State Complexity of Suffix Distance*

Timothy Ng, David Rappaport and Kai Salomaa
School of Computing, Queen's University, Kingston, Ontario K7L 3N6, Canada
tim.ng@uwaterloo.ca, {daver, ksalomaa}@cs.queensu.ca

Received (Day Month Year)
Accepted (Day Month Year)
Communicated by (xxxxxxxxxx)

The neighbourhood of a regular language with respect to the prefix, suffix and subword distance is always regular and a tight bound for the state complexity of prefix distance neighbourhoods is known. We give upper bounds for the state complexity of the neighbourhood of radius k of an n-state deterministic finite automaton language with respect to the suffix distance and the subword distance, respectively. For restricted values of k and n we give a matching lower bound for the state complexity of suffix distance neighbourhoods.

Keywords: regular languages, finite languages, state complexity, suffix distance

1. Introduction

Distances between words and languages are used in many applications [5, 7, 10, 11]. Perhaps the most commonly used distance, the Levenshtein distance (a.k.a. the edit distance), is defined in terms of the number of substitution, insertion and deletion operations needed to transform one word into another. The prefix distance [1, 4, 12] of words x and y is the sum of the lengths of the suffixes of x and y after their longest common prefix. The suffix distance (respectively, the subword distance) of two words is defined analogously in terms of the longest common suffix (respectively, subword) of the words.

Calude et al. [3] have shown that additive quasi-distances preserve regularity in the sense that a neighbourhood of a regular language is always regular. The edit distance is the best known example of additive distances. However, not all regularity preserving distances are additive. The prefix, suffix, and subword distances are not additive, but are known to preserve regularity [4].

In general, since the 90's there has been much work on the state complexity of regular languages. Recent surveys on the descriptional complexity of regular languages include [6, 8, 13]. For regularity preserving distances an important question

*A preliminary version of this paper appeared in the *Proceedings of the 19th International Conference on Descriptional Complexity of Formal Systems*, DCFS 2017, Lect. Notes Comput. Sci. 10316, Springer, 2017, pp. 287–298.

is to determine the state complexity of the distance, that is, what is the optimal size of a DFA (deterministic finite automaton) recognizing a neighbourhood of radius k of an n state DFA language. In the context of error correction this can be viewed also as the descriptional complexity of error detection [15, 16, 19]. The descriptional complexity of error systems has been considered from a different point of view by Kari and Konstantinidis [9]. They establish upper and lower bounds for the size of DFAs needed to recognize a given error system.

A neighbourhood of a language recognized by a DFA A with respect to the prefix distance, roughly speaking, can be recognized by simulating the computation of A and, for each non-final state, keeping track of the shortest path (up to the radius of the neighbourhood) to a final state of A. Additionally, we just need a number of error states equal to the radius of the neighbourhood. This means that prefix distance is an "inexpensive" operation in terms of state complexity. A tight lower bound for the state complexity of prefix distance neighbourhoods is known both for general regular languages and for finite languages [17, 18].

On the other hand, suffix distance (and subword distance) neighbourhoods are considerably more "difficult", that is, more expensive in terms of state complexity, to recognize by a DFA because the computation has no way of knowing where the longest common suffix begins. This means that the computation has to be inherently nondeterministic and as can, perhaps, be expected the state complexity of the neighbourhood depends exponentially on the size of the original DFA and the radius of the neighbourhood.

This paper shows that the suffix distance neighbourhood of radius k of an n state DFA language over an alphabet of size $\ell \geq 2$ can be recognized by a DFA with $\frac{\ell^k-1}{\ell-1} + 2^n - 1$ states when $k < n$. If A recognizes a finite language, the upper bound for the state complexity of the neighbourhood is $\frac{\ell^k-1}{\ell-1} + k \cdot 2^{\lceil \frac{n}{2} \rceil}$. We give matching lower bound constructions both for general regular languages and for finite languages using a binary alphabet in the case when n is roughly equal to $2 \cdot k$. For $k > n$, we show that the suffix distance neighbourhood can be recognized by a DFA with $(k-n) + 2^{n+1} - 2$ states and give matching lower bound constructions for both general regular languages and finite languages over an alphabet of size $n+1$. We show also that for the class of suffix-closed languages, the neighbourhood is recognized by a DFA with at most $n + k + 1$ states and that this bound is tight for all $k \in \mathbb{N}$. Finally, we derive an upper bound for the state complexity of subword distance neighbourhoods but it remains open whether the bound is tight.

2. Preliminaries

We recall some basic definitions on regular languages and distance measures. For all unexplained notions on finite automata and regular languages the reader may consult the textbook by Shallit [20] or the survey by Yu [21]. A survey of distances is given by Deza and Deza [5].

In the following Σ is always a finite alphabet, the set of words over Σ is Σ^* and ε

is the empty word. The set of nonnegative integers is $\mathbb{N}_0$. The cardinality of a finite set S is denoted $|S|$ and the powerset of S is 2^S. A word $w \in \Sigma^*$ is a *subword* of x if there exist words $u, v \in \Sigma^*$ such that $x = uwv$. If $u = \varepsilon$, then w is a *prefix* of x. If $v = \varepsilon$, then w is a *suffix* of x.

A *deterministic finite automaton* (DFA) is a tuple $A = (Q, \Sigma, \delta, q_0, F)$ where Q is a finite set of states, Σ is an alphabet, δ is a partial function $\delta : Q \times \Sigma \to Q$, $q_0 \in Q$ is the initial state, and $F \subseteq Q$ is a set of final states. We extend the transition function δ to a partial function $Q \times \Sigma^* \to Q$ in the usual way. A DFA A is *complete* if δ is defined for all $q \in Q$ and $a \in \Sigma$.

A word $w \in \Sigma^*$ is *accepted* by A if $\delta(q_0, w) \in F$. The language recognized by A is $L(A) = \{w \in \Sigma^* \mid \delta(q_0, w) \in F\}$. Two states p and q of A are equivalent if $\delta(p, w) \in F$ if and only if $\delta(q, w) \in F$ for every word $w \in \Sigma^*$. A DFA A is *minimal* if each state $q \in Q$ is reachable from the initial state and no two states are equivalent.

A *nondeterministic finite automaton* (NFA) is an extension of a DFA where the transition function is allowed to be multivalued, that is, δ is a function $Q \times \Sigma \to 2^Q$.

Note that our definition of a DFA allows some transitions to be undefined, that is, by a DFA we mean an incomplete DFA. It is well known that, for a regular language L, the sizes of the minimal incomplete and complete DFAs differ by at most one. The constructions in this paper are more convenient to formulate using incomplete DFAs but our results would not change in any significant way if we were to require that all DFAs are complete. The (incomplete deterministic) *state complexity* of a regular language L, $\mathrm{sc}(L)$, is the size of the minimal DFA recognizing L.

2.1. *Distances and neighbourhoods of regular languages*

We recall definitions of the distance measures used in the following. Generally, a function $d : \Sigma^* \times \Sigma^* \to [0, \infty)$ is a *distance* if it satisfies for all $x, y, z \in \Sigma^*$, the conditions $d(x, y) = 0$ if and only if $x = y$, $d(x, y) = d(y, x)$, and $d(x, z) \leq d(x, y) + d(y, z)$. The *neighbourhood* of a language L of radius k with respect to a distance d is the set

$$E(L, d, k) = \{w \in \Sigma^* \mid (\exists x \in L) d(w, x) \leq k\}.$$

Let $x, y \in \Sigma^*$. The *prefix distance* of x and y is the sum of the lengths of the remaining suffixes that are not part of the longest common prefix of x and y [4]. It is defined by

$$d_p(x, y) = |x| + |y| - 2 \cdot \max_{z \in \Sigma^*} \{|z| \mid x, y \in z\Sigma^*\}.$$

Similarly, the *suffix distance* of x and y is the sum of the lengths of the prefixes that are not part of the longest common suffix of x and y and is defined

$$d_s(x, y) = |x| + |y| - 2 \cdot \max_{z \in \Sigma^*} \{|z| \mid x, y \in \Sigma^* z\}.$$

The *subword distance* measures the similarity of x and y based on their longest common continuous subword and is defined

$$d_f(x,y) = |x| + |y| - 2 \cdot \max_{z \in \Sigma^*} \{|z| \mid x, y \in \Sigma^* z \Sigma^*\}.$$

The term "subword distance" is taken from Choffrut and Pighizzini [4]. However, "subword distance" has also been used for a distance defined in terms of the longest common noncontinuous subword [14].

