TUCS

Alexander Okhotin

# On the state complexity of scattered substrings and superstrings

Turku Centre *for* Computer Science

TUCS Technical Report
No 849, October 2007

# On the state complexity of scattered substrings and superstrings

Alexander Okhotin
Department of Mathematics, University of Turku, *and*
Turku Centre for Computer Science
Turku FIN–20014, Finland, *and*
Academy of Finland
alexander.okhotin@utu.fi

TUCS Technical Report
No 849, October 2007

# Abstract

It is proved that the set of *scattered substrings* of a language recognized by an $n$-state DFA requires at least $2^{n/2-2}$ states (the known upper bound is $2^n$), with witness languages given over an exponentially growing alphabet. For a 3-letter alphabet, scattered substrings are shown to require at least $2^{\sqrt{2n}-6}$ states. A similar state complexity function for *scattered superstrings* is shown to be $2^{n-2}+1$ for an alphabet of at least $n-2$ letters, and strictly less for any smaller alphabet. For a 3-letter alphabet, the state complexity of scattered superstrings is at least $\frac{1}{5}4^{\sqrt{\frac{n}{2}}}n^{-\frac{3}{4}}$.

**Keywords:** descriptional complexity, finite automata, state complexity, substring, subword, subsequence, Higman–Haines sets

**TUCS Laboratory**
Discrete Mathematics for Information Technology

# 1 Introduction

This paper contributes to the active research on the state complexity of operations on deterministic finite automata [1, 2, 5, 6, 7, 8, 9, 10, 11] by investigating the following operations. Let $L$ be a language over the alphabet $\Sigma^*$. The language of *scattered substrings* of $L$ is defined as

$$f_1(L) = \{a_1 \dots a_n \mid x_0 a_1 x_1 \dots a_n x_n \in L \text{ for some } x_i \in \Sigma^*\}.$$

The language of *scattered superstrings* of $L$ is

$$f_2(L) = \{x_0 a_1 x_1 \dots a_n x_n \mid x_i \in \Sigma^* \text{ and } a_1 \dots a_n \in L\}.$$

The state complexity of these operations with respect to nondeterministic automata (NFA) has recently been determined by Gruber, Holzer and Kutrib [4], who established that for a language $L$ given by an $n$-state NFA, $n$ states in an NFA are necessary and sufficient to represent $f_1(L)$ or $f_2(L)$.

If $L$ is recognized by an $n$-state DFA, then a DFA with $2^n$ states is clearly sufficient to represent $f_1(L)$ and $f_2(L)$. The question is, whether this number of states is necessary?

For a related operation of taking ordinary (contiguous) substrings, defined as

$$f_3(L) = \{w \mid xwy \in L \text{ for some } x, y \in \Sigma^*\},$$

it was established by Shallit [11] that $2^{n-1}$ states are sufficient to represent $f_3(L)$ for all $n$-state languages $L$, and that this bound is tight already for a 2-letter alphabet.

The state complexity of scattered substrings has recently been studied by Gruber, Holzer and Kutrib [4], who have shown, in particular, that the set of scattered substrings of an $n$-state language over a $\sqrt{n}$-letter alphabet requires at least $2^{\Theta(\sqrt{n}\log_2 n)}$ states. Two related results are obtained in the present paper. First, for an exponentially growing alphabet it is proved that the set of scattered substrings of an $n$-state language requires at least $2^{\frac{n}{2}-2}$ states, which yields a $2^{\Theta(n)}$ estimation of the state complexity of this operation. Second, it is shown that for a fixed 3-letter alphabet, the state complexity is at least $2^{\sqrt{2n+30}-6}$.

The cited paper by Gruber, Holzer and Kutrib [4] also mentions the state complexity of scattered superstrings: it is stated that, again, for a $\sqrt{n}$-letter alphabet the state complexity is at least $2^{\Theta(\sqrt{n}\log_2 n)}$. This estimation is improved in the present paper: it is established that $2^{n-2}+1$ states are sufficient to represent the set of scattered substrings of any $n$-state language, and that this number of states is necessary for every alphabet of at least $n-2$ letters. Furthermore, it is proved that if the number of letters is reduced, then this upper bound cannot be reached. At the same time, for a fixed 3-letter alphabet it is shown that this operation requires at least $C_{\lfloor\sqrt{\frac{n}{2}}\rfloor}$ states, where $C_k$ is the $k$-th Catalan number; this is at least $\frac{1}{5}4^{\sqrt{\frac{n}{2}}}n^{-\frac{3}{4}}$.

## 2 Definitions

A *deterministic finite automaton* (DFA) is a quintuple $(\Sigma, Q, \delta, q_0, F)$, in which $\Sigma$ is an input alphabet, $Q$ is a finite set of states, $\delta : Q \times \Sigma \to Q$ is a total transition function, $q_0 \in Q$ is the initial state, and $F \subseteq Q$ is the set of accepting states. A state of a DFA is called *dead* if no strings are accepted starting from it.

A *nondeterministic finite automaton* (NFA) is a quintuple $(\Sigma, Q, \delta, Q_0, F)$ with a set of initial states $Q_0 \subseteq Q$ and with a nondeterministic transition function $\delta : Q \times \Sigma \to 2^Q$. Any NFA can be converted to an equivalent DFA with the set of states $2^Q$; this transformation is known as the *subset construction.*

The *state complexity* of a regular language $L$, denoted $sc(L)$, is the least number of states in any DFA accepting $L$.

