Saint Petersburg State University

Semyon Petrov

Diploma thesis

Complexity of transforming two-way finite automata to unambiguous finite automata

Educational program
Bachelor of Mathematics

Course code: 01.03.01 Mathematics
Code EP: CB.5000.2016

Supervisor:

Professor
Faculty of Mahematics and Computer Science
Saint Petersburg State University
Doctor of Philosophy
A. S. Okhotin

Reviewer:

Associate Teaching Professor
Carnegie Mellon University in Qatar
Doctor of Philosophy
C. A. Kapoutsis

Saint Petersburg
2020

Contents

1 Introduction

Many variants of finite automata are known, and although all of them define the same class of *regular languages*, they differ in terms of succinctness of description. In particular, it is well-known that every *nondeterministic finite automaton* (NFA) with n states can be transformed to a *deterministic finite automaton* (DFA) with 2^n states, and this number of states is in the worst case necessary. This kind of succinctness tradeoffs have been studied for quite a few types of finite automata.

Transformations involving *two-way finite automata*, deterministic (2DFA) and nondeterministic (2NFA), have received particular attention in the literature [1, 2, 3, 9, 11, 12, 17, 18, 19]. In particular, the question of whether two-way automata can be determinized using polynomially many states is one of the most important open problems of automata theory, due to its connection to the L vs. NL problem [8]. Their transformation to one-way automata was studied over the years [1, 12, 18], until Kapoutsis [7] presented an optimal transformation. Kapoutsis [7] showed how to transform an n-state 2DFA to an NFA with $\binom{2n}{n+1}$ states, and proved that this number of states is necessary in the worst case; transforming a 2DFA to a DFA takes $n(n^n - (n-1)^n)$ states in the worst case [7].

Between these two perfectly conclusive results, there is an open question involving an intermediate model between DFA and NFA: the *unambiguous finite automata* (UFA), which can use nondeterminism, yet are bound to accept each string in at most one computation (as in the unambiguous complexity classes, such as UL and UP). The size of UFA has recently received some attention.

As shown by Leung [10], transforming an n-state UFA to a DFA requires 2^n states in the worst case, whereas the NFA-to-UFA transformation incurs a blowup from n to 2^n-1 states. In the case of a unary alphabet, transforming a UFA to a DFA in the worst case takes $e^{\Theta(\sqrt[3]{n\log^2 n})}$ states, and the NFA-to-UFA transformation requires $e^{\Theta(\sqrt{n\log n})}$ states [13]. Jirásek Jr. et al. [5] showed that complementing a UFA requires at least $2^{0.79n}$ states, with an upper bound of 2^n states. In the unary case, the known lower bound on complementing a UFA is $n^{\Omega(\log\log\log n)}$, due to Raskin [14].

Turning to the complexity of the 2DFA-to-UFA transformation, it is bound to lie between the two bounds of Kapoutsis [7], $\binom{2n}{n+1}$ and $n(n^n - (n-1)^n)$, and it is natural to ask what is the exact function in this case. This question is addressed in the present work.

The first task is to establish an upper bound that would improve over the 2DFA-to-DFA transformation. This is achieved by augmenting the NFA constructed by Kapoutsis [7] to store extra data that allows it to ensure the uniqueness of its accepting computation. The resulting UFA, presented in Section 3, has fewer than $2^n \cdot n!$ states. This construction can also be used for 2UFA-to-UFA trasformation.

Turning to a lower bound on the 2DFA-to-UFA transformation, a witness language is defined in Section 4 by constructing a 2DFA. The plan is to prove a lower bound on the size of every UFA recognizing the same language using *Schmidt's theorem* [15], which relies on the rank of a certain matrix related to the language. The rank of the matrix constructed in this work is estimated by first applying some linear transformation, and then reducing the problem to finding the rank of another matrix, defined entirely in terms of permutations.

Lower bounds on the rank of the latter matrix are established in Section 5. An easy, purely combinatorial estimation yields a lower bound of $\Omega((4\sqrt{2})^n \cdot n^{-1/2})$ states on the 2DFA-to-UFA tradeoff. An improved lower bound on the rank is obtained using the group representation theory, so that the task of calculating the rank is reduced to finding the dimension of a certain linear space. Then, the classical Maschke's theorem is used to decompose that linear space into a direct sum of irreducible representations of an understandable form, from which one can estimate the desired dimension. This way, it shall be proved that the 2DFA-to-UFA tradeoff is at least $\Omega(9^n \cdot n^{-3/2})$. Naturally, this value is also the lower bound on the 2UFA-to-UFA tradeoff. It is conjectured that the lower bound on the rank of the matrix established in this work is optimal, and under this assumption, the latter lower bound is the best that could be obtained using Schmidt's theorem.