It is known that neighbourhoods of regular languages with respect to the prefix, suffix and subword distance are always regular [4, 17]. We refer to the size of the minimal DFA recognizing the radius k neighbourhood of an n state DFA language with respect to a distance X simply as the state complexity of distance X. Tight bounds for the state complexity of the prefix distance are known [17]. Optimal bounds for the size of an NFA recognizing a suffix distance, or subword distance, neighbourhood of a regular language are also known.

Theorem 1 ([17]) *For a regular language $L \subseteq \Sigma^*$ recognized by an NFA with n states and an integer $k \geq 0$,*

(1) the nondeterministic state complexity of $E(L, d_s, k)$ is $n + k$,
(2) the nondeterministic state complexity of $E(L, d_f, k)$ is $(k+1) \cdot n + 2k$.

The bounds on the size of the NFAs imply the following upper bounds for deterministic state complexity of suffix distance and subword distance, respectively.

Proposition 2. *Suppose L is a regular language recognized by a DFA with n states and $k \in \mathbb{N}$. Then*

$$\mathrm{sc}(E(L, d_s, k)) \leq 2^{n+k} - 1 \quad \textit{and} \quad \mathrm{sc}(E(L, d_f, k)) \leq 2^{(k+1)n+2k} - 1.$$

Finally, we define the function $\psi_A : Q \to \mathbb{N}_0$ to give the length of the shortest path from the initial state q_0 to the state q. Formally, ψ_A is defined by

$$\psi_A(q) = \min_{w \in \Sigma^*} \{|w| \mid \delta(q_0, w) = q\}.$$

Note that under this definition, $\psi_A(q_0) = 0$ for the initial state q_0.

3. State Complexity of Suffix Neighbourhoods

In this section, we consider the deterministic state complexity of suffix distance neighbourhoods. First, we construct a DFA for the neighbourhood of an n-state DFA of radius k with respect to the suffix distance d_s, when $k < n$ and then give a matching lower bound when $k = \lfloor \frac{n}{2} \rfloor$ for an n state DFA.

Proposition 3. *Let $n > k \geq 0$ and L be a regular language recognized by a DFA with n states over an alphabet Σ, with $|\Sigma| \geq 2$. Then there is a DFA recognizing $E(L, d_s, k)$ with at most $\frac{|\Sigma|^k - 1}{|\Sigma| - 1} + 2^n - 1$ states.*

Proof. Let L be recognized by the DFA $A = (Q, \Sigma, \delta, q_0, F)$ with $|Q| = n$. We construct a DFA $A' = (Q', \Sigma, \delta', q_0', F')$ that recognizes the neighbourhood $E(L, d_s, k)$. First, let us consider what it means if $w \in E(L(A), d_s, k)$. If w is in the neighbourhood, then there exists a word x recognized by A such that $d_s(w, x) \leq k$. In other words, we can write $w = w'z$ and $x = x'z$ for words $w', x', z \in \Sigma^*$ such that $|w'| + |x'| \leq k$. However, while A' reads w, it is not known when such a common suffix z might begin. A common suffix may begin in each of the first k symbols of w, so A' must keep track of and compute all possible common suffixes that begin on each of the first k symbols of w.

We define the state set

$$Q' = \{0, \ldots, k\} \times 2^Q$$

and we define the initial state by $q_0' = (0, \{q \in Q \mid \psi_A(q) \leq k\})$. The set of final states is given by

$$F' = \{0, \ldots, k\} \times \{P \subseteq Q \mid P \cap F \neq \emptyset\}.$$

In other words, a state (i, P) of A' is final if and only if P contains a final state of A.

The state set consists of subsets of the original state set with a counter component. The operation of the automaton begins by counting the first k steps of computation. On the ith step of the initial k steps, the automaton reaches a state containing those states reachable from direct transitions from the set of states from the $(i-1)$th computation step and adds every state reachable from q_0 within $k - i$ steps and the counter component is incremented. After the kth computation step, no further steps need to be counted and the counter is no longer incremented since states are no longer added to the existing state sets.

The transition function δ' is defined for $a \in \Sigma$ by

- $\delta'((i, P), a) = (i + 1, X)$ for $0 \leq i \leq k - 1$, where X is defined as
$$X = \{\delta(p, a) \mid p \in P\} \cup \{q \in Q \mid \psi_A(q) \leq k - (i + 1)\},$$
- $\delta'((k, P), a) = (k, \{\delta(p, a) \mid p \in P\})$.

We now show that reading a word $w \in \Sigma^*$ reaches the state (i, P) if and only if there exists a word $x \in \Sigma^*$ such that $w = w'z$ and $x = x'z$ where $|w'| \leq i$, $|x'| \leq k - i$ and $\delta(q_0, x) \in Q$.

First, suppose that $\delta'(q_0', w) = (i, P)$. We write $w = w'z$ with $w', z \in \Sigma^*$ which may possibly be empty. By definition, $\delta'(q_0', w') = (|w'|, P')$ if $|w'| \leq k$ and P' contains all states q such that $\psi_A(q) \leq k - |w'|$. In other words, these are states $\delta(q_0, x')$ where $x' \in \Sigma^*$ is of length $\leq k - |w'|$. Choose q' to be one of these states and consider the state $\delta(q', z) = q$. Since $q' \in P'$ and $\delta'(q_0', w) = \delta'((|w'|, P'), z) = (i, P)$, we have $q \in P$. Thus, there exists a word $x = x'z$ such that $|x'| \leq k - i$ and $\delta(q_0, x) \in P$.

Now, conversely, suppose that for an input word $w = w'z$ with $|w'| \leq i$, there exists a word $x = x'z$ with $|x'| \leq k - i$ such that $q = \delta(q_0, x) \in P$. Since $|x'| \leq k - i$,

let $q' = \delta(q_0, x')$ and we have $\psi_A(q') \leq k - i$. Then this means we have $\delta'(q'_0, w') = (|w'|, P')$ with $q' \in P'$. Since $\delta(q', z) = q$, we have $\delta'((|w'|, P'), z) = (i, P)$ with $q \in P$ as desired.

Thus, $\delta(q'_0, w) \in F'$ if and only if there exists $x \in L$ such that $|w'| + |x'| \leq k$ for $w = w'z$ and $x = x'z$.

However, not all $(k+1) \cdot 2^n$ in $\{0, \ldots, k\} \times 2^Q$ are reachable. Note that for $i < k$, the only words that can be read to reach a state (i, P) are those of length exactly i. However, there are only $|\Sigma|^i$ words of length exactly i. Thus, the maximum number of reachable states for $0 \leq i < k$ is

$$\sum_{i=0}^{k-1} |\Sigma|^i = \frac{|\Sigma|^k - 1}{|\Sigma| - 1}.$$

Furthermore, the state $\emptyset \subseteq Q$ is unreachable. Thus, A' has at most $\frac{|\Sigma|^k - 1}{|\Sigma| - 1} + 2^n - 1$ reachable states. □

The statement of Proposition 3 assumes that the cardinality of the alphabet is at least two. For suffix distance neighbourhoods of unary languages we have the following bounds.