Consider a $k$-ary operation on languages $f : (2^{\Sigma^*})^k \to 2^{\Sigma^*}$ that preserves regularity in the sense that for all regular $L_1, \ldots, L_k$ the language $f(L_1, \ldots, L_k)$ is regular as well. Define the state complexity function of $f$ as $sc_f : \mathbb{N}^k \to \mathbb{N}$, so that $sc_f(n_1, \ldots, n_k)$ equals the greatest value of $sc(f(L_1, \ldots, L_k))$ over all vectors of languages $(L_1, \ldots, L_k)$ with $sc(L_i) = n_i$ for all $i$.

## 3 Scattered substrings

The set of scattered substrings of a given NFA can be recognized by an NFA with the same number of states. Essentially the following construction has been given by Gruber et al. [3]:

**Lemma 1.** *Let $A = (\Sigma, Q, q_0, \delta, F)$ be a DFA and consider an NFA $B = \big(\Sigma, Q, \{\delta(q_0, u) \mid u \in \Sigma^*\}, \delta', F'\big)$, where $\delta'(q, a) = \{\delta(q, au) \mid u \in \Sigma^*\}$ and $F' = \{q' \mid \exists u \in \Sigma^* : \delta(q', u) \in F\}$. Then $B$ recognizes the set of scattered substrings of $L(A)$.*

The strings $u$ in the definition correspond to the contiguous substrings erased from $w \in L(A)$ to obtain a scattered substring of $w$.

This, in particular, gives a $2^n$ upper bound on the state complexity of this operation. This upper bound can be closely approached by a lower bound using a growing alphabet of an exponential size:

**Lemma 2.** *Let $A_k$ be a $(2k + 2)$-state DFA over the $(2^k + 1)$-symbol alphabet $\{\ a_X \ \big|\ X \subseteq \{0, 1, \ldots, k - 1\}\ \} \cup \{c\}$ and with the set of states $\{q_0, \ldots, q_{k-1}, r_0, \ldots, r_{k-1}, q_{acc}, q_{dead}\}$, of which $q_0$ is the initial state and $q_{acc}$ is the sole accepting state. The transitions of $A_k$ are defined as follows (un-*

*defined transitions go to $q_{dead}$):*

$$\delta(q_i, a_X) = \begin{cases} r_i, & \text{if } i \in X \\ \varnothing, & \text{otherwise} \end{cases}$$
$$\delta(q_i, c) = q_{i+1 \pmod k}$$
$$\delta(r_i, a_{\{i\}}) = q_{acc}$$

*Then every DFA for the language of all scattered substrings of $L(A_k)$ requires at least $2^k$ states.*

The automaton $A_3$ over the alphabet $\{a_\varnothing, a_{\{0\}}, a_{\{1\}}, a_{\{2\}}, a_{\{0,1\}}, a_{\{0,2\}}, a_{\{1,2\}}, a_{\{0,1,2\}}, c\}$ is given in Figure 1.

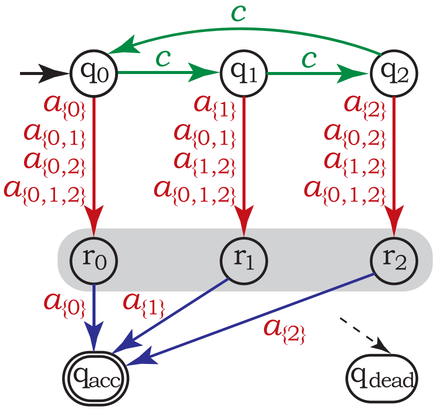

Figure 1: DFA $A_k$ from Lemma 2, for $k = 3$; undefined transitions go to $q_{dead}$.

The strongly connected component $\{q_0, \dots, q_{k-1}\}$ constitutes a "switchbox", while the states $r_0, \dots, r_{k-1}$ are "memory cells". The idea of the construction is that a transition by $a_X$ sets all cells corresponding to the numbers in $X$ (or, to be more precise, clears all cells corresponding to the numbers not in $X$), and then the symbols $a_{\{i\}}$ are used to probe the values in these cells.

*Proof.* Consider the NFA constructed for the set of scattered substrings of $L(A_k)$ according to Lemma 1. The corresponding DFA obtained by the subset construction has the set of all states as the initial subset. By each symbol $a_X$, the DFA goes to the subset $S_X = \{r_i \mid i \in X\} \cup S'_X$, where $S'_X \subseteq \{q_{acc}, q_{dead}\}$.

Consider any two subsets of this form. If $S_X$ and $S_Y$ with $X \neq Y$, then there exists $i \in X \,\Delta\, Y$. Assume, without loss of generality, that $i \in X$ and $i \notin Y$. Then $r_i \in S_X$ and $r_i \notin S_Y$, and the one-symbol string $a_{\{i\}}$ is accepted from $S_X$ and is not accepted from $S_Y$.

