State Complexity of Suffix Distance

Timothy Ng, David Rappaport, Kai Salomaa

State Complexity of Suffix Distance

Timothy Ng, David Rappaport, and Kai Salomaa

School of Computing, Queen's University, Kingston, Ontario K7L 3N6, Canada
{ng, daver, ksalomaa}@cs.queensu.ca

Abstract. 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 DFA (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.

1 Introduction

Distances between strings and languages are used in many applications [4, 7, 10, 9]. 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 string into another. The prefix distance [3, 11] of strings 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 strings is defined analogously in terms of the longest common suffix (respectively, subword) of the strings.

Calude et al. [2] 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 [3].

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 [5, 6, 12]. For regularity preserving distances an important question 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 [14]. The descriptional complexity of error systems has been considered from a different point of view by Kari and Konstantinidis [8]. 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 [15, 16].

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 [18] or the survey by Yu [19]. A survey of distances is given by Deza and Deza [4].

In the following Σ is always a finite alphabet, the set of strings over Σ is Σ^* and ε is the empty string. 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 string $w \in \Sigma^*$ is a *subword* of x if there exist strings $u, v \in \Sigma^*$ 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 $Q \times \Sigma^* \to Q$ in the usual way. A DFA A is *complete* if δ is defined for all $q \in Q$ and $a \in \Sigma$.

A string $w \in \Sigma^*$ 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 string $w \in \Sigma^*$. 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 counts the number of symbols which do not belong to the longest common prefix of x and y [3]. 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 counts the number of symbols which do not belong to 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 [3]. However, "subword distance" has also been used for a distance defined in terms of the longest common noncontinuous subword [13].

It is known that neighbourhoods of regular languages with respect to the prefix, suffix and subword distance are always regular [3, 15]. 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 [15]. Optimal bounds for the size of an NFA recognizing a suffix distance, or subword distance, neighbourhood of a regular language are also known [15]. 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 1. *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 $k = \left\lfloor \frac{n}{2} \right\rfloor$ for an n state DFA.

Proposition 2. *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 this means that there exists a word x recognized by A such that $d(w, x) \leq k$. 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, when 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 machine begins by counting the first k steps of computation. On the ith step of the initial k steps, the machine 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 2 assumes that the cardinality of the alphabet is at least two. For suffix distance neighbourhoods of unary languages we have the following bounds. We note that in the unary case the suffix distance coincides with the prefix distance and leave the easy proof for the reader.

Lemma 1. *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 $\mathrm{sc}(E(L(A), d_s, k)) = n$.

For a constant size alphabet, the bound of Proposition 2 is significantly better than the bound implied by known results on nondeterministic state complexity in Proposition 1. Next we show that, at least for some values of the radius k, the bound of Proposition 2 is tight.

Lemma 2. *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. □

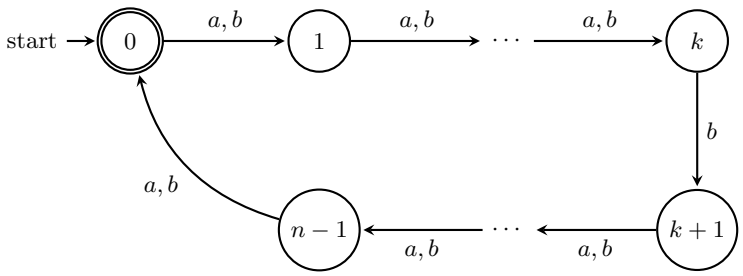

Fig. 1. The DFA A_n.

The following theorem then follows from Proposition 2 and Lemma 2.

Theorem 1. *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 3. *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 2 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 us $k-n$ states. Then on the following n steps, we proceed as in the rest of Proposition 2. 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, 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 us a total of at most $(k-n) + 2^{n+1} - 2$ states. □

We will show that the bound from Proposition 3 is reachable for a family of n state DFAs over an alphabet of size $n + 1$.

Lemma 3. *Let $k > n > 0$. Then there exists a DFA B_n with n states over an 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

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

□

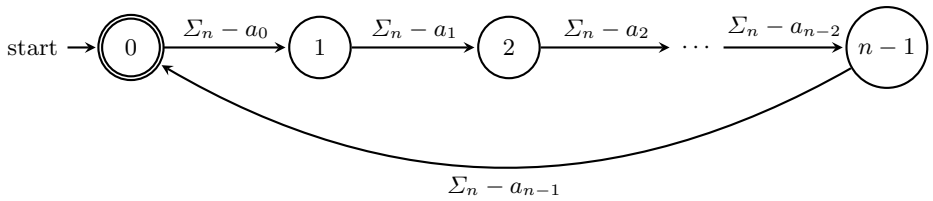

Fig. 2. The DFA B_n.

Proposition 3 and Lemma 3 can then be summarized in the following theorem.

Theorem 2. *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 1.

Proposition 4. *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.*

The bound of Proposition 4 is significantly better than the bound implied by nondeterministic state complexity [14] (in Proposition 1) 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 languages which belong to subregular language classes. First, we consider neighbourhoods of finite languages.

Proposition 5. *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 2. Observe that, as is the case for general regular languages, not all $(k+1) \cdot 2^n$ states that are defined 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 of the form (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 string of length i and is not reachable on any string of length $j > i$.