Lemma 4. *Let A be an n state DFA over a unary alphabet and $k \in \mathbb{N}$. Then*

$$\mathrm{sc}(E(L(A), d_s, k)) \leq \begin{cases} n & \text{if } L(A) \text{ is infinite and } n > 2k, \\ \max\{1, n-k\} & \text{if } L(A) \text{ is infinite and } n \leq 2k, \\ n+k & \text{if } L(A) \text{ is finite.} \end{cases}$$

For every $n, k \in \mathbb{N}$ there exists an n state unary DFA A recognizing a finite language such that $\mathrm{sc}(E(L(A), d_s, k)) = n + k$. For values $n, k \in \mathbb{N}$ where $n > 2k$ there exists a unary DFA A with n states recognizing an infinite language such that $\mathrm{sc}(E(L(A), d_s, k)) = n$.

Proof. Let $A = (Q, \Sigma, \delta, q_0, F)$. Recall that a unary DFA always consists of a sequence of states, called a tail, followed by a cycle of states, which may possibly be empty. We observe that over a unary alphabet Σ, $d_s(x, y) = d_p(x, y)$ for all $x, y \in \Sigma^*$. Therefore, in the following, we will construct minimal DFAs for the language $E(L(A), d_p, k)$.

If $L(A)$ is finite, we make a state q of A into an accepting state if there exists a word $w \in \Sigma^*$ such that $|w| \leq k$ and $\delta(q, w) \in F$. We also add k new states. Thus, the new automaton has at most $n + k$ states. To see this bound is reachable, we consider the language $L_n = \{a^{n-1}\}$ which has a DFA with n states. Then it is clear that for every k, the language $E(L_n, d_p, k)$ requires $n + k$ states.

For the infinite case, we first consider when $n > 2k$. Again, we make a state q with $\delta(q, w) \in F$ and $|w| \leq k$ a final state in the new automaton. But instead of adding k new states to the automaton, for states q where there is a final state f such that $\delta(f, w) = q$ and $|w| \leq k$, we make q a final state. To see this bound

is reachable, we consider the language $L_n = (a^{n-1})^*$ recognized by a DFA with n states. Let $0 \leq i, j \leq n-1$ and without loss of generality let $i < j$ and consider two words a^i and a^j.

(1) If $i + (n - j) \leq k$, then $a^j a^{n-j} a^{k-(i+(n-j))+1} \in E(L_n, d_p, k)$ and $a^i a^{n-j} a^{k-(i+(n-j))+1} \notin E(L_n, d_p, k)$.
(2) If $k < i+(n-j) < n-k$, then $a^i a^{n-j} \notin E(L_n, d_p, k)$ and $a^j a^{n-j} \in E(L_n, d_p, k)$.
(3) If $n-k \leq i+(n-j) \leq n-1$, then $a^j a^{n-j} a^{k+1} \notin E(L_n, d_p, k)$ and $a^i a^{n-j} a^{k+1} \in E(L_n, d_p, k)$.

Thus, for a DFA recognizing $E(L_n, d_p, k)$, the states reached by a^i and a^j cannot be equivalent.

Finally, we consider when $n \leq 2k$. Suppose A has $n = \ell + m$ states consisting of a tail of ℓ states and a cycle of m states. Since $L(A)$ is infinite, $m \geq 1$. If $\ell > k$, then $k > 2k - \ell \geq m$. In this case, the minimal DFA for $E(L_n, d_p, k)$ will have a tail of size at most $n - k$ followed by a cycle consisting of a single state with a self loop. If $\ell \leq k$, then we have $E(L_n, d_p, k) = a^*$, since $m < 2k$. Thus, the DFA for $E(L_n, d_p, k)$ has at most either 1 or $n - k$ states. These bounds are reached by languages $L_n = a^\ell (a^m)^*$ with $n = \ell + m$. □

For a constant size alphabet, the bound of Proposition 3 is significantly better than the bound implied by known results on nondeterministic state complexity in Proposition 2. Next we show that, at least for some values of the radius k, the bound of Proposition 3 is tight.

Lemma 5. *Let $k = \lfloor \frac{n}{2} \rfloor$. Then there exists a DFA A_n with n states over a binary alphabet such that*

$$\mathrm{sc}(E(L(A_n), d_s, k)) \geq 2^k + 2^n - 2.$$

Proof. Let $A_n = (Q_n, \{a, b\}, \delta_n, 0, \{0\})$, shown in Figure 1, with $Q_n = \{0, \ldots, n-1\}$ and the transition function δ_n is defined by

- $\delta_n(i, b) = i + 1 \bmod n$ for all $0 \leq i \leq n - 1$,
- $\delta_n(i, a) = i + 1 \bmod n$ for $i = 0, \ldots, k-1, k+1 \ldots, n-1$.

The DFA A_n operates as follows. From every state $i < k$, A_n reaches $i+1$ on any symbol. From state k, the state $k + 1$ is reachable only on the symbol b. Otherwise, on reading the symbol a, the computation terminates. For each state $i > k$, the DFA A_n can reach the state $i + 1$ on any symbol, except from state $n - 1$, which has a transition to state 0.

We want to show that the automaton A'_n from the construction of Proposition 3 is the minimal DFA for $E(L(A_n), d_s, k)$. We do this by showing that all states of A'_n are reachable and no two states are equivalent.

First, we show that all the states are reachable. For the first k steps of the computation, states are of the form (i, P) where $P \subseteq Q_n$. Since the number of

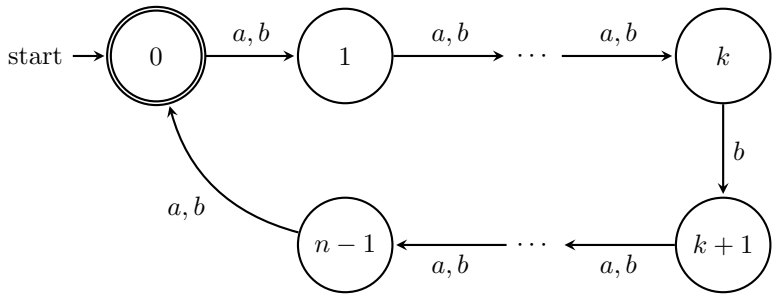

Fig. 1. The DFA A_n.

reachable states of this form is constrained by the number of words of length up to $k-1$, there are only $\frac{|\Sigma|^k-1}{|\Sigma|-1} = 2^k - 1$ such reachable states. Thus, we need to show that reading each word of length up to $k-1$ reaches a different state. Then since a state (i, P) is reachable on a word of length exactly i, we need only show that reading no two words of the same length $i < k$ will reach the same state (i, P).

Let $w = w_1 w_2 \cdots w_i$ be a word that reaches the state (i, P_i). We claim that $P_i = \{0, \ldots, k\} \cup R_i$, where $R_i = \{j_1, \ldots j_r\} \subseteq \{k+1, \ldots, k+i\}$ such that $k+1 \leq j_1 < \cdots < j_r \leq k+i$. Furthermore, we have $j_\ell \in R$ if $w_{i-j_\ell} = b$ and all other symbols of w are a's. We begin by observing that for $i = 0$, we have $w = \varepsilon$ and $P_0 = \{0, \ldots, k\}$.

Now, suppose our claim holds for arbitrary i and consider the case for $i+1$. That is, we have reached the state (i, P_i) on the word $w = w_1 \cdots w_i$ and $P_i = \{0, \ldots, k\} \cup R_i$ as defined above and we now consider the transition upon reading w_{i+1}. First, we consider every element $q \in P_i \setminus \{k\}$ and note that $\delta(q, w_{i+1}) = q + 1$. This gives us $\{1, \ldots, k, j_1 + 1, \ldots, j_r + 1\} \subseteq P_{i+1}$. We also observe that since $\psi_{A_n}(0) = 0$, we have $0 \in P_{i+1}$. Finally, we consider $\delta(k, w_{i+1})$. If $w_{i+1} = b$, then by definition $\delta(k, w_{i+1}) = k + 1$. Otherwise, if $w_{i+1} = a$, the transition is undefined.