Hence, $2^k$ reachable and pairwise inequivalent subsets have been constructed, which establishes the lemma. □

At the same time, a superpolynomial upper bound can be achieved using a fixed 3-letter alphabet:

**Lemma 3.** *Let $k \geqslant 1$ and let $k' = 2^{\lceil \log_2 k \rceil}$ be $k$ rounded up to the next power of 2. Let $A'_k$ be a DFA over $\Sigma = \{a, b, c\}$ with the set of states $Q \cup R \cup P \cup \{q_{dead}\}$, where*

$$\begin{aligned} Q &= \{q_i \mid 0 \leqslant i \leqslant k-1\} \cup \{q'_i \mid 0 \leqslant i \leqslant k-2\}, \\ R &= \{r_{i,j} \mid 0 \leqslant i < j \leqslant k-1\} \\ P &= \{p_i \mid 1 \leqslant i < k' + k\}, \end{aligned}$$

*of which $q_0$ is the initial state and $p_1$ is the sole accepting state. The transitions of $A'_k$ are defined as follows (with the rest of transitions going to $q_{dead}$):*

$$\begin{aligned} \delta(q_i, c) &= q'_i && (0 \leqslant i \leqslant k-2) \\ \delta(q'_i, c) &= q_i && (0 \leqslant i \leqslant k-2) \\ \delta(q_i, a) &= r_{i,i+1} && (0 \leqslant i \leqslant k-1) \\ \delta(q'_i, s) &= q_{i+1} && (0 \leqslant i \leqslant k-2,\ s \in \{a, b\}) \\ \delta(r_{i,j}, s) &= r_{i,j+1} && (0 \leqslant i < j \leqslant k-1,\ s \in \{a, b\}) \\ \delta(r_{i,k-1}, s) &= p_{k'+i} && (0 \leqslant i \leqslant k-2,\ s \in \{a, b\}) \\ \delta(q_{k-1}, a) &= p_{k'+k-1} && \\ \delta(p_{2i}, a) &= p_i && (1 < 2i < k' + k) \\ \delta(p_{2i+1}, b) &= p_i && (1 < 2i + 1 < k' + k) \end{aligned}$$

*Then $A'_k$ contains at most $\frac{k^2+9k-4}{2}$ states, while every DFA for the language of all scattered substrings of $L(A'_k)$ requires at least $2^k$ states.*

These automata are illustrated in Figure 2. This construction simulates the operation of the automata from Lemma 2 by using long "wires" to transfer values (instead of an instant transfer by the means of exponentially many symbols). Now each strongly connected component $\{q_i, q'_i\}$ is a "switch-box", while $p_{k'+i}$ is the corresponding "memory cell". The value of every $i$-th cell is determined in the $i$-th switch-box, and for different cells this is done at a different time. The determined values move towards the memory cells along the wires $r_{i,i+1}, \ldots r_{i,k-1}$, and the values reach the cells synchronously.

The states from $P$ represent wires of equal length used to probe the values of the memory cells. The transitions between these states form a binary tree, so that from each state $p_i \in P$ the automaton $A'_k$ accepts a unique string of length $\log_2 k'$ representing the binary notation of the number $i$.

*Proof.* Consider the number of states in $A'_k$: obviously, $|Q| = 2k - 1$ and $|R| = \frac{k(k-1)}{2}$. The number $k' = 2^{\lceil \log_2 k \rceil}$ equals at most $2k - 2$, hence $|P| \leqslant 3k - 2$. With the addition of the dead state, the total number of states is at most $5k - 2 + \frac{k(k-1)}{2} = \frac{k^2+9k-4}{2}$.

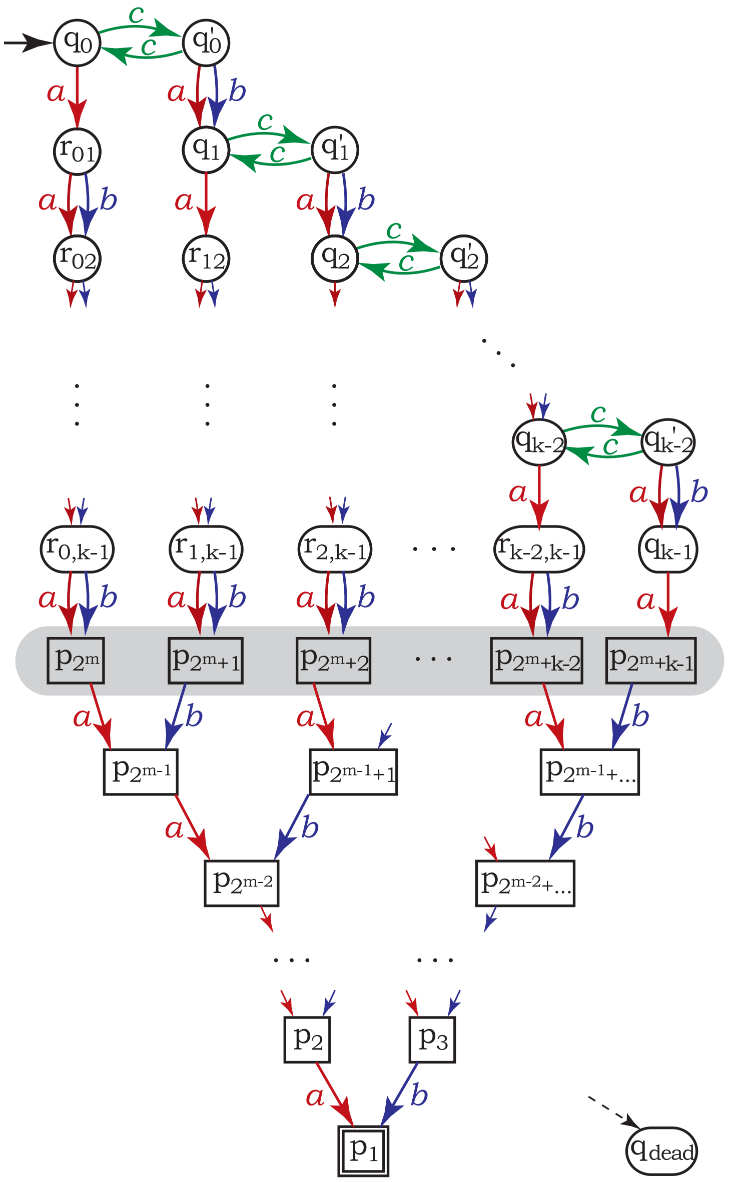

Figure 2: DFA $A'_k$ from Lemma 3; undefined transitions go to $q_{dead}$.