2 Definitions

The work uses standard finite automata models: two-way deterministic automata, one-way unambiguous automata and two-way unambiguous automata.

Definition 1. A *two-way deterministic finite automaton* (2DFA) is a quintuple $\mathcal{A} = (\Sigma, Q, q_0, \delta, F)$, in which Σ is a finite alphabet; Q is a finite set of states; $q_0 \in Q$ is the initial state; $\delta \colon Q \times (\Sigma \cup \{\vdash, \dashv\}) \to Q \times \{-1, +1\}$ is the transition function, which defines a transition in a given state while observing a given tape symbol; $F \subseteq Q$ is the set of accepting states, effective at the right end-marker $\dashv$.

Given an input string $w = a_1 \ldots a_\ell$, a 2DFA operates on a read-only tape $\vdash w \dashv$. It begins its computation in the initial state, with the head at the left end-marker ($\vdash$). At every step of the computation, the automaton is in a state $q \in Q$ and observes a symbol $a \in \Sigma \cup \{\vdash, \dashv\}$; the transition function gives a pair $\delta(q, a) = (r, d)$ representing the next state and the direction in which the head moves. The set of strings, on which the computation eventually reaches the right end-marker in an accepting state, is denoted by $L(\mathcal{A})$.

Definition 2. A *nondeterministic finite automaton* (NFA) is a quintuple $\mathcal{B} = (\Sigma, Q, Q_0, \delta, F)$, in which Σ is a finite alphabet; Q is a finite set of states; $Q_0 \subseteq Q$ is the set of initial states; the transition function $\delta \colon Q \times \Sigma \to 2^Q$ defines possible next states after reading a given symbol in a given state; $F \subseteq Q$ is the set of accepting states.

On an input string $w = a_1 \ldots a_\ell$, a *computation* is a sequence of states $p_0, p_1, \ldots, p_\ell$ satisfying $p_0 \in Q_0$ and $p_{i+1} \in \delta(p_i, a_{i+1})$ for all i. It is *accepting* if, furthermore, $p_\ell \in F$. The set of strings, on which there is at least one accepting computation, is denoted by $L(\mathcal{B})$.

Definition 3. A *two-way nondeterministic finite automaton* (2NFA) is a quintuple $\mathcal{C} = (\Sigma, Q, Q_0, \delta, F)$, in which Σ is a finite alphabet; Q is a finite set of states; $Q_0 \subseteq Q$ is the set of initial states; the transition function $\delta \colon Q \times (\Sigma \cup \{\vdash, \dashv\}) \to 2^{Q \times \{-1, +1\}}$ defines possible transitions after reading a given symbol in a given state; $F \subseteq Q$ is the set of accepting states, effective at the right end-marker $\dashv$.

On an input string $w = a_1 \ldots a_\ell$, a *computation* is a sequence of pairs $(p_0, k_0), (p_1, k_1), \ldots, (p_n, k_n)$ where $p_i \in Q$ is a state, k_i is the position of the head, with $0 \leqslant k_i \leqslant \ell + 1$. Also, the following conditions should be satisfied: $p_0 \in Q_0$, $k_0 = 0$, and $(p_{i+1}, k_{i+1} - k_i) \in \delta(p_i, a_{k_i})$ for all i. Here, $a_0 = \vdash$ and $a_{\ell+1} = \dashv$.

The computation is *accepting* if, furthermore, $p_n \in F$ and $k_n = \ell + 1$, and there is no $i < n$ such that $p_i \in F$, $k_i = \ell + 1$. The set of strings, on which there is at least one accepting computation, is denoted by $L(\mathcal{C})$.

An NFA or 2NFA is said to be *unambiguous* (UFA or 2UFA respectively), if there is at most one accepting computation on each string.

