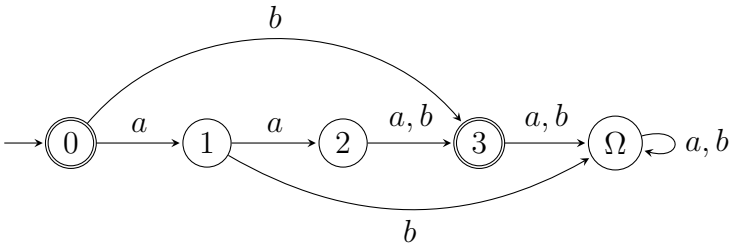

Figure 2: For the language $L = \{\varepsilon, b, a^3, aab\}$, we have that $\mathsf{D}(L) = \{\varepsilon, a, b, a^2, ab, a^3, a^2b\}$, $\underline{\mathsf{D}}(L) = \{\varepsilon, a, b, a^2\}$, $\widetilde{\mathsf{D}}(L) = \{\varepsilon, a, b, a^2, ab\}$, and $\widetilde{\underline{\mathsf{D}}}\,(L) = \{\varepsilon, a, b\}$.

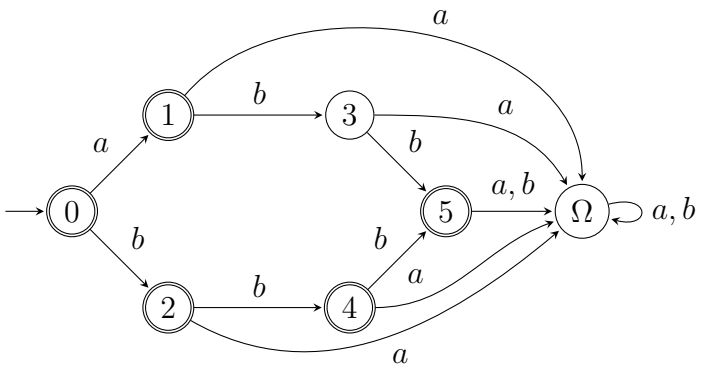

Figure 3: For the language $L = \{\varepsilon, a, b, b^2, ab^2, b^3\}$, we have that $\mathsf{D}(L) = L$, $\underline{\mathsf{D}}(L) = \{\varepsilon, a, b, b^2\}$, $\widetilde{\mathsf{D}}(L) = \{\varepsilon, a, b\}$, and $\underline{\widetilde{\mathsf{D}}}\,(L) = \widetilde{\mathsf{D}}(L)$.

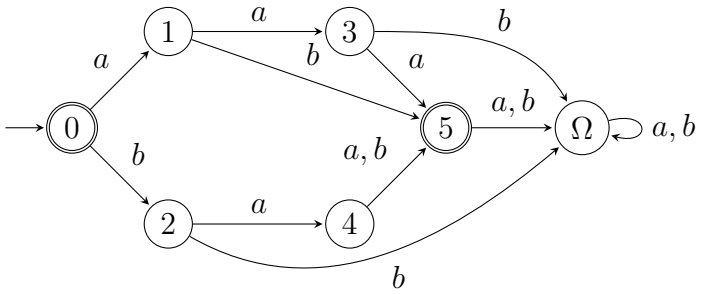

Figure 4: For the language $L = \{\varepsilon, b, ab, aaa, baa, bab\}$, we have that $\mathsf{D}(L) = \{\varepsilon, a, b, aa, ab, aaa, baa, bab\}$, $\underline{\mathsf{D}}(L) = \{\varepsilon, a, b, aa\}$, $\widetilde{\mathsf{D}}(L) = \{\varepsilon, a, b, aa, ab\}$, and $\underline{\widetilde{\mathsf{D}}}\,(L) = \{\varepsilon, a, b\}$.

4. Some Properties of $\widetilde{\mathsf{D}}$ and $\underline{\widetilde{\mathsf{D}}}$

In this section we give several characterizations of the dissimilarity languages.

Theorem 4.1 *If L is a regular language and $l \in \mathbb{N}$, then the language $\widetilde{\mathsf{D}}(L, l)$ is suffix closed, i.e.,*

$$(\forall w \in \widetilde{\mathsf{D}}(L,l))(\forall x, y \in \Sigma^{\leq l-1})(w = xy \implies y \in \widetilde{\mathsf{D}}(L,l)).$$

Proof. Let $w \in \widetilde{\mathsf{D}}(L, l)$, i.e., there exist $x, y \in \Sigma^{\leq l}$ such that $w \in \Sigma^{\leq l - \max\{|x|,|y|\}}$ and $xw \in L$ and $yw \notin L$. If v is a suffix of w, i.e., $w = uv$, for an $u \in \Sigma^\star$, $v \in \Sigma^{\leq l - \max\{|xu|,|yu|\}}$, then we can write $xuv \in L$ and $yuv \notin L$, which means that $v \in \widetilde{\mathsf{D}}(L, l)$. □

Theorem 4.2 *If L is a regular language and $l \in \mathbb{N}$, then $\underline{\widetilde{\mathsf{D}}}\,(L, l)$ is suffix closed.*

Proof. Let $w \in \underline{\widetilde{\mathsf{D}}}\,(L, l)$, and let $w = uv$, with $u, v \in \Sigma^\star$. Because $w \in \underline{\widetilde{\mathsf{D}}}\,(L, l)$, we can find two other words, $x, y \in \Sigma^\star$, such that $xw \in L$ and $yw \notin L$, i.e., $xuv \in L$ and $yuv \notin L$. It follows that $v \in \widetilde{\mathsf{D}}_{L,l}(xu, yu)$. Since $v \in \widetilde{\mathsf{D}}_{L,l}(xu, yu)$, there exists $v' = \underline{\widetilde{\mathsf{D}}}_L(xu, yu)$ and $v' \preceq_l v$.