As in the proof of the previous lemma, consider the NFA for the set of scattered substrings of $L(A'_k)$ given in Lemma 1. Construct an equivalent DFA using the subset construction. Its initial subset is the set of all states.

Consider any string $s_1 \ldots s_k$ with $s_i \in \{a, b\}$. Let $S_{s_1 \ldots s_k}$ be the subset reached from the initial subset by this string. Then

$$S_{s_1 \ldots s_k} = \{p_{k'+i} \mid s_{i+1} = a\} \cup S'_{s_1 \ldots s_k},$$

for some irrelevant set $S'_{s_1 \ldots s_k} \subseteq \{p_1, \ldots, p_{k'-1}\}$.

Now consider the subsets $S_{s_1 \ldots s_k}$ and $S_{s'_1 \ldots s'_k}$ corresponding to distinct strings $s_1 \ldots s_k$ and $s'_1 \ldots s'_k$, with $s_m, s'_m \in \{a, b\}$. Then there exists some $(m+1)$-th position, such that $s_{m+1} \neq s'_{m+1}$. Assume, without loss of generality, that $s_{m+1} = a$ and $s'_{m+1} = b$. Then $p_{k'+m} \in S_{s_1 \ldots s_k}$ and $p_{k'+m} \notin S_{s'_1 \ldots s'_k}$.

Consider a string $w_m = s_1 \ldots s_{\log_2 k'}$, where each $i$-th symbol is defined as $a$ if the $i$-th digit in the binary representation of $m$ is 0, and $s_i = b$ if this digit is 1. This string is accepted from $p_{k'+m}$, since the automaton can read it passing through the states $p_{\lfloor (k'+m)/2^i \rfloor}$ for $i = 1, 2, \ldots, \log_2 k'$, where the last state is $p_1$. However, for any other state $p_{k'+\ell}$ with $0 \leqslant \ell < k$ and $\ell \neq m$, the unique path of length $\log_2 k'$ from $p_{k'+\ell}$ to $p_1$ has labels forming a string different from $w_m$, and hence the string $w_m$ is not accepted from $p_{k'+\ell}$.

It has thus been shown that the DFA contains $2^k$ reachable and pairwise inequivalent subsets corresponding to different binary strings of length $k$, which proves the lemma. □

Altogether the following results have been obtained:

**Theorem 1.** *State complexity of taking scattered substrings is at least* $2^{\frac{n}{2}-2}$ *(over an unbounded alphabet) and at most* $2^n$*. For a 3-letter alphabet, it is at least* $2^{\sqrt{2n+30}-6}$.

*Proof.* The upper bound of $2^n$ is due to Gruber et al. [3] and it is stated in Lemma 1.

For every $n \geqslant 1$, Lemma 2 defines an automaton $A_k$ with $k = \lfloor \frac{n}{2} \rfloor - 1$, which contains at most $n$ states. It is stated that the set of scattered substrings of $L(A_k)$ requires at least $2^k = 2^{\lfloor \frac{n}{2} \rfloor - 1} \geqslant 2^{\frac{n}{2}-2}$ states.

For a 3-letter alphabet, note that the DFA $A'_k$ constructed in Lemma 3 contains at most $\frac{k^2+9k-4}{2} \leqslant \frac{(k+5)^2-k-29}{2} \leqslant \frac{(k+5)^2-30}{2}$ states, and hence for every number $n$ the automaton $A'_k$ with $k = \lfloor \sqrt{2n+30} \rfloor - 5$ contains at most $n$ states. By Lemma 3, the set of scattered substrings of $L(A'_k)$ requires at least $2^k = 2^{\lfloor \sqrt{2n+30} \rfloor - 5} \geqslant 2^{\sqrt{2n+30}-6}$ states. □

It remains open whether a lower bound of $2^{\Theta(n)}$ can be established using a fixed alphabet. Also, the given lower bound for an unbounded alphabet still leaves room for improvement: it is an open question whether $\Theta(2^n)$ states are necessary to represent the set of scattered substrings.

# 4 Scattered superstrings

A string $x \in \Sigma^*$ is a scattered superstring of a string $w \in \Sigma^*$ if $w = w_1 \dots w_k$ and $x \in \Sigma^* w_1 \Sigma^* \dots w_k \Sigma^*$. It is known that the set of scattered substrings of a language generated by an $n$-state NFA can be recognized by another $n$-state NFA. The following construction is adapted from Gruber et al. [3]:

**Lemma 4.** *Let $A = (\Sigma, Q, q_0, \delta, F)$ be a DFA. Then the NFA $B = (\Sigma, Q, \{q_0\}, \delta', F)$, where $\delta'(q, a) = \{\delta(q, a), q\}$, recognizes the set of scattered superstrings of $L(A)$.*

This gives an $2^n$ upper bound on the state complexity of this operation. The actual state complexity is lower, since some of the subsets are unreachable and some are equivalent:

**Lemma 5.** *Consider the NFA $B$ with states $\{0, \dots, n-1\}$ constructed in Lemma 4, and consider the DFA obtained out of $B$ using the subset construction. Then*

1. *If a subset $Y$ is reachable from a subset $X$, then $X \subseteq Y$. In particular, every reachable subset contains state 0.*

2. *All subsets containing an accepting state are equivalent.*

*Proof.* Let $\delta''$ be the transition function of the DFA. It is sufficient to show that $X \subseteq \delta''(X, a)$ for every $a \in \Sigma$. Indeed, for every $i \in X$, since $i \in \delta'(i, a)$ for every $a \in \Sigma$, there holds $i \in \delta''(i, X)$.