It remains to confirm that each state $j_\ell \geq k + 1$ in P_{i+1} satisfies the condition that $j_\ell \in P_{i+1}$ only if $w_{i+1-j_\ell} = b$. By our assumption, we know that j_ℓ satisfied this condition in P_i, since j_ℓ takes a transition to state $j_\ell + 1$ in P_{i+1}. Then we simply verify that $w_{i+1-(j_\ell+1)} = w_{i-j_\ell} = b$. Thus, reading each word w_i of length i reaches a state (i, P_i) where $P_i = \{0, \ldots, k\} \cup R_i$ and R_i depends on the word w_i as defined above.

Next, we consider states of the form $(k, T) \subseteq \{k\} \times 2^{Q_n}$, which are reachable on words of length greater than or equal to k. First, we consider two states (k, S) and (k, T) with $S \subsetneq T \subseteq Q_n$. We want to show that (k, S) is reachable from (k, T). Let $T \subseteq Q_n$ be some subset $\{i_1, \ldots, i_t\}$ with $i_1 < i_2 < \cdots < i_t$. Let $i_s \in T$ and consider the subset $T \setminus \{i_s\}$. For $i_s \leq k$, the state $(k, T \setminus \{i_s\})$ is reachable on the word $b^{k-i_s} a b^{n-(k+1)+i_s}$. For $i_s > k - 1$, the state $(k, T \setminus \{i_s\})$ is reachable on the word $b^{n-i_s+k} a b^{i_s-k-1}$. Having established that (k, S) is reachable from (k, T) for

all $S \subseteq T$, we observe that the state $(k, \{0, \ldots, n-1\})$ is reachable on the word b^k. Then every state (k, T) with subset $T \subseteq \{0, \ldots, n-1\}$ (except the empty set) is reachable in A'_n and we have $2^n - 1$ reachable states of the form (k, T).

To show that states of A'_n are pairwise distinguishable, we first consider two states (k, T) and (k, T') with subsets $T, T' \subseteq Q_n$ and $T \neq T'$. Then there is some element $i_s \in Q_n$ such that $i_s \in T$ and $i_s \notin T'$. Then a final state is reachable from the state (k, T) on the word b^{n-i_s}, while from (k, T'), no final state can be reached on the same word.

Next, consider two states $(i, P), (i, P')$ with $P, P' \subseteq Q_n$ and $P \neq P'$ and $0 \leq i \leq k$. Recall that when $i \leq k$, the state P (P' respectively) is of the form $\{0. \ldots, k\} \cup R$ (R' respectively). Then let $i_s \in R$ be an element that is not in R'. On the word b^{k-i}, the state $(k, \hat{P})$ is reachable from (i, P) and $(k, \hat{P}')$ is reachable from (i, P'). We note that we now have an element $i_s + k - i$ in $\hat{P}$ that is not in $\hat{P}'$. Thus, we have two states $(k, \hat{P})$ and $(k, \hat{P}')$ with $\hat{P} \neq \hat{P}'$ and we can apply the same argument from above.

Now, we consider two states (i, P) and (i', P') with $i < i'$ and $P \neq P'$. Let i_s be an element in P' that is not in P. By definition, we have $i_s > k$. On the word $a^{k-i'}$, the state $(i + k - i', R)$ is reachable from (i, P) and (k, R') is reachable from (i', P'), such that the element $i_s + k - i'$ is in R' but is not in R. Then on the word a^k, the state (k, S) is reachable from $(i + k - i', R)$ and (k, S') from (k, R'). There exists an element $i_s - i' \in S'$ that is reachable from $i_s + k - i' \in R'$. However, as $i_s + k - i' \notin R$, we have $i_s - i' \notin S$. Thus, we have two states (k, S) and (k, S') with $S \neq S'$ and we can apply the prior argument again.

To conclude, we observe that this lower bound is tight only when $k = \lfloor \frac{n}{2} \rfloor$. If $k < \lfloor \frac{n}{2} \rfloor$, then the state $(k, \{0, \ldots, n-1\})$ is unreachable. If A'_n reads the word b^k, the automaton reaches the state $(k, \{0, \ldots, 2k-1\}$ and the state $(k, \{1, \ldots, 2k\})$ can be reached by reading b. Therefore, not all $2^n - 1$ states of the form (k, T) are reachable.

In the other case, if $k > \lfloor \frac{n}{2} \rfloor$, then there exist words of length $\lfloor \frac{n}{2} \rfloor < \ell \leq k$ that reach the same states (ℓ, P) for some $P \subseteq Q_n$. To see this, we observe that reading both words b^ℓ and $ab^{\ell-1}$ brings the automaton to the state $(\ell, \{0, \ldots, n-1\})$ for every $\lfloor \frac{n}{2} \rfloor < \ell \leq k$. Thus, there are fewer than 2^k states that can be reached on words of length less than k.

Thus we have shown that there are $2^k + 2^n - 2$ reachable states and they are all pairwise distinguishable for $k = \lfloor \frac{n}{2} \rfloor$. □

From Proposition 3 and Lemma 5, we obtain the following theorem.

Theorem 6. *Let $n > k$ and let L be a regular language recognized by an n-state DFA over an alphabet Σ with $|\Sigma| \geq 2$. Then a DFA recognizing $E(L, d_s, k)$ requires at most $\frac{|\Sigma|^k - 1}{|\Sigma| - 1} + 2^n - 1$ states. There is a family of DFAs with n states over a binary alphabet which reaches this bound when $k = \lfloor \frac{n}{2} \rfloor$.*

Now we will consider the case when the distance k is greater than the number of states n of the given DFA and give a matching lower bound.

Proposition 7. *Let $k > n > 0$ and L be a regular language recognized by a DFA with n states over an alphabet Σ with $|\Sigma| \geq 2$. Then there is a DFA recognizing $E(L, d_s, k)$ with at most $(k-n) + 2^{n+1} - 2$ states.*

Proof. Let $A = (Q, \Sigma, \delta, q_0, F)$ with $|Q| = n$. Then we follow the construction given in the proof of Proposition 3 to obtain the DFA $A' = (Q', \Sigma, \delta', q_0', F')$ that recognizes the neighbourhood $E(L(A), d_s, k)$ with $k > n$. We note that $\psi_A(q) \leq n$ for all $q \in Q$ and thus by the definition of the transition function, we have, for $0 \leq i \leq k-n$ and all words w of length i, $\delta(q_0, w) = (i, Q)$. This gives $k-n$ states. Then on the following n steps, we proceed as in the rest of the proof of Proposition 3. This suggests that there are at most $\frac{|\Sigma|^n - 1}{|\Sigma| - 1}$ states. However, in this case, there are far fewer states than this.

To consider how many states there are, we observe that the above bound requires that each word of length $i > k-n$ reaches a different state (i, P), giving us a total of $|\Sigma|^{i-(k-n)}$ states for each i. Then we must consider how many different subsets $P \subseteq Q$ are reachable. Recall that by definition, all states q with $\psi_A(q) \leq k-i$ are contained in P for (i, P). Thus, on step i, for two states (i, P) and (i, P') both P and P' contain the subset $\{q \in Q \mid \psi_A(q) \leq k-i\}$. Then if P and P' are different, they must contain different subsets of the set $\{q \in Q \mid \psi_A(q) > k-i\}$.

Let j be the size of the set $\{q \in Q \mid \psi_A(q) > k-i\}$. Then in order for each word of length i to reach a different state, we must have $|\Sigma|^{i-(k-n)} \leq 2^j$ different subsets. This means that we must have at least $(i-(k-n)) \cdot \log_2 |\Sigma|$ states q with $\psi_A(q) > k-i$ on step i of a computation on A'. In other words, for each $1 \leq i \leq \max_{q \in Q} \psi_A(q)$, there are at least $\log_2 |\Sigma|$ states q with $\psi_A(q) = i$. However, since $k > n$, the number of states of A are further restricted by this condition.