Recall that A recognizes a finite language and in each state (k, P) of A', the set P is a subset of states of A. First, we observe that the above property does not hold for subsets $P \subseteq Q$ in 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, why we can narrow our focus only to those states of the form (k, P).

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 states that are reachable, 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 string 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 5 assumes that the alphabet is binary. A tight bound is known from Lemma 1 also for finite languages.

Lemma 4. *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. We construct the DFA C'_n recognizing the neighbourhood by using the construction from Proposition 2. □

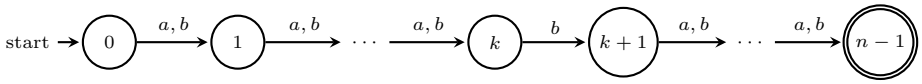

Fig. 3. The DFA C_n.

We can summarize the results of Proposition 5 and Lemma 4 as follows:

Theorem 3. *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 $k = \lfloor \frac{n}{2} \rfloor$.*

Now, we show that if $k > n$, the lower bound coincides with the upper bound for regular languages.

Theorem 4. *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.*

Proof. 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

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

□

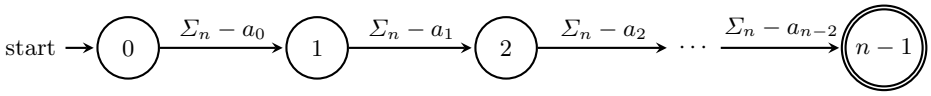

Fig. 4. The DFA D_n.

Next, we consider the class of suffix-closed languages [1]. 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 5. *Let L be a suffix-closed language recognized by an n-state DFA. Then 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 $k \in \mathbb{N}$.*

The DFA E_n is shown in Figure 5.

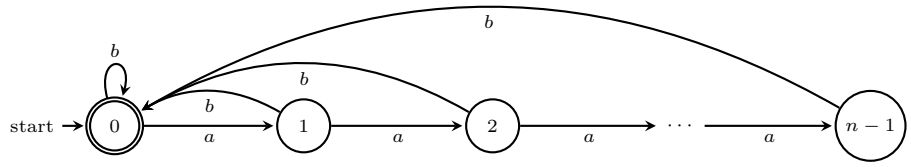

Fig. 5. The DFA E_n.

5 Conclusion

The state complexity of radius k prefix distance neighbourhoods of an n state DFA language depends linearly on n and on k [15]. 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 4 for the state complexity of subword distance neighbourhoods.

References

1. Brzozowski, J., Jirásková, G., Zou, C.: Quotient Complexity of Closed Languages. Theory Comput. Syst. **54**(2) (2014) 277-292
2. Calude, C.S., Salomaa, K., Yu, S.: Additive Distances and Quasi-Distances Between Words. Journal of Universal Computer Science **8**(2) (2002) 141–152
3. Choffrut, C., Pighizzini, G.: Distances between languages and reflexivity of relations. Theoretical Computer Science **286**(1) (2002) 117–138
4. Deza, M.M., Deza, E.: Encyclopedia of Distances. Springer Berlin Heidelberg (2009)
5. Gao, Y., Moreira, N., Reis, R., Yu, S.: A survey on operational state complexity. `arXiv:1509.03254v1 [cs.FL]`, Sept. 2015. To appear in *Computer Science Review.*
6. Holzer, M. Kutrib, M.: Descriptional and computational complexity of finite automata — A survey. *Inform. Comput.* **209** (2011) 456–470.
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** (2013) 1067–1082
8. Kari, L., Konstantinidis, S.: Descriptional complexity of error/edit systems. Journal of Automata, Languages, and Combinatorics **9** (2004) 293–309
9. 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)
10. Konstantinidis, S.: Computing the edit distance of a regular language. Information and Computation **205** (2007) 1307–1316
11. 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. **8327** (2014), Springer, 419–430
12. Kutrib, M., Pighizzini, G.: Recent trends in descriptional complexity of formal languages. *Bulletin of the EATCS* 111 (2013) 70–86.
13. Lothaire, M.: Applied Combinatorics on Words, Ch. 1 Algorithms on Words. Encyclopedia of Mathematics and It's Applications 105, Cambridge University Press, New York, 2005
14. Ng, T., Rappaport, D., Salomaa, K.: State complexity of neighbourhoods and approximate pattern matching. In: Developments in Language Theory, DLT 2015, Liverpool, UK, July 27–30, Lect. Notes Comput. Sci. 9168 (2015) 389–400
15. Ng, T., Rappaport, D., Salomaa, K.: State complexity of prefix distance. In: Implementation and Application of Automata, CIAA 2015, Lect. Notes Comput. Sci. 9223 (2015) 238–249
16. Ng, T., Rappaport, D., Salomaa, K.: State complexity of prefix distance of subregular languages. In: Descriptional Complexity of Formal Systems, DCFS 2016, Bucharest, Romania, July 6–8, 2016, lect. Notes Comput. Sci. 9777 (2016) 192–204
17. Salomaa, K., Yu, S.: NFA to DFA Transformation for Finite Languages. In: Automata Implementation, WIA '96, Lect. Notes Comput. Sci. 1260 (1997) pp. 149-158
18. Shallit, J.: A second course in formal languages and automata theory. Cambridge University Press, Cambridge, MA (2009)
19. Yu, S.: Regular languages. In Rozenberg, G., Salomaa, A., eds.: Handbook of Formal Languages. Springer-Verlag, Berlin, Heidelberg (1997) 41–110