Since the initial subset is $\{0\}$, all reachable subsets contain 0.

Suppose a subset $X$ contains an accepting state $i \in F$. Then $i \in \delta''(X, w)$ for every $w \in \Sigma^*$, that is, every string is accepted from $X$. This makes all such subsets equivalent. □

Therefore, at most $2^{n-1}$ subsets are reachable. If $k$ is the number of accepting states, then at most $2^{n-1-k}$ subsets containing 0 and not containing any accepting states can be pairwise inequivalent, while the rest of the subsets can always be merged into a single equivalence class. Setting $k = 1$, this gives an upper bound of $2^{n-2} + 1$ states in a DFA recognizing the set of scattered superstrings. This bound is actually precise, which can be shown using a linearly growing alphabet:

**Lemma 6.** *Let $A_n$ be a DFA over $\Sigma = \{a_1, \dots, a_{n-2}\}$, with the set of states $\{0, \dots, n-1\}$, of which 0 is the initial state and $n-1$ is the sole accepting state, and with transitions $\delta(0, a_i) = i$ and $\delta(i, a_i) = n-1$; the rest of the transitions are defined as $\delta(i, a_j) = i$. This DFA is depicted in Figure 3. Then every DFA for the language of all scattered superstrings of $L(A_n)$ requires $2^{n-2} + 1$ states.*

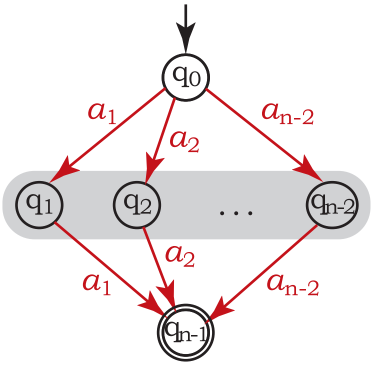

Figure 3: DFA $A_n$ from Lemma 6; undefined transitions are self-loops.

*Proof.* Consider the $n$-state NFA for the language of scattered superstrings of $L(A_n)$ and the DFA obtained out of it by the subset construction.

Let us first show that for every $X \subseteq \{1, \dots, n-2\}$, the subset $\{0\} \cup X$ is reachable. Let $X = \{i_1, \dots i_k\}$, with $k \geqslant 0$ and $1 \leqslant i_j \leqslant n-2$. Then $\{0\} \cup X$ is reachable from the initial subset $\{0\}$ by the string $a_{i_1} \dots a_{i_k}$. In addition, the set of all states is reachable from $\{0, 1, \dots, n-2\}$ by $a_1$.

Having presented $2^{n-2}+1$ reachable subsets, it remains to show their pairwise inequivalence. The set of all states is the only subset from which $\varepsilon$ is accepted. For every two states $\{0\} \cup X$ and $\{0\} \cup Y$, with $X, Y \subseteq \{1, \dots, n-2\}$ and $X \neq Y$, let $i \in X \,\Delta\, Y$, and assume without loss of generality that $i \in X$ and $i \notin Y$. Then the string $a_i$ is accepted from $X$ and is not accepted from $Y$. This completes the proof. □

Lemma 6 uses an alphabet of size $n-2$ to establish the precise value of the state complexity. It is natural to ask whether this bound could be established using a smaller alphabet, such as an alphabet of a fixed size or a slower growing alphabet. The answer is negative: $n-2$ symbols are necessary to reach all $2^{n-2}+1$ states.

**Lemma 7.** *Let $A$ be any $n$-state DFA over an $(n-3)$-symbol alphabet. Then the set of scattered superstrings of $L(A)$ can be represented using strictly less than $2^{n-2}+1$ states.*

*Proof.* Let $\{0, 1, \dots, n-1\}$ be the set of states of $A$, with 0 as the initial state. If $A$ has multiple accepting states, then the statement follows from Lemma 5, so assume $A$ has a unique accepting state $n-1$.

Consider the NFA for this language given in Lemma 4, and then the corresponding subset DFA. Its initial subset is $\{0\}$. Due to the monotonicity property given by Lemma 5, if a subset of the form $\{0, i\}$, with $1 \leqslant i \leqslant n-2$, is reachable, it must be reachable by a direct transition from $\{0\}$. Since there are $n-2$ such subsets but only $n-3$ symbols, at least one of these subsets remains unreachable. The number of states in the minimal DFA for this language is accordingly less than the upper bound $2^{n-2}+1$. □

On the other hand, like in the case of scattered substrings, a fixed 3-letter alphabet is sufficient to obtain a relatively high lower bound.