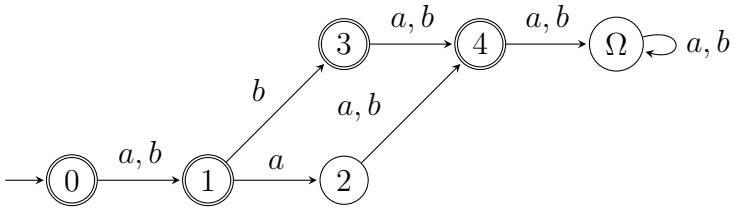

Figure 5: For the language $L = \{\varepsilon, a, b, ab, bb, aba, abb, bba, bbb, aaa, baa, aab, bab\}$, $l = 3$, we have a strict inclusion for $\widetilde{\mathsf{D}}(L) \subsetneq \mathsf{D}(L) \cap \Sigma^{\leq l-1}$, with $\mathsf{D}(L) = \Sigma^{\leq 3}$, $\widetilde{\mathsf{D}}(L) = \{\varepsilon, a, aa, ba\}$, $\underline{\mathsf{D}}(L) = \{\varepsilon, a, b, ab\}$ and $\underline{\widetilde{\mathsf{D}}}\,(L) = \{\varepsilon, a\}$.

Hence, $uv' \preceq_l uv$ and $uv' \in \underline{\widetilde{\mathsf{D}}}_L(x, y)$, which implies that $w = uv \preceq_l uv'$. Then we must have $uv' = uv$, which implies that $v = v' = \underline{\widetilde{\mathsf{D}}}_L(xu, yu) \in \underline{\widetilde{\mathsf{D}}}\,(L)$. □

For a language L and $l \in \mathbb{N}$, let $\mathsf{suff}_l(L) = \mathsf{suff}(L) \cap \Sigma^{\leq l}$. Because $\underline{\widetilde{\mathsf{D}}}\,(L, l)$ and $\widetilde{\mathsf{D}}(L, l)$ are suffix closed and $\underline{\widetilde{\mathsf{D}}}\,(L, l) \subseteq \widetilde{\mathsf{D}}(L, l) \subseteq \Sigma^{\leq l-1}$, it follows that

$$\mathsf{suff}_{l-1}(\underline{\widetilde{\mathsf{D}}}\,(L, l)) = \underline{\widetilde{\mathsf{D}}}\,(L, l) \text{ and } \mathsf{suff}_{l-1}(\widetilde{\mathsf{D}}(L, l)) = \widetilde{\mathsf{D}}(L, l).$$

Considering that $\mathsf{D}(L) = \mathsf{suff}(L) \cap \mathsf{suff}(\overline{L})$ (c.f. [5]), we can formalize the observations made in the previous section.

Theorem 4.3

1. $\widetilde{\mathsf{D}}(L, l) \subseteq \mathsf{suff}_{l-1}(L) \cap \mathsf{suff}_{l-1}(\overline{L})$.
2. *There is a finite language* L *such that* $\widetilde{\mathsf{D}}(L) \neq \mathsf{suff}_{l-1}(L) \cap \mathsf{suff}_{l-1}(\overline{L})$.

Proof. For the first inclusion is enough to see that if $w \in \widetilde{\mathsf{D}}(L, l)$, then $xw \in L$ and $yw \notin L$ for some $x, y \in \Sigma^{\leq l}$, thus $w \in \mathsf{suff}(\overline{L})$. We already know that $w \in \mathsf{suff}(L)$ and $w \in \Sigma^{\leq l-1}$, therefore it follows that $w \in \mathsf{suff}_{l-1}(L) \cap \mathsf{suff}_{l-1}(\overline{L})$. For the second part, we consider the finite language accepted by the automaton in Figure 5. For example, $bb \in \mathsf{suff}_{l-1}(L) \cap \mathsf{suff}_{l-1}(\overline{L}) = \mathsf{D}(L) \cap \Sigma^{\leq l-1} = \{\varepsilon, a, b, aa, ab, ba, bb\}$. The word bb could only distinguish between states on level 0 or level 1, i.e., between state 0 and state 1. However, $\delta(0, bb) = 3 \in F$ and $\delta(1, bb) = 4 \in F$, therefore $bb \notin \widetilde{\mathsf{D}}(L)$. □

After applying the dissimilarity operation to a finite language having the longest word at most equal to l, we obtain another language with the length of the longest word at most $l - 1$, which obviously contrasts to the case of distinguishability operation, where we could even have a fixed point after only two iterations. In the case of dissimilarity nonempty fixed points obviously do not exist, but it is clear that after at most $l - 1$ iterations we will get only one state, thus the dissimilarity language is $\emptyset$, and this is the only possible fixed point for $\widetilde{\mathsf{D}}$.

Given L and for any $l \in \mathbb{N}$ we have $\widetilde{\mathsf{D}}(L \cap \Sigma^{\leq l}) \subseteq \mathsf{D}(L) \cap \Sigma^{\leq l-1} \subseteq \mathsf{D}(L)$. If $w \in \mathsf{D}(L)$, then we have at least two words x, y such that $xw \in L$ and $yw \notin L$. Taking $l = \max\{|xw|, |yw|\} + 2$ we discover that $w \in \widetilde{\mathsf{D}}(L \cap \Sigma^{\leq l})$. Thus, for the sequence $\widetilde{\mathsf{D}}(L \cap \Sigma^{\leq l})$ this simple computation shows that $\cup_{l \in \mathbb{N}}(\widetilde{\mathsf{D}}(L \cap \Sigma^{\leq l})) = \mathsf{D}(L)$. Thus if L is regular, then $\cup_{l \in \mathbb{N}}(\widetilde{\mathsf{D}}(L \cap \Sigma^{\leq l}))$ is also a regular language [5].