Let $\ell = \max_{q \in Q} \psi_A(q)$. Then there are at most

$$k - \frac{n}{\log_2 |\Sigma|} + \frac{|\Sigma|^{\frac{n}{\log_2 |\Sigma|}} - 1}{|\Sigma| - 1}$$

reachable states for words of length up to k. We observe that this is maximized when $|\Sigma| = 2$. That is, for any alphabet of size at least 2, the maximum is achieved when we have for each i exactly one state q such that $\psi_A(q) = i$. This gives us a maximum of $2^n - 1$ reachable states of the form (i, P) for $i < k$.

After the kth step of computation, there are $2^n - 1$ reachable states of the form (k, P) as usual. This gives a total of at most $(k-n) + 2^{n+1} - 2$ states. □

We will show that the bound from Proposition 7 is reachable for a family of n state DFAs over an alphabet of size $n+1$.

Lemma 8. *Let $k > n > 0$. Then there exists a DFA B_n with n states over an*

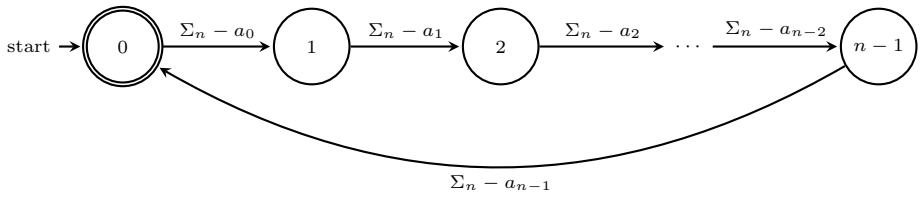

Fig. 2. The DFA B_n.

alphabet of size $n+1$ *such that*

$$\mathrm{sc}(E(L(A_n), d_s, k)) \geq (k-n) + 2^{n+1} - 2.$$

Proof. Let $B_n = (Q_n, \Sigma_n, \delta_n, 0, \{0\})$, shown in Figure 2, with $\Sigma_n = \{a_0, a_1, \ldots, a_n\}$ and the transition function is defined by

$$\delta(i, a_j) = i + 1 \bmod n \quad \text{for all } 0 \leq i \leq n-1,\ 0 \leq j \leq n, \text{ and } i \neq j.$$

We want to show that B'_n, acquired from the construction of Proposition 3 is the minimal DFA for $E(L(B_n), d_s, k)$.

First, we show that all states of B'_n are reachable. States (i, Q) for $0 \leq i \leq k-n$ are reachable on any word of length i, say a_n^i. Next, we show that for each $k-n < i < k$ we can reach 2^j states for each $i = k-n+j$. Recall that for every state (i, P), the subset P contains the set of states $\{q \in Q \mid \psi_A(q) \leq k-i\}$. But since $i = k-n+j$ means $k-i = n-j$, the two states (i, P) and (i, P') must have P and P' which are different subsets $\{q \in Q \mid \psi_A(q) > n-j\}$.

Then for the state (i, P) we have $i = k-n+j$ and $P = Q \setminus R$ with $R \subseteq \{n-j+1, \ldots, n-1\}$. We write $R = \{i_1, \ldots, i_m\}$ where $m < j$ and $i_1 > i_2 > \cdots > i_m$. We observe that (i, P) is reachable on the word $a_n^{k-n} a_{i_1-1} a_{i_2-1} \cdots a_{i_m-1}$. In this way, we can reach any subset R of j states on step i, giving us a total of $1 + 2 + \cdots + 2^{n-1} = 2^n - 1$ reachable states (i, P) with $k-n < i < k$.

Finally, we show that states (k, T) are reachable for all $T \subseteq Q$. First, we note that (k, Q) is reachable on the word a_n^k. We will show how to reach any subset $T \subseteq Q$ from (k, Q). Let $T = Q \setminus \{i_1, i_2, \ldots, i_m\}$. Then we can reach (k, T) on the word $a_n^k a_{i_1} a_n^{n-2} a_{i_2} a_n^{n-2} \cdots a_n^{n-2} a_{i_m} a_n^{n-2}$.

In total, this gives us $(k-n) + 2^n - 1 + 2^n - 1$ reachable states. Now, we will show that these states are pairwise inequivalent. Consider two distinct states (i, T) and (i', T'). Setting $i = i'$, we have $T \neq T'$ and there is an element $i_t \in T$ but $i_t \notin T'$. Then a final state is reachable from the state (i, T) by reading the word $a_n^{n-i_t}$ but no final state can be reached from the state (i, T') on the same word.

Next, set $i \neq i'$. Without loss of generality, let $i' > i$. Reading the word $a_n^{k-i'} a_{n-1}$ from (i', T') brings the automaton to the state (k, S'), where $0 \notin S'$, and thus (k, S') is not a final state. However, reading the same word from (i, T) brings the automaton

to the state $(i+k-i', S)$. Note that $i+k-i' < k$ and in this case, we have $0 \in S$, since for all states (j, P) with $j < k$, we have $0 \in P$. Thus, (i, T) is a final state.

Thus, we have shown that there are $(k-n) + 2^{n+1} - 2$ reachable states in A' and they are all pairwise distinguishable. □

Proposition 7 and Lemma 8 can then be summarized in the following theorem.

Theorem 9. *Let $k > n$ and let L be a regular language recognized by an n-state DFA over an alphabet Σ with $|\Sigma| \geq 2$. Then a DFA recognizing $E(L, d_s, k)$ requires at most $(k-n) + 2^{n+1} - 2$ states. There is a family of DFAs with n states over an alphabet of size $n+1$ which reaches this bound.*

3.1. *State Complexity of Subword Distance*

Now, we give an upper bound on the deterministic state complexity of subword neighbourhoods by giving a construction for a DFA for the neighbourhood of radius k with respect to the subword distance d_f. In the construction we again assume that the cardinality of the alphabet is at least two. For unary alphabets, the subword distance coincides with the suffix distance and a tight bound is obtained from Lemma 4.

Proposition 10. *Let $n > k \geq 0$ and L be a regular language recognized by a DFA with n states over the alphabet Σ with $|\Sigma| \geq 2$. Then there is a DFA recognizing $E(L, d_f, k)$ with at most $\frac{|\Sigma|^k - 1}{|\Sigma| - 1} + (k+2) \cdot 2^{n \cdot (k+1)}$ states.*

Proof. Let L be recognized by the DFA $A = (Q, \Sigma, \delta, q_0, F)$ with $|Q| = n$. For a given DFA $A = (Q, \Sigma, \delta, q_0, F)$, we define the function $\varphi_A : Q \to \mathbb{N}_0$ to be the length of the shortest path from the state q to a reachable final state. Formally, we define φ_A by

$$\varphi_A(q) = \min_{w \in \Sigma^*} \{|w| \mid \delta(q, w) \in F\}.$$

Note that under this definition, if $q \in F$, then $\varphi_A(q) = 0$.

We construct a DFA $A' = (Q', \Sigma, \delta', q_0', F')$. First, we define the set $R = Q \times \{0, \ldots, k\}$. The state set Q' is then defined by

$$Q' = \{0, 1, \ldots, k\} \times 2^R \times \{0, \ldots, k+1\}.$$

We define the initial state by $q_0' = (0, I_0, 0)$, where I_i is defined for $0 \leq i \leq k$ by

$$I_i = \{(q, \psi_A(q) + i) \in R \mid \psi_A(q) \leq k - i\}$$

The set of final states F' is given by

$$F' = \{(i, P, j) \in Q' \mid j \leq k \text{ or } P \cap Q \times \{0, \ldots, k-i\} \neq \emptyset\}.$$

States of A' are of the form (i, P, j). The first component i counts the first k steps of computation. As in the automaton from the proof of Proposition 3, on the ith

computation step, the subset P contains tuples for all states reachable from direct transitions from the $(i-1)$th computation step and all of those states which are reachable from q_0 within $k-i$ steps.

The second component P is a subset of R. Elements of R are 2-tuples (p, ℓ) where the first component is a state $p \in Q$ of the original automaton A and the second component is an integer $0 \leq \ell \leq k$. The first component p is the state of a simulation of a computation on A. The second component remembers the length of the word that was read by A' when the tuple was first added to R.