**Lemma 8.** *Let $k \geqslant 1$ and define $k' = 2^{\lceil \log_2 \frac{k^2-k}{2} \rceil}$. Define a DFA $A'_k$ over $\Sigma = \{a, b, c\}$ with the set of states $Q \cup R \cup P$, where*

$$\begin{aligned} Q &= \{q_i \mid 0 \leqslant i \leqslant k-2\} \\ R &= \{r_{i,j} \mid 0 \leqslant i \leqslant k-2,\ 1 \leqslant j \leqslant k-i-1\} \\ P &= \{p_i \mid 1 \leqslant i < k' + |R|\}, \end{aligned}$$

*of which $q_0$ is the initial state and $p_1$ is the sole accepting state. Let $\pi : R \to \{p_i \mid k' \leqslant i < k' + |R|\}$ be any bijective mapping and define the transitions of $A'_k$ as follows (undefined transitions are assumed to be self-loops):*

$$\begin{aligned} \delta(q_i, a) &= r_{i,1} && (0 \leqslant i \leqslant k-2) \\ \delta(q_i, b) &= q_{i+1} && (0 \leqslant i < k-2) \\ \delta(r_{i,j}) &= r_{i,j+1} && (0 \leqslant i \leqslant k-2, 1 \leqslant j < k-i-1) \\ \delta(r_{ij}, c) &= \pi(r_{ij}) && (r_{ij} \in R) \\ \delta(p_{2i}, a) &= p_i && (1 < i < 2k') \\ \delta(p_{2i+1}, b) &= p_i && (1 < i < 2k') \end{aligned}$$

*Then $A'_k$ contains at most $2k^2-k-3$ states, while every DFA for the language of all scattered superstrings of $L(A'_k)$ requires at least $C_k$ states, where $C_k = \frac{(2k)!}{k!(k+1)!}$ is the $k$-th Catalan number.*

An automaton of this form is shown in Figure 4. Consider its upper part formed by the states from $Q$ and $R$. Each column can be regarded as a counter, and if state $q_i$ is reached, let us say that the $i$-th counter has been activated. Initially, only counter 0 is activated. A transition by $b$ activates one more counter. A transition by $a$ adds 1 to the values of all currently active counters. By using a different number of $a$s between $b$s, one can assign different combinations of values to the counters.

The values of the counters are probed in the lower part of the automaton, which is exactly the same as in Lemma 3. The transitions between states in $P$ form a binary tree, and a unique string of length $\log_2 k'$ is accepted from each state $p_i \in P$; this string represents the binary notation of the number $i$.

*Proof.* Note that $k'$ equals $|R|$ rounded up to the next power of two. Then $k' \leqslant k^2 - k - 2$, and the number of states in $A'_k$ is at most $(k-1) + \frac{k^2-k}{2} + (k^2 - k - 2 + \frac{k^2-k}{2}) = 2k^2 - k - 3$.

As in all lower bound proofs in this paper, consider the NFA for the set of scattered superstrings of $L(A'_k)$ (as in Figure 4, with the addition of the remaining self-loops) and the DFA obtained from it by the subset construction. Its initial subset is $\{q_0\}$. The first step is to construct a family

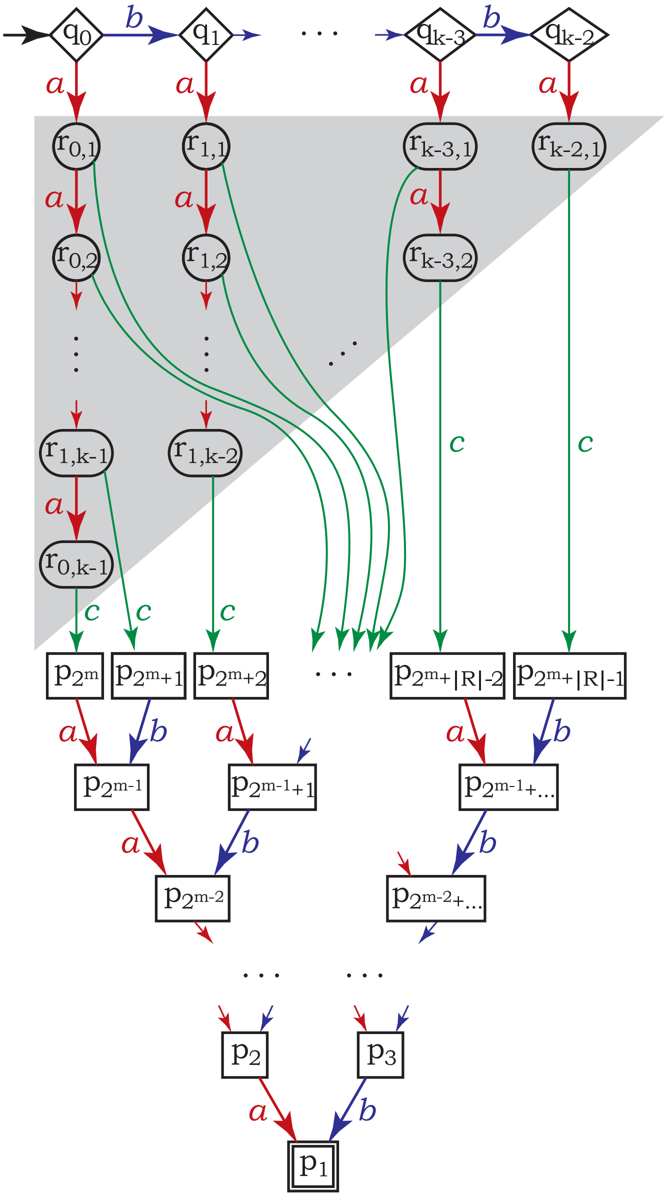

Figure 4: DFA $A'_k$ from Lemma 8; undefined transitions are self-loops.

of reachable subsets corresponding to vectors of integers $(i_0, \ldots, i_{k-1})$, with $k-1 \geqslant i_0 \geqslant i_1 \geqslant \ldots \geqslant i_{k-2} \geqslant i_{k-1} = 0$ and with $i_j \leqslant k-j-1$. It is well-known that there are exactly $C_k$ such vectors [12].