However, if $\mathsf{D}(L)$ is regular we cannot say the same thing about L.

Lemma 4.4 *There is a non-regular language L such that $\mathsf{D}(L)$ is regular.*

Proof. If we consider the language $L = \{a^{n^2} \mid n \geq 0\}$, we have that $\mathsf{D}(L) = \{a^n \mid n \geq 0\}$, which is regular. □

The next result gives an upper-bound for the number of elements of minimal dissimilar words, $\widetilde{\underline{\mathsf{D}}}(L)$.

Theorem 4.5 *If L is a regular language with state complexity $n \geq 2$, then $|\widetilde{\underline{\mathsf{D}}}(L, l)| \leq n - 1$.*

Proof. We can use the exact same argument as in the case of $\underline{\mathsf{D}}$ operation on regular languages: for any three sets A, B and C we have the equality $(A\Delta B)\Delta(B\Delta C) = A\Delta C$. Therefore, we can distinguish any pair from n distinct sets with at most $n - 1$ elements. □

Now, we prove that the upper-bound is reached.

Theorem 4.6 *The bound $n - 1$ for the size of $\widetilde{\underline{\mathsf{D}}}(L)$, for a regular language L with state complexity $n \geq 2$, is tight.*

Proof. Consider the language $S_m = \{a^m b\}$, $m \geq 0$, therefore $sc(S_m) = m + 3$. We know [5] that $\mathsf{D}(S_m) = \mathsf{suff}(\{a^m b\})$, therefore we have $\widetilde{\mathsf{D}}(S_m) = \mathsf{D}(S_m) \cap \Sigma^{\leq m+1-1} = \{a^i b \mid 0 \leq i < m\}$. Because $\widetilde{\underline{\mathsf{D}}}(S_m) = \widetilde{\mathsf{D}}(S_m)$ and $|\widetilde{\mathsf{D}}(S_m)| = m = sc(S_m) - 2$, it follows that the upper-bound is reached. □

5. State Complexity of Distinguishability and Dissimilarity on Finite Languages

The state complexity of the D operation for regular languages with state complexity n is $2^n - n$ [5]. We also know that for a finite language L, $\mathsf{D}(L) = \mathsf{suff}(L)$. If $L = \{w\}$, for $w \in \Sigma^\star$ it is known [16] that $sc(\mathsf{suff}(L)) \leq 2|w|$. Now let us consider the family of languages $V_n = \{v_n\}$, where $v_n = ab^n$, $n \geq 1$. It is easy to check, for $v = v_n$ that, $|v| = n+1$, $sc(V_n) = n+3 = |v|+2$, and $sc(\mathsf{suff}(V_n)) = 2n + 2 = 2|v|$. Hence, we have $sc(\mathsf{D}(V_n)) = 2n + 2 = 2|v| = 2sc(V_n) - 4$. Therefore, we have just proved the following lemma.

Lemma 5.1 *The state complexity of D operation on singleton languages is equal to $2n-4$ and the upper bound is reached.*

Next proposition relates the state complexity, cover state complexity, distinguishability and dissimilarity operations.

Proposition 5.2 *There exists a family of finite languages X_n over a binary alphabet such that the length of the longest word in X_n is $l = 2n + 1$, and the following statements hold true*

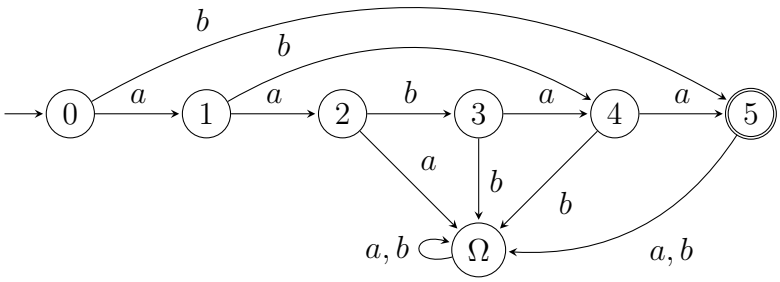

Figure 6: The automaton for the language X_n, where $n = 2$.

1. $sc(X_n) = csc(X_n) = l + 2$;
2. $\widetilde{\mathsf{D}}(X_n) = \mathsf{D}(X_n) \cap \Sigma^{\leq l-1}$;
3. $sc(\mathsf{D}(X_n)) = \frac{(n+1)(n+4)}{2} + 1 = sc(\widetilde{\mathsf{D}}(X_n)) + n + 1$;
4. $\underline{\mathsf{D}}(X_n) = \underline{\widetilde{\mathsf{D}}}\,(X_n)$;
5. $|\underline{\mathsf{D}}(X_n)| = l + 1 = sc(X_n) - 1$ *and* $sc(\underline{\mathsf{D}}(X_n)) = n + 3$.

Proof. Let us consider the language $X_n = \{a^i b a^i \mid 0 \leq i \leq n\}$, $n \geq 0$. For $n = 2$ the automaton is depicted in Figure 6. We observe that the minimal DFA is also a minimal DFCA, as no two states are similar. For the language X_n, we have $sc(X_n) = csc(X_n) = 3 + 2n = l + 2$, thus *1.* holds.