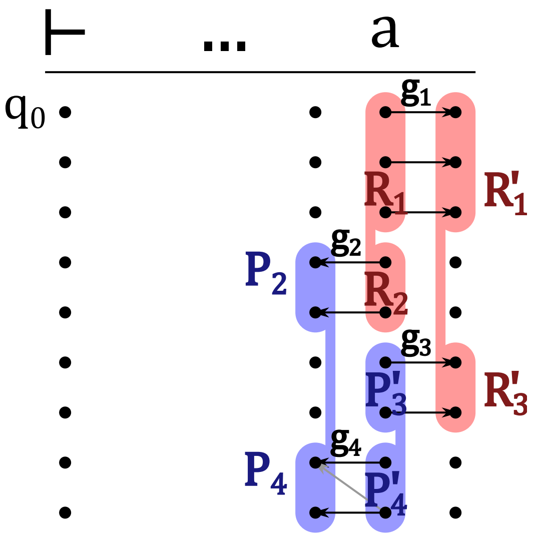


Figure 1: A transition between frontiers (P, R) and (P', R'). Generally, some automaton transitions might be unused; gray arrow represents this.

3 Upper bound

The proposed new transformation of 2UFA to UFA is derived from the known 2NFA-to-NFA transformation by Kapoutsis [7].

3.1 Transformation from 2NFA to NFA (by Kapoutsis)

For a 2NFA with a set of states Q, Kapoutsis [7] constructs an NFA with states of the form (P, R), with $P, R \subseteq Q$ and $|P| + 1 = |R|$. For an input string uv, after reading a prefix u, the NFA guesses a *frontier* of one of the 2NFA's computations on $\vdash uv \dashv$: this is a pair (P, R), where the set R consists of all states, into which the 2NFA moves to the right from the last symbol of u; states in P are those, into which the 2NFA moves to the last symbol of u from the right. The constructed NFA guesses a 2NFA computation's frontier at every step of its computation.

Theorem A (Kapoutsis [7]). *For every 2NFA with n states, there exists an NFA with $\binom{2n}{n+1}$ states that recognizes the same language.*

Proof. The set of initial states Q'_0 of the new NFA is the set of all frontiers reachable after processing the left end-marker $\vdash$. More formally, for every $q_0 \in Q_0$ and for every $q_1 \in Q$ such that $(q_1, +1) \in \delta(q_0, \vdash)$, Q'_0 contains all frontiers (P, R) such that $q_1 \in R$ and there exists a bijection $f\colon P \to R \setminus \{q_1\}$ with the following property: $(f(p), +1) \in \delta(p, \vdash)$ for any $p \in P$.

Transitions are defined as follows: for any symbol a and two frontiers (P, R) and (P', R') there is a transition from (P, R) to (P', R') on the symbol a (that is, $(P', R') \in \delta'((P, R), a)$) if $R \cap P' = \varnothing$ and there exist partitions $P = P_2 \uplus P_4$, $R = R_1 \uplus R_2$, $P' = P'_3 \uplus P'_4$ and $R' = R'_1 \uplus R'_3$ with bijections $g_1\colon R_1 \to R'_1$, $g_2\colon R_2 \to P_2$, $g_3\colon P'_3 \to R'_3$ and $g_4\colon P'_4 \to P_4$ such that the following conditions are satisfied:

1. if $r \in R_1$, then $(g_1(r), +1) \in \delta(r, a)$;
2. if $r \in R_2$, then $(g_2(r), -1) \in \delta(r, a)$;
3. if $p' \in P'_3$, then $(g_3(p'), +1) \in \delta(p', a)$;
4. if $p' \in P'_4$, then $(g_4(p'), -1) \in \delta(p', a)$.

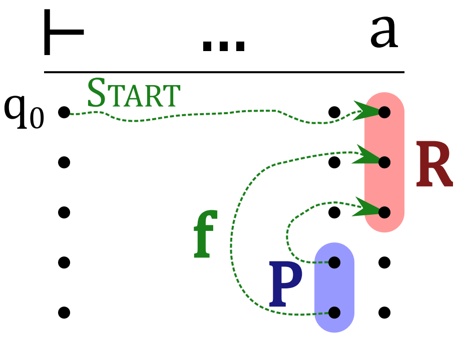


Figure 2: A profile (P, R, f) of a computation that holds short of reading the symbol a.

These rules ensure that transitions between frontiers are consistent with the transitions of the 2NFA. Transition rules are illustrated in Figure 1.

Finally, a frontier (P, R) is accepting in the NFA, if there exists a state $q \in R \cap F$ such that all other elements of P and R can be split into pairs (p, r) with $(p, -1) \in \delta(r, \dashv)$.

Since there are $\binom{2n}{n+1}$ frontiers for n states, the constructed NFA has $\binom{2n}{n+1}$ states. The correctness of the construction can be proven by induction on the length of a string. □

This construction is known to be optimal already for the transformation of a 2DFA to a NFA. The task is to adapt the construction to produce a UFA, given a 2UFA.