Define the subset $S_{i_0 \ldots i_{k-1}}$ corresponding to a vector $(i_0, \ldots, i_{k-1})$ as the set of states reached from the initial state by the string $a^{i_0-i_1}ba^{i_1-i_2}b \ldots ba^{i_{k-2}-i_{k-3}}ba^{i_{k-2}-i_{k-1}}$. It is easy to see that

$$S_{i_0 \ldots i_{k-1}} = Q \cup \{r_{j,\ell} \mid 1 \leqslant \ell \leqslant i_j\} \subseteq Q \cup R$$

Let $(i_0, \ldots, i_{k-1})$ and $(i'_0, \ldots, i'_{k-1})$ be any two different vectors of this form. Then there exists a number $j_0$, such that $i_{j_0} \neq i'_{j_0}$ and $i_j = i'_j$ for all $j$ with $j_0 < j \leqslant k-2$. Therefore the state $r_{j_0, \max(i_{j_0}, i'_{j_0})}$ is contained either in $S_{i_0 \ldots i_{k-1}}$ or in $S_{i'_0 \ldots i'_{k-1}}$, but not in both.

This proves that the subsets $\{S_{i_0 \ldots i_{k-1}}\}$ are pairwise distinct. In order to show that these subsets are pairwise inequivalent, it is sufficient to prove that for every state $r \in R$ there exists a string accepted by the NFA from $r$, but from no other state in $R$. Let $\pi(r) = p_m$. Then this string is defined as $w_r = cs_1 \ldots s_{\log_2 k'}$, where, for each $i$, $s_i = a$ if the $i$-th digit in the binary representation of $m$ is 0, and $s_i = b$ if this digit is 1. The automaton can read this string, passing through the states $p_{\lfloor m/2^i \rfloor}$ for $i = 1, 2, \ldots, \log_2 k'$, where the last state is $p_1$. For any other state $r' \in R$, the unique path of length $\log_2 k' + 1$ from $r'$ to $p_1$ has labels forming a string different from $w_r$, and hence the string $w_r$ is not accepted from $r'$.

This shows that the minimal DFA for the language of scattered substrings of $L(A'_k)$ must have at least $C_k$ states corresponding to the constructed subsets. □

It remains to represent the given lower bound for a $(2k^2-k-3)$-state DFA in the form of $f(n)$ states for an $n$-state DFA. Using $n$ states, one can represent a DFA $A'_k$ from Lemma 8, with

$$k = \left\lfloor \sqrt{\frac{n}{2} + \frac{25}{16}} + \frac{1}{4} \right\rfloor \geqslant \sqrt{\frac{n}{2}} - 1.$$

For the final step, it is convenient to use the following lower bound on Catalan numbers (a proof is included in the appendix):

**Proposition 1.** *For every $k \geqslant 1$, $C_k > \frac{4^k}{(k+1)^{3/2}\sqrt{\pi}}$.*

Then the state complexity of scattered superstrings must be at least

$$\frac{4^{\sqrt{\frac{n}{2}}-1}}{(\sqrt{\frac{n}{2}})^{3/2}\sqrt{\pi}} = \frac{2^{\frac{3}{4}}}{4\sqrt{\pi}} \frac{4^{\sqrt{\frac{n}{2}}}}{n^{\frac{3}{4}}} \geqslant \frac{1}{5} 4^{\sqrt{\frac{n}{2}}} n^{-\frac{3}{4}}.$$

All results on scattered substrings obtained in this section are put together in the following theorem:

**Theorem 2.** *State complexity of taking scattered superstrings is exactly* $2^{n-2}+1$. *The bound is reached for a growing* $(n-2)$*-letter alphabet. For alphabets of* $n-3$ *symbols or fewer this bound is not reached. For a 3-letter alphabet the state complexity is at least* $\frac{1}{5}4^{\sqrt{n/2}}n^{-\frac{3}{4}}$.

It remains unknown whether the state complexity of scattered superstrings for a fixed alphabet is $2^{\Theta(n)}$ or $2^{o(n)}$.

# Acknowledgements

I am grateful to Géraud Sénizergues for pointing out these problems to me. I wish to thank Markus Holzer and Hermann Gruber for an interesting discussion. I am indebted to Galina Jirásková for her comments on the manuscript. Thanks are due to Michael Domaratzki for alerting me of some related work, and to Michal Kunc and Tommi Lehtinen for advising me how to prove Proposition 1.

Research supported by the Academy of Finland under grant 118540.

# References

[1] J.-C. Birget, "Intersection and union of regular languages and state complexity", *Information Processing Letters*, 43 (1992), 185–190.

[2] C. Câmpeanu, K. Salomaa, S. Yu, "Tight lower bound for the state complexity of shuffle of regular languages", *Journal of Automata, Languages and Combinatorics*, 7 (2002), 303–310.

[3] H. Gruber, M. Holzer, M. Kutrib, "The size of Higman–Haines sets", *Theoretical Computer Science*, 2007, to appear.

[4] H. Gruber, M. Holzer, M. Kutrib, "More on the size of Higman–Haines sets: effective constructions", *Machines, Computations and Universality* (MCU 2007, Orléans, France, September 10–14, 2007), LNCS 4664, 193–204.

[5] G. Jirásková, "State complexity of some operations on binary regular languages", *Theoretical Computer Science*, 330 (2005) 287–298.

[6] G. Jirásková, A. Okhotin, "State complexity of cyclic shift", *RAIRO Informatique Théorique et Applications*, to appear; preliminary version presented at DCFS 2005.

[7] A. N. Maslov, "Estimates of the number of states of finite automata", *Soviet Mathematics Doklady*, 11 (1970), 1373–1375.