The third component j allows the DFA A' to keep track of the length of the simulated computation of A on a suffix that is different from the actual input. This value takes into account the length of the simulated computation of A on a prefix that is different from the actual input of A'.

The transition function δ' is defined for $a \in \Sigma$ by

- $\delta'((i, P, j), a) = (i+1, I_i \cup P', j')$ for $P \subseteq R$ and $0 \leq i \leq k-1$,
- $\delta'((k, P, j), a) = (k, P', j')$ for $P \subseteq R$,

where $P' = \{(\delta(q, a), \ell) \in R \mid (q, \ell) \in P\}$ and

$$j' = \min\{\ell + \varphi_A(\delta(q, a)) \mid (q, \ell) \in P\} \cup \{j+1\} \cup \{k+1\}.$$

Again, note that the value of the second component of a member of R does not change during any computation, as it is set when it is first added to a subset in a computation. We also note that in the simulated computation of the DFA A, the DFA A' only keeps track of the length of the minimal suffix that deviates from the input word for A'.

We show that on reading a word $w \in \Sigma^*$, the automaton A' reaches the state (i, P, j) if and only if for $w = w_1 z w_2$, there exists a word $x = x_1 z x_2$ in $L(A')$ such that $|w_1| \leq i$, $|x_1| \leq k-i$, and $j \leq |x_1| + |w_1| + |x_2| + |w_2|$ if $j \leq k$.

First, suppose that there exists a word $w \in \Sigma^*$ such that $\delta'(q_0', w) = (i, P, j)$. We consider $w' = w_1 z$ and note that by definition $\delta'(q_0', w_1) = (|w_1|, P', j')$ for $|w_1| \leq k$. Then P' contains some state (p, ℓ) such that $\ell = \psi_A(p) \leq k - |w_1|$. This means that there is a word x_1 such that $\delta(q_0, x_1) = p$ and $|x_1| \leq k - |w_1|$.

If we choose x_1 to minimize its length, we have $\ell = \psi_A(p) = |x_1|$. By definition, we have $(p, \psi_A(p) + |w_1|) = (p, |x_1| + |w_1|) \in P'$. Now, consider the state $q = \delta(p, z)$. Since $(p, |x_1| + |w_1|) \in P'$ and $\delta'(q_0', w') = \delta'((|w_1|, P', j'), z) = (i'', P'', j'')$, we have $(q, |x_1| + |w_1|) \in P''$ where $q = \delta(q_0, x)$.

Now, there exists a word x_2 such that $\delta(q, x_2) \in F$. We choose the shortest such x_2 so that $|x_2| = \varphi_A(q)$. For the current state (i'', P'', j''), we have via $(q, |w_1| + |x_1|)$,

$$j'' \leq \varphi_A(q) + |x_1| + |w_1|.$$

Then $\delta'(q_0', w) = \delta'((i'', P'', j''), w_2) = (i, P, j)$ with $j \leq |w_2| + |x_1| + |w_1| + \varphi_A(q)$. If $j \leq k$, we have $j \leq |x_1| + |w_1| + |x_2| + |w_2|$, as required.

We now show the other direction. For $w = w_1 z w_2$, we suppose there exists a word $x = x_1 z x_2$ in $L(A)$ such that $|w_1| \leq i$, $|x_1| \leq k-i$, and $j \leq |x_1| + |w_1| + |x_2| + |w_2|$ if

$j \leq k$. Reading w_1 takes the automaton to the state $(|w_1|, P', j')$. Let $p = \delta(q_0, x_1)$. We have $\psi_A(p) \leq |x_1| \leq k - i$. Then $(p, \psi_A(p) + |w_1|) \in P'$. Next, reading z takes the automaton to state (i'', P'', j''). If $q = \delta(p, z)$, then $(q, \psi_A(p) + |w_1|) \in P''$. Recall that $\delta(q, x_2) \in F$, which means that $\varphi_A(q) \leq |y_2|$. This implies that $j'' \leq \varphi_A(q) + \psi_A(p) + |w_1|$.

Reading w_2 from (i'', P'', j'') brings us to the state (i, P, j). If $j \leq k$, we have

$$j \leq |w_2| + j'' \leq |w_2| + \varphi_A(q) + \psi_A(p) + |w_1| \leq |w_1| + |w_2| + |x_1| + |x_2|$$

as required.

Then if (i, P, j) is a final state of A', this means that reading a word $w \in \Sigma^*$ reaches the state (i, P, j) if and only if $|x_1| + |w_1| + |x_2| + |w_2| \leq k$. This means that $d_f(w, L(A))$ and therefore $w \in E(L(A), d_f, k)$.

However, note that not all $(k + 1) \cdot (k + 2) \cdot 2^{n(k+1)}$ are reachable. Recall that for $i < k$, the only words that can reach a state (i, P, j) are of length exactly i. However, there are only $|\Sigma|^i$ words of length exactly i. Thus, the maximum number of reachable states with $i < k$ is

$$\sum_{i=0}^{k-1} |\Sigma|^i = \frac{|\Sigma|^k - 1}{|\Sigma| - 1}.$$

Thus, A' has at most $\frac{|\Sigma|^k - 1}{|\Sigma| - 1} + (k + 2) \cdot 2^{n(k+1)}$ reachable states. □

The bound of Proposition 10 is significantly better than the bound implied by nondeterministic state complexity [17] (in Proposition 2) for a fixed alphabet Σ. However, we do not know whether the bound is the best possible.

4. State Complexity of Suffix Distance on Subregular Languages

Here, we consider the state complexity of neighbourhoods with respect to the suffix distance of subregular languages. First, we consider neighbourhoods of finite languages.

Proposition 11. *Let $n > k \geq 0$ and L be a finite language recognized by a DFA with n states over a binary alphabet. Then there is a DFA recognizing $E(L, d_s, k)$ with at most $2^k + k \cdot 2^{\lfloor \frac{n}{2} \rfloor} - 1$ states.*

Proof. We use the construction for A' from the proof of Proposition 3. Observe that, as is the case for general regular languages, not all $(k + 1) \cdot 2^n$ states are reachable. Recall that the states of A' are pairs (i, P) where i is a counter from 0 to k and P is a subset of states of A and that a word w reaches a state (i, P) if and only if there exists a word $x \in \Sigma^*$ such that $w = w'z$ and $x = x'z$ where $|w'| \leq i$, $|x'| \leq k - i$ and $\delta(q_0, x) \in Q$. We also note that for $i < k$, any state (i, P) with $P \subseteq Q$ is reachable on a word of length exactly i. This gives us at most $\sum_{i<k} 2^i = 2^k - 1$ reachable states of the form (i, P) for $i < k$.

It remains to show how many states (k, P) with $P \subseteq Q$ are reachable. Since P is a subset of the set of states of A, we would like to know how many different subsets P exist such that (k, P) is reachable. Since A recognizes a finite language, there exists at least one state q of A with $\psi_A(q) = i$ that is reachable on some word of length i and is not reachable on any word of length $j > i$. Note that this property does not hold for states of the form (i, P) with $i < k$. To see this, we consider some i and observe that every state $q \in Q$ with $\psi_A(q) \leq k - i$ is in some subset P with (i, P) reachable for all $i < k$ by definition. Hence, we can restrict consideration to states where the first component is k.

Let (k, T) be a state that is reached on a word w of length k. Since A' is deterministic, there are up to 2^k possible such states. Let $R_i \subseteq Q$ denote the set of states of A that are not contained in any state $P \subseteq Q$, where (k, P) is reachable on a word of length greater than $k + i$. In other words, R_i is the set of states of A which become unreachable in A on a word of length i. We note that R_i must contain at least one element, since A recognizes a finite language.

We write $T = R \cup S$, where $R \subseteq \bigcup_{0 \leq i \leq k} R_i$ and $S \subseteq Q \setminus R$. We have $|Q \setminus R| \leq n-k$, since $k < n$. From this, we can see that to maximize the number of reachable states, each R_i must contain at most one element. This gives us a total of 2^{n-k} possible subsets S.