3.2 Transformation from 2UFA to UFA

The NFA constructed by the method of Kapoutsis is, in general, ambiguous, because, while guessing the next frontier, it may produce a closed cycle alongside the main computation. This closed cycle shall eventually be cancelled out, without the NFA's noticing, whereas the correctly guessed accepting computation of the 2UFA would drive the NFA to acceptance. This yields multiple accepting computations. This is possible even if the given 2UFA is deterministic.

The above construction shall now be elaborated to ensure unambiguity. Besides a pair (P, R), the automaton shall remember a bijection $f : P \cup \{\textsc{Start}\} \to R$ representing the states in R reached from each state in P, as well as from the initial configuration. Such a triple (P, R, f), illustrated in Figure 2, shall be called a *(prefix) profile*.

How many profiles are there? For every $k = |R|$, there are $\binom{n}{k-1}$ ways to choose the set P, and $\binom{n}{k}$ ways to choose the set R, and $k!$ different bijections f. Overall, there are $\sum_{k=1}^{n} \binom{n}{k-1}\binom{n}{k}k!$ profiles. With the frontiers replaced by profiles, the following theorem is obtained.

Theorem 1. *For every n-state 2UFA, there is a UFA with $\sum_{k=1}^{n} \binom{n}{k-1}\binom{n}{k}k!$ states that recognizes the same language.*

Proof. Let $A = (\Sigma, Q, Q_0, \delta, F)$ be an n-state 2UFA. The goal is to construct a UFA $B = (\Sigma, Q', Q'_0, \delta', F')$ with $\sum_{k=1}^{n} \binom{n}{k-1}\binom{n}{k}k!$ states that recognizes the same language.

Let Q' be the set of all profiles for the set Q. Then, $|Q'| = \sum_{k=1}^{n} \binom{n}{k-1}\binom{n}{k}k!$, and the condition on the number of states is satisfied.

Transitions are defined as follows: for any symbol a and two profiles (P, R, f) and (P', R', f') such that $R \cap P' = \varnothing$, there is a transition from (P, R, f) to (P', R', f') on the symbol a (that is, $(P', R', f') \in \delta'((P, R, f), a))$ if there exist partitions $P = P_2 \uplus P_4$, $R = R_1 \uplus R_2$, $P' = P'_3 \uplus P'_4$ and $R' = R'_1 \uplus R'_3$ with bijections $g_1 : R_1 \to R'_1$, $g_2 : R_2 \to P_2$, $g_3 : P'_3 \to R'_3$ and $g_4 : P'_4 \to P_4$ such that the following conditions are satisfied:

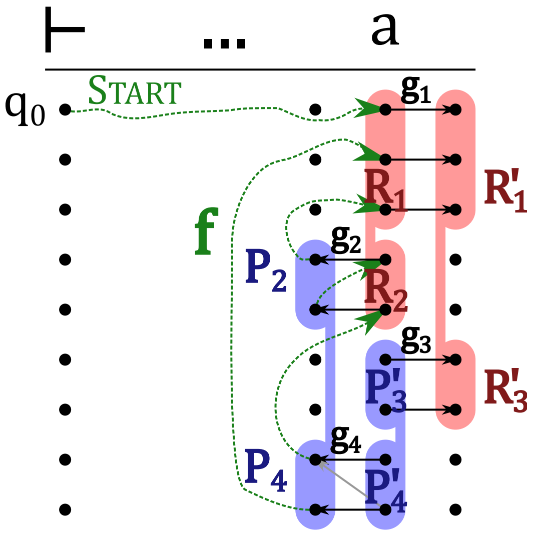


Figure 3: A transition between profiles (P, R, f) and (P', R', f'). Generally, some automaton transitions might be unused; gray arrow represents this.

1. if $r \in R_1$, then $(g_1(r), +1) \in \delta(r, a)$;
2. if $r \in R_2$, then $(g_2(r), -1) \in \delta(r, a)$;
3. if $p' \in P'_3$, then $(g_3(p'), +1) \in \delta(p', a)$;
4. if $p' \in P'_4$, then $(g_4(p'), -1) \in \delta(p', a)$.