We have

$$\mathsf{D}(X_n) = \left\{a^i \mid 0 \leq i \leq n\right\} \cup \left\{ba^i \mid 0 \leq i \leq n\right\} \cup \left\{a^i b a^j \mid 0 \leq i \leq n, i \leq j \leq n\right\},$$

and

$$\begin{aligned}\widetilde{\mathsf{D}}(X_n) &= \left\{a^i \mid 0 \leq i \leq n\right\} \cup \left\{ba^i \mid 0 \leq i \leq n\right\} \cup \left\{a^i b a^j \mid 0 \leq i \leq n-1, i \leq j \leq n\right\} \\ &= \mathsf{D}(X_n) \cap \Sigma^{\leq l-1},\end{aligned}$$

therefore *2.* is true. For the state complexity of $\mathsf{D}(X_n)$ we observe the following, where $\equiv$ is the Myhill-Nerode relation induced by $\mathsf{D}(X_n)$:

1. for $0 \leq j \leq n$, $0 \leq i < j$, $a^i b a^j \equiv b a^j$;
2. for $0 \leq i < j \leq n$, $a^i b a^i \in \mathsf{D}(X_n)$ and $a^j b a^i \notin \mathsf{D}(X_n)$, thus $a^i \not\equiv a^j$;
3. for $0 \leq i < j \leq n$, $a^i b a^i \in \mathsf{D}(X_n)$ and $b a^j b a^i \notin \mathsf{D}(X_n)$, thus $a^i \not\equiv b a^j$;
4. for $0 \leq i < j \leq n$, $b a^i b a^j \notin \mathsf{D}(X_n)$ and $a^j b a^j \in \mathsf{D}(X_n)$, thus $b a^i \not\equiv a^j$;
5. for $0 \leq i \leq n$, $b a^i b a^i \notin \mathsf{D}(X_n)$ and $a^i b a^i \in \mathsf{D}(X_n)$, thus $a^i \not\equiv b a^i$;
6. for $0 \leq m, i < j < k \leq n$, $a^m b a^j a^{n-j} \in \mathsf{D}(X_n)$ and $a^i b a^k a^{n-j} \notin \mathsf{D}(X_n)$, thus $a^m b a^j \not\equiv a^i b a^k$.

Hence,

$$sc(\mathsf{D}(X_n)) = (n+1) + \frac{(n+1)(n+2)}{2} + 1 = \frac{(n+1)(n+4)}{2} + 1.$$

For $\widetilde{\mathsf{D}}(X_n)$, we loose the word a^nba^n (the longest word), thus

$$sc(\widetilde{\mathsf{D}}(X_n)) = \frac{(n+1)(n+4)}{2} + 1 - (n+1) = \frac{(n+1)(n+4)}{2} - n.$$

Hence, *3.* holds. For the minimal words, we have

$$\underline{\widetilde{\mathsf{D}}}\,(X_n) = \left\{a^i \mid 0 \leq i \leq n\right\} \cup \left\{ba^i \mid 0 \leq i \leq n\right\},$$

and

$$\underline{\mathsf{D}}(X_n) = \left\{a^i \mid 0 \leq i \leq n\right\} \cup \left\{ba^i \mid 0 \leq i \leq n\right\},$$

thus *4.* holds. We also have $|\underline{\mathsf{D}}(X_n)| = 2n+2 = l+1$ and $sc(\underline{\mathsf{D}}(X_n)) = n+3$, hence *5.* holds.

□

To obtain the state complexity of the $\widetilde{\mathsf{D}}$ operation, we first analyze in more detail the D operation for finite languages, i.e., $\mathsf{D}(L) = \mathsf{suff}(L)$. For regular languages L with $sc(L) = n$ and $\emptyset$ as one of the quotients, i.e., the minimal DFA has a dead state, the state complexity of suffix closure is 2^{n-1} [2]. However, this bound is too large for finite languages and it can be improved. Let $\mathcal{A} = \langle\{0,\ldots,n-1\}, \Sigma, 0, \delta, F\rangle$ be the minimal DFA for a finite language L, and let the states of $\mathcal{A}$ be topologically ordered. In $\mathcal{A}$, there exists a *last* final state, or *pre-dead* state, for which all transitions go to the dead state. We assume that the dead state is $n-1$ and the last final state is $n-2$.

The standard construction for obtaining a DFA for $\mathsf{suff}(L)$ is to mark as initial all useful states of $\mathcal{A}$, and apply the subset construction. Because $\mathcal{A}$ is acyclic and topologically ordered, at level 0 we have state $\{0,\ldots,n-1\}$, at level 1 we have states that are subsets of $\{1,\ldots,n-1\}$, and with every increment of the level we loose at least one state. Because we need to reach all the subsets, and at each level i we can get at most k^i states, where $k = |\Sigma|$, it follows that the state complexity of D operation is bounded by

$$sc(\mathsf{D}(L)) \leq \sum_{i=0}^{n-2} \min\left\{2^{n-1-i}, k^i\right\} = \sum_{i=0}^{r-1} k^i + 2^{n-1-r},$$

where r is minimal with the property that $2^{n-1-r} \leq k^r$. This bound is exactly the same as the one for the reversal operation on finite languages, which is known to be $O(k^{\frac{n}{1+\log k}})$ [4, 17]. For the binary case, i.e., for $k = 2$, the above upper-bound becomes: if $sc(L) = n = 2t$, then $r = t$ and $sc(\mathsf{D}(L)) = 2^t + 2^{t-1} - 1$; and if $sc(L) = n = 2t-1$, then $r = t-1$ and $sc(\mathsf{D}(L)) = 2^t - 1$.