Then for each set $T = R \cup S$ that is reachable on a word of length k, there is a state $T_i = (R \setminus \bigcup_{j=0}^{i} R_j) \cup S$ that is reachable on a word of length $k + i$ for $1 \leq i \leq k$. Since each R_i has one element, each subset S is contained in up to k different subsets of Q that are reachable in A'. This gives $k \cdot 2^{\lfloor \frac{n}{2} \rfloor}$ possible subsets that can be reached on each word of length greater than k.

Thus, A' can have up to $\frac{|\Sigma|^k - 1}{|\Sigma| - 1} + k \cdot 2^{\lfloor \frac{n}{2} \rfloor} - 1$ states in total. □

The statement of Proposition 11 assumes that the alphabet is binary. A tight bound is also known from Lemma 4 for unary finite languages. Next, we give a lower bound construction for the suffix distance neighbourhoods of finite languages.

Lemma 12. *Let $k = \lfloor \frac{n}{2} \rfloor$. Then there exists a DFA C_n with n states over a binary alphabet recognizing a finite language such that*

$$\mathrm{sc}(E(L(C_n), d_s, k)) \geq 2^k + k \cdot 2^{\lfloor \frac{n}{2} \rfloor} - 1.$$

Proof. Let $C_n = (Q_n, \{a, b\}, \delta_n, 0, \{n-1\})$, shown in Figure 3, with $Q_n = \{0, \ldots, n-1\}$ and the transition function δ_n is given by

- $\delta_n(i, a) = i + 1$ for $i = 0, \ldots, k-1, k+1, \ldots n-2$,
- $\delta_n(i, b) = i + 1$ for all $0 \leq i \leq n-2$.

The DFA C_n recognizes the subset of $\{a, b\}^{n-1}$ consisting of all words $w = ubv$ with $|w| = n-1$, $|v| = \lfloor \frac{n}{2} \rfloor$, and the $(k+1)$st symbol is b. On every state, A_n has a transition to the next state on both a and b except for state k where there is

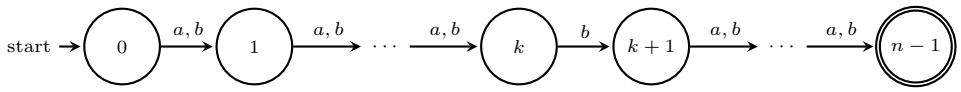

Fig. 3. The DFA C_n.

a transition to state $k+1$ only on b. We construct the DFA C'_n recognizing the neighbourhood by using the construction from Proposition 3.

First, we show that all the states are reachable. For the first k steps of the computation, we get a total of $\frac{|\Sigma|^k-1}{|\Sigma|-1} = 2^k - 1$ states by the same reasoning as in the proof of Lemma 5.

Now, we consider states (k, T) with $T \subseteq Q$, which are reachable on words of length k or greater. We can write $T = R \cup S$, where $R \subseteq \{0, \ldots, k\}$ and $S \subseteq \{k+1, \ldots, n-1\}$. Then the state $(k, \{0, \ldots, k\} \cup S)$ is reachable on the word $w_S = w_1 \cdots w_k$ with $w_i = b$ if $n - i \in T$ and $w_i = a$ if $n - i \notin T$. Since $k = \lfloor \frac{n}{2} \rfloor$, we have $2^{\lfloor \frac{n}{2} \rfloor}$ reachable states. From this, we see that we can reach the state $(k, \{i, \ldots, k\} \cup S)$ on the word $a^i w_S$. Then for $1 \leq i \leq k$, we get $k \cdot 2^{\lfloor \frac{n}{2} \rfloor}$ states.

Now we show that these states are pairwise distinguishable. Consider two distinct states (i, T) and (i', T'). Setting $i = i'$, we have $T \neq T'$ and there is an element $i_t \in T$ but $i_t \notin T'$. Then a final state is reachable from the state (i, T) on the word b^{n-i_t}, while no final state is reachable from (i, T') on the same word.

Next, set $i \neq i'$. Without loss of generality, let $i' > i$. Then $n-1$ is reachable from (i, T) on the word $b^{k-i}b^{n-k}$ but the same computation from (i', T') would reach $n-1$ before the automaton finished reading the word.

Thus, we have shown that there are $2^k + k \cdot 2^{\lfloor \frac{n}{2} \rfloor} - 1$ reachable states in C'_n and they are all pairwise distinguishable. □

We can summarize the results of Proposition 11 and Lemma 12 as follows:

Theorem 13. *Let L be a finite language recognized by an n-state DFA over an alphabet Σ with $|\Sigma| \geq 2$ and $k \leq n$. Then a DFA recognizing $E(L, d_s, k)$ requires at most $\frac{|\Sigma|^k-1}{|\Sigma|-1} + k \cdot 2^{\lfloor \frac{n}{2} \rfloor} - 1$ states. There is a family of DFAs with n states over a binary alphabet which reaches this bound when $k = \lfloor \frac{n}{2} \rfloor$.*

Next, we show that in the case $k > n$, the upper bound for the state complexity of suffix distance neighbourhoods for general regular languages can be reached by a neighbourhood of a finite language.

Theorem 14. *Let L be a finite language recognized by an n-state DFA over an alphabet Σ with $|\Sigma| \geq 2$ and $k > n$. Then a DFA recognizing $E(L, d_s, k)$ requires at most $(k-n) + 2^{n+1} - 2$ states. There is a family of DFAs with n states over an alphabet of size n which reaches this bound.*

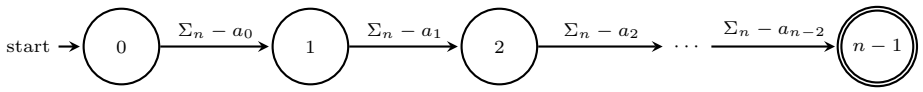

Fig. 4. The DFA D_n.

Proof. The upper bound follows from the upper bound for general regular languages given in Proposition 3.

Let $D_n = (Q_n, \Sigma_n, \delta_n, 0, \{0\})$, shown in Figure 4, with $\Sigma_n = \{a_0, a_1, \ldots, a_{n-1}\}$ and the transition function is defined by

$$\delta(i, a_j) = i + 1 \quad \text{for all } 0 \leq i < n-1,\ 0 \leq j \leq n-1, \text{ and } i \neq j.$$

We will show that D'_n, obtained by the construction of Proposition 3 is the minimal DFA for $E(L(D_n), d_s, k)$.

First, we show that all states of D'_n are reachable. States (i, Q) for $0 \leq i \leq k-n$ are reachable on the word a_{n-1}^i. Next, we show that for each $k-n < i \leq k$ we can reach 2^j states for each $i = k-n+j$. By a similar argument from the proof of Lemma 8, we note that for the state (i, P) we have $i = k-n+j$ and $P = Q \setminus R$ with $R \subseteq \{n-j+1, \ldots, n-1\}$. Writing $R = \{i_1, \ldots, i_m\}$ where $m < j$ and $i_1 > i_2 > \cdots > i_m$, we observe that (i, P) is reachable on the word $a_n^{k-n} a_{i_1-1} a_{i_2-1} \cdots a_{i_m-1}$. In this way, we can reach any subset R of j states on step i, giving us a total of $1 + 2 + \cdots + 2^n = 2^{n+1} - 1$ reachable states (i, P) with $k-n < i \leq k$.

In total, this gives us $(k-n) + 2^{n+1} - 1$ reachable states. Now, we will show that these states are pairwise inequivalent. Consider two distinct states (i, T) and (i', T'). Setting $i = i'$, we have $T \neq T'$ and there is an element $i_t \in T$ but $i_t \notin T'$. Then a final state is reachable from the state (i, T) by reading the word $a_{n-1}^{n-1-i_t}$ but no final state is reachable from the state (i, T') on the same word.