[8] N. Rampersad, “The state complexity of $L^2$ and $L^k$”, *Information Processing Letters*, 98 (2006), 231–234.

[9] A. Salomaa, K. Salomaa, S. Yu, “State complexity of combined operations”, *Theoretical Computer Science*, 383:2–3 (2007), 140–152.

[10] A. Salomaa, D. Wood, S. Yu, “On the state complexity of reversals of regular languages”, *Theoretical Computer Science*, 320 (2004), 315–329.

[11] J. Shallit, “New directions in state complexity”, *DCFS 2006.*

[12] R. P. Stanley, *Enumerative Combinatorics*, vol. 2, Cambridge University Press, 1999.

# Appendix: lower bound on Catalan numbers

**Proposition 1.** *For every $k \geqslant 1$, $C_k > \frac{4^k}{(k+1)^{3/2}\sqrt{\pi}}$.*

*Proof.* Using Stirling's approximation of the factorial in the form

$$\sqrt{2\pi}n^{n+\frac{1}{2}}e^{-n+\frac{1}{12n+1}} < n! < \sqrt{2\pi}n^{n+\frac{1}{2}}e^{-n+\frac{1}{12n}},$$

the following lower bound on Catalan number $C_k = \frac{(2k)!}{k!(k+1)!}$ can be obtained:

$$\begin{aligned}
C_k &> \frac{\sqrt{2\pi}(2k)^{2k+\frac{1}{2}}e^{-2k+\frac{1}{24k+1}}}{\sqrt{2\pi}k^{k+\frac{1}{2}}e^{-k+\frac{1}{12k}} \cdot \sqrt{2\pi}(k+1)^{k+1+\frac{1}{2}}e^{-k-1+\frac{1}{12(k+1)}}} \\
&= \frac{1}{\sqrt{2\pi}}2^{2k+\frac{1}{2}}\frac{k^{2k+\frac{1}{2}}}{k^{k+\frac{1}{2}}(k+1)^{k+\frac{3}{2}}}\frac{e^{-2k}}{e^{-2k-1}}e^{\frac{1}{24k+1}-\frac{1}{12k}-\frac{1}{12k+12}} \\
&= \frac{e}{\sqrt{\pi}}4^k\frac{k^k}{(k+1)^{k+\frac{3}{2}}}e^{\frac{1}{24k+1}-\frac{1}{12k}-\frac{1}{12k+12}} \\
&= \frac{e}{\sqrt{\pi}}\frac{4^k}{(k+1)^{\frac{3}{2}}}\left(\frac{k}{k+1}\right)^k e^{\frac{1}{24k+1}-\frac{1}{12k}-\frac{1}{12k+12}} \\
&> \frac{1}{\sqrt{\pi}}\frac{4^k}{(k+1)^{\frac{3}{2}}}\frac{e}{\left(1+\frac{1}{k}\right)^k}e^{-\frac{1}{6k}}
\end{aligned}$$

It remains to prove that

$$\frac{e}{\left(1+\frac{1}{k}\right)^k}e^{-\frac{1}{6k}} \geqslant 1.$$

This statement can be equivalently rewritten as

$$\left(1+\frac{1}{k}\right)^{\frac{6k^2}{6k-1}} \leqslant e$$

Since $\frac{6k^2}{6k-1} = k + \frac{1}{6} + \frac{1}{6(6k-1)} < k + \frac{1}{5}$, it is sufficient to establish the following stronger statement for each $k \geqslant 2$:

$$\left(1+\frac{1}{k}\right)^{k+\frac{1}{5}} \leqslant e.$$

To establish this, take the logarithm of both sides and substitute $x = \frac{1}{k}$, so that it remains to prove the following statement for all $x \in (0,1)$:

$$\left(\frac{1}{x}+\frac{1}{5}\right)\ln(1+x) \leqslant 1$$

Expanding the logarithm into a Taylor series, one obtains

$$\left(\frac{1}{x}+\frac{1}{5}\right)\sum_{n=0}^{\infty}\frac{(-1)^n}{n+1}x^{n+1}=\sum_{n=0}^{\infty}\frac{(-1)^n}{n+1}x^n+\sum_{n=1}^{\infty}\frac{(-1)^{n+1}}{5n}x^n=$$
$$1+\sum_{n=1}^{\infty}(-1)^n\frac{4n-1}{5n(n+1)}x^n,$$

and hence the statement to be proved takes the form

$$\sum_{n=1}^{\infty}(-1)^{n+1}\frac{4n-1}{5n(n+1)}x^n\geqslant 0.$$

Grouping pairs of consecutive terms, the series can be expressed as

$$\sum_{n=1}^{\infty}x^{2n}\left(\frac{8n-5}{10n(2n-1)}-x\frac{8n-1}{10n(2n+1)}\right)$$

Now every term of the series is positive, because

$$\frac{8n-5}{10n(2n-1)}>\frac{8n-1}{10n(2n+1)}>x\frac{8n-1}{10n(2n+1)},$$

where the first inequality is easy to verify, while the second one holds true because $x<1$. This completes the proof. □

Turku
Centre *for*
Computer
Science

Lemminkäisenkatu 14 A, 20520 Turku, Finland | www.tucs.fi

University of Turku

- Department of Information Technology
- Department of Mathematical Sciences

Åbo Akademi University

- Department of Computer Science
- Institute for Advanced Management Systems Research

Turku School of Economics and Business Administration

- Institute of Information Systems Sciences

ISBN 978–952–12–1968–9
ISSN 1239–1891