We now consider the language $M_{m,h} = \{wav \mid |w| = m, |v| = h\} = \Sigma^m a \Sigma^h$, for $0 \leq h \leq m$, over the alphabet $\Sigma = \{a, b\}$. The language $M_{m,h}$ has the following properties:

1. $sc(M_{m,h}) = (m+1) + (h+1) + 1 = m + h + 3$;
2. $\mathsf{D}(M_{m,h}) = \Sigma^{\leq h} \cup a\Sigma^h \cup \Sigma^{\leq m} a \Sigma^h = \Sigma^{\leq h} \cup \Sigma^{\leq m} a \Sigma^h$;
3. For the state complexity of $\mathsf{D}(M_{m,h})$ we consider two distinct words $w_1, w_2 \in \Sigma^{\leq m+1}$. If $|w_1| < |w_2|$, w_1 is not $\equiv_{\mathsf{D}(M_{m,h})}$ equivalent to w_2, because $b^{m-|w_1|}ab^h$ will distinguish them.

Note that if $|w_2| \leq m$, it is clear and if $|w_2| = m+1$ one has $|w_1| \leq m$. In both cases, $w_1 b^{m-|w_1|} ab^h \in \mathsf{D}(M_{m,h})$ and $w_2 b^{m-|w_1|} ab^h \notin \mathsf{D}(M_{m,h})$.
In case $|w_1| = |w_2|$, let $1 \leq i \leq h+1$ be the first position from the right, where w_1 and w_2 are different. Then b^{h+1-i} will distinguish w_1 from w_2: if w_1 has an a at position i (from the right), then $w_1 b^{h+1-i} \in \mathsf{D}(M_{m,h})$ and $w_2 b^{h+1-i} \notin \mathsf{D}(M_{m,h})$, or vice-versa. Therefore, all words up to length $h+1$ are not equivalent; there are $1+2+\cdots+2^{h+1}$ such words. For each length between $h+2$ and $m+1$, we have at least 2^{h+1} different equivalence classes. Hence, there are at least

$$1+2+\cdots+2^{h+1}+\underbrace{2^{h+1}+\cdots+2^{h+1}}_{m-h \text{ terms}} = 2^{h+2}-1+(m-h)2^{h+1} \tag{10}$$

equivalence classes. In case of $m = h$,

$$sc(\mathsf{D}(M_{h,h})) = 2^{h+2}-1 = 2^t-1,$$

where $sc(M_{h,h}) = 2h+3 = 2t-1$. In case of $m = h+1$,

$$sc(\mathsf{D}(M_{h+1,h})) = 2^{h+2}-1+2\cdot 2^h = 2^t+2^{t-1}-1,$$

where $sc(M_{h+1,h}) = h+1+h+3 = 2h+4 = 2t$.

We have just proved the following theorem:

Theorem 5.3 *For a finite language L with state complexity n, the state complexity of $\mathsf{D}(L)$ is $O(k^{\frac{n}{1+log k}})$, where $|\Sigma| = k$. In case L is a finite language over a binary alphabet, then*

$$sc(\mathsf{D}(L)) = 2^t + 2^{t-1} - 1,$$

if $n = 2t$, and

$$sc(\mathsf{D}(L)) = 2^t - 1,$$

if $n = 2t-1$. The upper-bound of the state complexity for D operation is reached.

For the state complexity of $\widetilde{\mathsf{D}}$, we observe that $\widetilde{\mathsf{D}}(L) \subseteq \mathsf{D}(L) \cap \Sigma^{\leq l-1} = \mathsf{suff}_{l-1}(L)$.

Thus, the state complexity of $\widetilde{\mathsf{D}}(L)$ must be smaller than the state complexity of $\mathsf{D}(L)$, as some states must be merged.

For a lower bound of the state complexity of the $\widetilde{\mathsf{D}}$ operation we consider again the language $M_{m,h}$ with $l = m+h+1$. It is clear that $\widetilde{\mathsf{D}}(M_{m,h}) = \mathsf{D}(M_{m,h}) \cap \Sigma^{\leq l-1} =$, i.e.,

$$\widetilde{\mathsf{D}}(M_{m,h}) = \Sigma^{\leq h} \cup \Sigma^{\leq m-1} a \Sigma^h.$$

Now we count the number of distinguishable words with respect to the language $\widetilde{\mathsf{D}}(M_{m,h})$. Using similar arguments as we used for D operation, we have that $sc(\widetilde{\mathsf{D}}(M_{h,h})) = 2^{h+1}+2^{h-1}-1$, and in case $m = h+1$, we have $sc(\widetilde{\mathsf{D}}(M_{m,h})) = 2^{h+2}-1$. The second case coincides with the case $m = h$ for $\mathsf{D}(M_{h,h})$. For the first case, let $w_1, w_2 \in \Sigma^{\leq h+1}$. We will denote $z \in \Sigma^\star$ a *witness* of non equivalence under $\equiv_{\widetilde{\mathsf{D}}(M_{h,h})}$ if $w_1 z \in \widetilde{\mathsf{D}}(M_{h,h})$ and $w_2 z \notin \widetilde{\mathsf{D}}(M_{h,h})$.

If $|w_1| \leq |w_2|$ we have the following cases.