Next, set $i \neq i'$. Without loss of generality, let $i' > i$. Reading the word $a_{n-1}^{k-i+n-1}$ from (i, T) brings the automaton to the state (k, S) with $n-1 \in S$ so (k, S) is accepting. However, reading the same word from (i', T') causes the automaton to crash since $k+n-1$ is the length of the longest word in $E(L(D_n), d_s, k)$ and $i' + k - i + n - 1 > k + n - 1$.

Thus, we have shown that there are $(k-n) + 2^{n+1} - 2$ reachable states in A' and they are all pairwise distinguishable. □

Next, we consider the class of suffix-closed languages [2]. A language L is *suffix-closed* if $wx \in L$ implies $x \in L$. It is well known that the class of suffix-closed languages is a subclass of the regular languages. We will give a tight bound on the size of the DFA for neighbourhoods of suffix-closed languages with respect to the suffix distance.

Theorem 15. *Let L be a suffix-closed language recognized by an n-state DFA. Then*

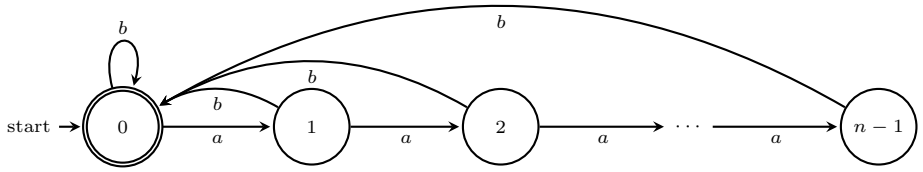

Fig. 5. The DFA E_n.

a DFA recognizing $E(L, d_s, k)$ requires at most $n + k + 1$ states. For each $n \in \mathbb{N}$ there exists an n-state DFA E_n recognizing a suffix-closed language such that the state complexity of $E(L(E_n), d_s, k)$ is $n + k + 1$ for all $k \in \mathbb{N}$.

Proof. First, we show that if L is suffix-closed, then $E(L, d_s, k) = \Sigma^{\leq k}L$. Consider a word w and let $v \in L$ be the closest word in L to w. We have $|w| \geq |v|$, since otherwise, we can find a suffix v' of v such that $|w| \geq |v'|$ which is closer to w than v. In this case, v must be a proper suffix of w. Suppose otherwise and v and w share a suffix, say v'. Again, w would be closer to v' than v. Thus, $w = uv$ and $d_s(w, v) = |u|$ and $E(L, d_s, k) = \Sigma^{\leq k}L$.

Brzozowski et al. [2] show that for two suffix-closed languages L_1 and L_2 with state complexity m_1 and m_2, respectively, the state complexity of L_1L_2 is $((m_1 + 1) - f)m_2 + f$, where f is the number of accepting states for the DFA recognizing L_1. Since the DFA recognizing L has n states and $\Sigma^{\leq k}$ has a DFA with $k+1$ states where all states are final, this implies that the state complexity of $E(L, d_s, k) = \Sigma^{\leq k} \cdot L$ has a DFA with at most $n + k + 1$ states.

To show that this bound is reachable, we consider the language L_n recognized by the DFA $E_n = (Q_n, \{a, b\}, \delta_n, q_0, F_n)$ shown in Figure 5. Consider two words $x = a^i$ and $y = a^j$ with $1 \leq i, j \leq n + k$ and we show that the states reached by these words are distinct. We choose $z = a^{n+k-1-j}b$ and observe that $xz \in E(L_n, d_s, k)$ and $yz \notin E(L_n, d_s, k)$. □

5. Conclusion

The state complexity of radius k prefix distance neighbourhoods of an n state DFA language depends linearly on n and on k [17]. As we have seen, the corresponding bounds for the suffix and the subword distance neighbourhoods depend exponentially on n and k and, furthermore, coming up with matching lower bounds is considerably more involved.

For suffix distance neighbourhoods where the radius k equals, roughly, half of the number of states n, we have given a matching lower bound construction based on a binary alphabet. However (and perhaps curiously), the construction does not seem to extend, at least not directly, for other values of the radius when $k < n$.

The precise state complexity of subword distance neighbourhoods remains open.

We do not have a lower bound construction matching the upper bound of Proposition 10 for the state complexity of subword distance neighbourhoods.

References

[1] Bruschi, D., Pighizzini, G.: String Distances and Intrusion Detection: Bridging the Gap Between Formal Languages and Computer Security. RAIRO Informatique Théorique et Applications **40** (2006) 303–313
[2] Brzozowski, J., Jirásková, G., Zou, C.: Quotient Complexity of Closed Languages. Theory of Computing Systems **54**(2) (2014) 277–292
[3] Calude, C.S., Salomaa, K., Yu, S.: Additive Distances and Quasi-Distances Between Words. Journal of Universal Computer Science **8**(2) (2002) 141–152
[4] Choffrut, C., Pighizzini, G.: Distances Between Languages and Reflexivity of Relations. Theoretical Computer Science **286**(1) (2002) 117–138
[5] Deza, M.M., Deza, E.: Encyclopedia of Distances. Springer Berlin Heidelberg (2009)
[6] Gao, Y., Moreira, N., Reis, R., Yu, S.: A Survey on Operational State Complexity. Journal of Automata, Languages and Combinatorics **21**(4) (2016) 251–310
[7] Han, Y.S., Ko, S.K., Salomaa, K.: The Edit-Distance Between a Regular Language and a Context-Free Language. International Journal of Foundations of Computer Science **24**(07) (2013) 1067–1082
[8] Holzer, M., Kutrib, M.: Descriptional and Computational Complexity of Finite Automata - A Survey. Information and Computation **209** (2011) 456–470
[9] Kari, L., Konstantinidis, S.: Descriptional Complexity of Error/Edit Systems. Journal of Automata, Languages and Combinatorics **9**(2/3) (2004) 293–309
[10] Kari, L., Konstantinidis, S., Kopecki, S., Yang, M.: An Efficient Algorithm for Computing the Edit Distance of a Regular Language via Input-Altering Transducers. CoRR abs/1406.1041 (2014)
[11] Konstantinidis, S.: Computing the Edit Distance of a Regular Language. Information and Computation **205**(9) (2007) 1307–1316
[12] Kutrib, M., Meckel, K., Wendlandt, M.: Parameterized Prefix Distance between Regular Languages. In: SOFSEM 2014: Theory and Practice of Computer Science, Lect. Notes Comput. Sci., Vol. 8327 (2014) 419–430
[13] Kutrib, M., Pighizzini, G.: Recent Trends in Descriptional Complexity of Formal Languages. Bulletin of the EATCS **111** (2013) 70–86
[14] Lothaire, M.: Applied Combinatorics on Words. Cambridge University Press (2005)
[15] Ng, T., Rappaport, D., Salomaa, K.: State Complexity of Neighbourhoods and Approximate Pattern Matching. International Journal of Foundations of Computer Science **29**(02) (2018) 315–329
[16] Ng, T., Rappaport, D., Salomaa, K.: Descriptional Complexity of Error Detection. In: Emergent Computation: A Festschrift for Selim G. Akl. Springer International Publishing (2017) 101–119
[17] Ng, T., Rappaport, D., Salomaa, K.: State Complexity of Prefix Distance. Theoretical Computer Science **679** (2017) 863–878
[18] Ng, T., Rappaport, D., Salomaa, K.: State Complexity of Prefix Distance of Subregular Languages. Journal of Automata, Languages and Combinatorics **22**(1-3) (2017) 169–188
[19] Povarov, G.: Descriptive Complexity of the Hamming Neighborhood of a Regular Language. In: LATA 2007. Proceedings of the 1st International Conference on Language and Automata Theory and Applications. (2007) 509–520
[20] Shallit, J.: A Second Course in Formal Languages and Automata Theory. Cambridge

University Press, Cambridge, MA (2009)
[21] Yu, S.: Regular Languages. In Rozenberg, G., Salomaa, A., eds.: Handbook of Formal Languages. Springer-Verlag, Berlin, Heidelberg (1997) 41–110