1. if $|w_1| \leq h-1$ then $w_1 \not\equiv_{\widetilde{\mathsf{D}}(M_{h,h})} w_2$ with witness $z = b^{h-1-|w_1|}ab^h$;
2. if $|w_1| = h$ the following subcases need to be considered. Let $y, y_1 \in \Sigma^{h-1}$ and $x \in \{a, b\}$.
   (a) if $w_2 = ayx$ and $w_1 = yb$ then $w_1 \equiv_{\widetilde{\mathsf{D}}(M_{h,h})} w_2$. First note that $w_1, w_2 \in \widetilde{\mathsf{D}}(M_{h,h})$. For $z \in \Sigma^+$, $w_1 z \in \widetilde{\mathsf{D}}(M_{h,h})$ if and only if $y = x_1 a x_2$ with $|x_1|, |x_2| < h-1$, but then also $w_2 z \in \widetilde{\mathsf{D}}(M_{h,h})$. If $y = b^{h-1}$, then $w_1 z, w_2 z \notin \widetilde{\mathsf{D}}(M_{h,h})$, for $z \in \Sigma^+$.
   (b) if $w_2 = ayx$ and $w_1 = ya$ then $w_1 \not\equiv_{\widetilde{\mathsf{D}}(M_{h,h})} w_2$. For $z = b^h$, $w_1 z \in \widetilde{\mathsf{D}}(M_{h,h})$ and $w_2 z \notin \widetilde{\mathsf{D}}(M_{h,h})$.
   (c) if $w_2 = ayx$ and $w_1 = y_1 b$ then $w_1 \not\equiv_{\widetilde{\mathsf{D}}(M_{h,h})} w_2$. Let i be the first position from the right such that y_1 and y disagree. Then either $w_1 b^{h-i+1} \in \widetilde{\mathsf{D}}(M_{h,h})$ or $w_2 b^{h-i+1} \in \widetilde{\mathsf{D}}(M_{h,h})$, but not both.
   (d) if $w_2 = by$, then $w_1 \not\equiv_{\widetilde{\mathsf{D}}(M_{h,h})} w_2$ with witness $z = \varepsilon$.

If $|w_1| = |w_2|$ we have:

1. $|w_1| = |w_2| \leq h$. Let i, with $1 \leq i \leq h$, be the first position (from the left) such that w_1 and w_2 disagree. Then, $w_1 \not\equiv_{\widetilde{\mathsf{D}}(M_{h,h})} w_2$ with witness $z = b^i$.
2. $|w_1| = |w_2| = h+1$ and there exists a position i, with $1 \leq i \leq h$, such that i is the first position (from the left) such that w_1 and w_2 disagree. Then, $w_1 \not\equiv_{\widetilde{\mathsf{D}}(M_{h,h})} w_2$ with witness $z = b^{i-1}$.
3. $|w_1| = |w_2| = h+1$, $w_1 = ya$ and $w_2 = yb$. Then $w_1 \equiv_{\widetilde{\mathsf{D}}(M_{h,h})} w_2$ because for $z \in \Sigma^\star$ $w_1 z$, $w_2 z \in \widetilde{\mathsf{D}}(M_{h,h})$ if and only if there exists an a in y at a position $i \leq h-1$ and $|z| = i-1$.

We can conclude that all words up to length h, i.e., $2^{h+1}-1$ words, are not equivalent. There are 2^h classes of words of length exactly $h+1$ and 2^{h-1} classes of words of length $h+1$ that coincide with classes of words of length 2^h (case 2a). Finally, let us consider a word $w \in \Sigma^{h+1+n}$, with $1 \leq n \leq h-1$. Let $w = xys$ with $|x| = n$, $|y| = h-n$ and $|s| = n+1$, then $w \equiv_{\widetilde{\mathsf{D}}(M_{h,h})} yb^{n+1}$. We have that $wz \in \widetilde{\mathsf{D}}(M_{h,h})$ if and only if $y = x_1 a x_2$ and $|z| = h-n-1-|x_2|$, thus if and only if $yb^{n+1}z \in \widetilde{\mathsf{D}}(M_{h,h})$. It follows that the next theorem holds:

Theorem 5.4 *For a finite language L over a binary alphabet Σ, with state complexity n, the state complexity of $\widetilde{\mathsf{D}}(L)$, in the worst case, is at least $2^t - 1$, if $n = 2t$ and by $2^t + 2^{t-2} - 1$, if $n = 2t+1$.*

6. Relation with Minimal DFCAs and D Operation

It is known that minimal DFAs may not be minimal DFCAs. For example, for the language of Figure 1, the language of a minimal DFCA has only four states, as its language is $(a^3)^\star$. This is also the case for the languages of Figure 2 and Figure 3.

One may think that if the minimal DFA is also a minimal DFCA, the dissimilarity language can be easily computed from the distinguishability language, or, if we know that only longer words from the distinguishability language are not in the dissimilarity language, then the minimal DFA is also a minimal DFCA. In general that is not the case, as the following proposition shows.

Proposition 6.1 *Let L be a finite language with the longest word of length l, and $\mathcal{A}$ the minimal DFA for L. None of the following two sentences implies the other, and both can be either true or false simultaneously.*

(1) $\mathcal{A}$ is minimal DFCA for L.
(2) $\widetilde{\mathsf{D}}(\mathcal{A}) = \mathsf{D}(\mathcal{A}) \cap \Sigma^{\leq l-1}$.

Proof. We consider all cases and for each one a witness language L and a minimal $\mathcal{A}$.

$\neg(1) \wedge (2)$**:** It is enough to consider the example in Figure 2, where $\mathsf{D}(L) \cap \Sigma^{\leq 2} = \widetilde{\mathsf{D}}(L)$, but the automaton $\mathcal{A}$ is not a minimal DFCA, as state 4 can be merged with state 1 ($4 \sim_L 1$).

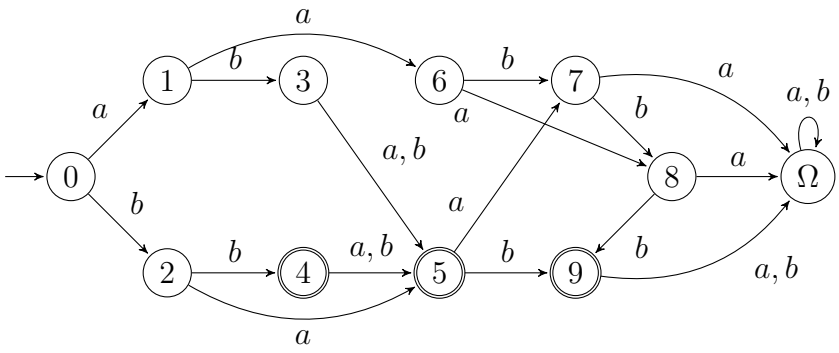

Figure 7: Example of a DFA $\mathcal{A}$ that is a minimal DFCA and $\widetilde{\mathsf{D}}(\mathcal{A}) \subsetneq \mathsf{D}(\mathcal{A}) \cap \Sigma^{\leq l-1}$

.

$(1) \wedge \neg(2)$**:** Consider the example in Figure 7, where $\mathsf{D}(L) \cap \Sigma^{\leq 5} \subsetneq \widetilde{\mathsf{D}}(L)$, because *baabb* is in $\mathsf{D}(L) \cap \Sigma^{\leq 5}$, but not in $\widetilde{\mathsf{D}}(L)$, and the automaton $\mathcal{A}$ is also a minimal DFCA.

$(1) \wedge (2)$**:** Consider the example of Figure 8. The minimal DFA $\mathcal{A}$ is also the minimal DFCA, $L = \{a, aa, ba, bb, ab, aba\}$, $l = 3$, and $\widetilde{\mathsf{D}}(L) = \mathsf{D}(\mathcal{A}) \cap \Sigma^{\leq l-1}$.

$\neg(1) \wedge \neg(2)$**:** Consider the example in Figure 5. We have $\widetilde{\mathsf{D}}(L) \subsetneq \mathsf{D}(L) \cap \Sigma^{\leq 2}$, because *bb* is in $\Sigma^{\leq 2} \cap \mathsf{D}(L)$, but not in $\widetilde{\mathsf{D}}(L)$, and the automaton $\mathcal{A}$ is also a minimal DFCA.

□

Dissimilarity operation must be distinctly analyzed. Proposition 6.1 shows that we cannot establish a direct relation between the minimality of an automaton as a DFCA for the finite language L, distinguishability language, and dissimilarity language for the same language L.

However, we do have a strong relation given by the distinguishability language of minimal words, and dissimilarity language of minimal words in relation to minimality of cover automata.

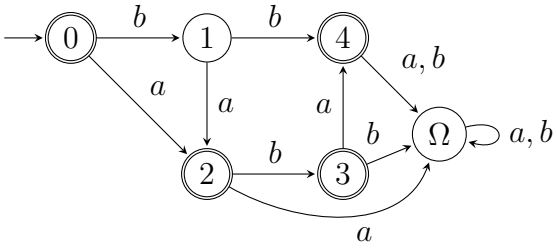

Figure 8: Example of a DFA $\mathcal{A}$ that is a minimal DFCA and $\widetilde{\mathsf{D}}(\mathcal{A}) = \mathsf{D}(\mathcal{A}) \cap \Sigma^{\leq l-1}$.

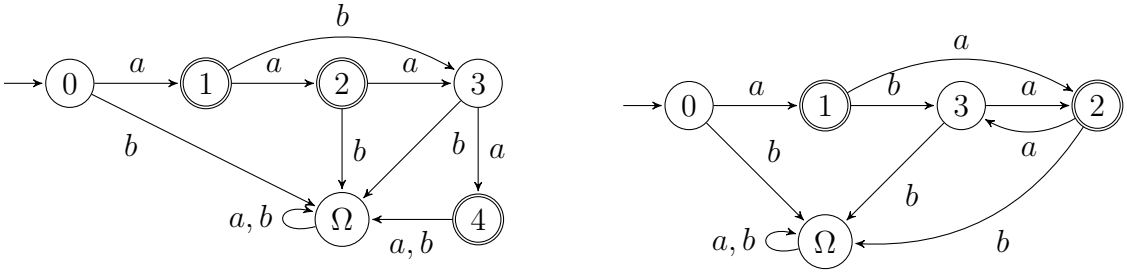

Figure 9: Example of a language $L = \{a, aa, aba, aaaa\}$, where $\underline{\mathsf{D}}(L) = \underline{\widetilde{\mathsf{D}}}(L) = \{\varepsilon, a, aa\}$, but the minimal DFA (left) is not a minimal DFCA (right).

Theorem 6.2 *Let $\mathcal{A}$ be the minimal DFA for the finite language L. If $\mathcal{A}$ is also a minimal DFCA for L, then $\underline{\widetilde{\mathsf{D}}}(\mathcal{A}) = \underline{\mathsf{D}}(\mathcal{A}) \cap \Sigma^{\leq l-1}$, but the reverse implication is not necessarily true.*

Proof. Because the DFA $\mathcal{A}$ is also a minimal DFCA for every distinct pair of states (p, q), such that $\text{level}(p) \leq \text{level}(q)$, the states p and q are not similar. Thus, we can find $v \in \Sigma^{\leq l-\text{level}(q)} \subseteq \Sigma^{\leq l-1}$ such that either $\delta(p, v) \in F$ and $\delta(q, v) \notin F$, or $\delta(p, v) \notin F$ and $\delta(q, v) \in F$. We can assume that v is the minimal with this property, thus v must be also in $\underline{\mathsf{D}}(\mathcal{A})$. It follows:

$$\underline{\mathsf{D}}(\mathcal{A}) \cap \Sigma^{\leq l-1} = \bigcup_{\substack{p \not\equiv q \\ \text{level}(p) \leq \text{level}(q)}} \underline{\mathsf{D}}_{\mathcal{A}}(p, q) = \bigcup_{\substack{p \not\equiv q \\ \text{level}(p) \leq \text{level}(q)}} \underline{\widetilde{\mathsf{D}}}_{\mathcal{A}}(p, q) = \underline{\widetilde{\mathsf{D}}}(\mathcal{A}).$$

For the reverse implication we consider the language $L = \{a, aa, aba, aaaa\}$. We have that $\underline{\mathsf{D}}(L) = \underline{\widetilde{\mathsf{D}}}(L) = \{\varepsilon, a, aa\}$, but the minimal DFA is not a minimal DFCA, as we can see in Figure 9. □

7. Conclusion

In this paper we introduced a new operation: the dissimilarity operation on finite languages. We studied the properties of dissimilarity operation and compared it with distinguishability operation; we gave several examples to show the difficulty in relating the new operation to other known operations on (regular) languages. Bounds for the state complexity of distinguishability

and dissimilarity operations on finite languages were established, and proved tight for distinguishability. We showed the connection between minimal dissimilar words and the minimality of a DFA as a DFCA, and we plan to address the corresponding relations in an extended version of the paper. It will be also interesting if we can apply the concepts of distinguishability and dissimilarity of words to define some metric between formal languages, as it is done in [15].

References

[1] Janusz A. Brzozowski. In search of most complex regular languages. *Int. J. Found. Comput. Sci.*, 24(6):691–708, 2013.

[2] Janusz A. Brzozowski, Galina Jirásková, and Chenglong Zou. Quotient complexity of closed languages. *Theory Comput. Syst.*, 54(2):277–292, 2014.

[3] Cezar Câmpeanu. Cover languages and implementations. In Stavros Konstantinidis, editor, *Proc. 18th CIAA*, volume 7982 of *LNCS*, page 1. Springer, 2013.

[4] Cezar Câmpeanu, Karel Culik II, Kai Salomaa, and Sheng Yu. State complexity of basic operations on finite languages. In Oliver Boldt and Helmut Jürgensen, editors, *4th WIA'99*, volume 2214 of *LNCS*, pages 60–70. Springer, 2001.

[5] Cezar Câmpeanu, Nelma Moreira, and Rogério Reis. The Distinguishability Operation on Regular Languages. In Suna Bensch, Rudolf Freund, and Friedrich Otto, editors, *Proc. 6th NCMA 2014*, pages 85–100. Oesterreichische Computer Gesellschaft, 2014.

[6] Cezar Câmpeanu, Andrei Paun, and Jason R. Smith. Incremental construction of minimal deterministic finite cover automata. *Theor. Comput. Sci.*, 363(2):135–148, 2006.

[7] Cezar Câmpeanu, Andrei Paun, and Sheng Yu. An efficient algorithm for constructing minimal cover automata for finite languages. *Int. J. Found. Comput. Sci.*, 13(1):83–97, 2002.

[8] Cezar Câmpeanu, Nicolae Santean, and Sheng Yu. Minimal Cover-Automata for Finite Languages. *Theor. Comput. Sci.*, 267:3–16, 2001.

[9] Jean-Marc Champarnaud, Franck Guingne, and Georges Hansel. Similarity Relations and Cover Automata. *RAIRO-Inf. Theor. Appl.*, 39:115 – 123, 2005.

[10] Cynthia Dwork and Larry J. Stockmeyer. A Time complexity Gap for Two-way Probabilistic Finite-state Automata. *SIAM Journal of Computing*, 19:1011–1023, 1990.

[11] Markus Holzer and Sebastian Jakobi. On the computational complexity of problems related to distinguishability sets. In Alexander Okhotin and Jeffrey Shallit, editors, *Proc. 17th DCFS 2015*, number 9118 in LNCS, pages 117–128. Springer, 2015.

[12] Artur Jez and Andreas Maletti. Computing all l-cover automata fast. In Béatrice Bouchou-Markhoff, Pascal Caron, Jean-Marc Champarnaud, and Denis Maurel, editors, *Proc. 16th CIAA 2011*, volume 6807 of *LNCS*, pages 203–214. Springer, 2011.

[13] Janis Kaneps and Rusins Freivalds. Minimal Nontrivial Space Complexity of Probabilistic One-Way Turing Machines. In Branislav Rovan, editor, *Proc. MFCS'90*, volume 452 of *LNCS*, pages 355–361. Springer, 1990.

[14] Heiko Körner. A time and space efficient algorithm for minimizing cover automata for finite languages. *Int. J. Found. Comput. Sci.*, 14(6):1071–1086, 2003.

[15] Manfred Kudlek and Benedek Nagy. Distances of formal languages. *Pure Mathematics and Applications - PU.MA*, 17:349–357, 2006.

[16] Mehryar Mohri, Pedro Moreno, and Eugene Weinstein. General suffix automaton construction algorithm and space bounds. *Theor. Comput. Sci.*, 410(37):3553–3562, 2009.

[17] Kai Salomaa and Sheng Yu. NFA to DFA transformation for finite languages over arbitrary alphabets. *Journal of Automata, Languages and Combinatorics*, 2(3):177–186, 1997.