These conditions are the same as in the construction by Kapoutsis in Theorem A. In addition, there are also conditions on the functions f and f':

- if $p' \in P'_3$, then $f'(p') = g_3(p')$ ($g_3(p')$ is reached from p' in one move);
- if $p' \in P'_4$, then there exists a sequence $p_1, r_1 = f(p_1), p_2 = g_2(r_1), \ldots, p_m = g_2(r_{m-1}), r_m = f(p_m)$ such that $p_1 = g_4(p')$ and $f'(p') = g_1(r_m)$ (new path from p' to $f'(p')$ as combination of old ones). The states p_i and r_i represent old paths between P_2 and R_2;
- there is no sequence $p_1, r_1 = f(p_1), p_2 = g_2(r_1), \ldots, p_m = g_2(r_{m-1}), r_m = f(p_m)$ such that $g_2(r_m) = p_1$ (no cycles in the computation).

The last condition is the most important one, as it eliminates the source of ambiguity in the construction by Kapoutsis.

Transition rules are illustrated in Figure 3. For element $p' \in P'$ the value of $f'(p')$ can be obtained by using arrows until entering the set R'.

The set of starting states Q'_0 is defined in terms of transitions as the set of profiles reachable from any of the profiles $(\varnothing, \{q_0\}, \textsc{Start} \mapsto q_0)$ for $q_0 \in Q_0$ by the left end-marker $\vdash$.

To define the set of accepting states F', let $\delta(q, \dashv) = \{(q, +1)\}$ if $q \in F$. Then, F is the set of profiles from which at least one of profiles $(\varnothing, \{q\}, \textsc{Start} \mapsto q)$ for $q \in F$ can be reached by symbol $\dashv$.

It is left to prove that B is unambiguous and recognizes the same language as 2UFA A.

Claim 1. *If 2UFA A accepts a string w, then B also accepts the string w.*

Proof. Consider the accepting computation of A on the string w of length n. Construct profiles (P^i, R^i, f^i) for every symbol w_i of the string for $i \in \{1, \ldots, n+1\}$ with $w_{n+1} = \dashv$ as follows:

- P^i is the set of states in which the computation moves from the i-th to the $(i-1)$-th symbol,
- R^i is the set of states in which the computation moves from the $(i-1)$-th to the i-th symbol,
- $f^i(p)$ for $p \in P^i$ is the state in which the computation beginning at the $(i-1)$-th symbol in the state p first comes to the i-th symbol.

It is claimed that those profiles form an accepting computation of B.

Note that $(P^1, R^1, f^1) \in Q'_0$: indeed, it is impossible to move to left at the left end-marker $\vdash$, so $(f^1(p), +1) \in \delta(p, \vdash)$ for any $p \in P^1$. Also, $(f^1(\textsc{Start}), +1) \in \delta(q_0, \vdash)$. Then, (P^1, R^1, f^1) is reachable from $(\varnothing, \{q_0\}, \textsc{Start} \mapsto q_0)$ by the symbol $\vdash$: the corresponding partitions and bijections are $R_1 = \{q_0\}$, $R'_1 = \{f^1(\textsc{Start})\}$, $P'_3 = P^1$, $R'_3 = R^1 \setminus \{f^1(\textsc{Start})\}$, $g_1 : q_0 \mapsto f^1(\textsc{Start})$, $g_3 = f^1|_{P^1}$.

Similarly, $(P^{n+1}, R^{n+1}, f^{n+1}) \in F'$: the partitions and bijections corresponding to the transition to $(\varnothing, \{q\}, \textsc{Start} \mapsto q)$ (where q is the last state in the accepting computation of A on w) are $R_1 = R'_1 = \{q\}$, $P_2 = P^{n+1}$, $R_2 = R^{n+1} \setminus \{q\}$, $g_1 : q \mapsto q$, and $g_2(r) = p$ if $\delta(r, \dashv) = (p, -1)$.

Finally, $(P^{i+1}, R^{i+1}, f^{i+1}) \in \delta'((P^i, R^i, f^i), w_i)$ for all $i \in \{1, \ldots, n-1\}$. To prove that, define R_1 as the set of states in R^i in which A moves to the right upon reading the i-th symbol, and R'_1 as the set of resulting states after such move. Other pairs of sets and bijections are defined likewise. All conditions will be satisfied, since both profiles are constructed from a valid computation without cycles and self-intersections. □

Claim 2. *If B accepts a string w, then A also accepts the string w. Furthermore, the accepting computation of B on w is unique.*

Proof. Let n be the length of w, and fix an accepting computation of B on this string. Denote by (P^i, R^i, f^i) the state of B before reading the i-th symbol (for $i \in \{1, \ldots, n\}$)). Also, let $(P^{n+1}, R^{n+1}, f^{n+1})$ be the state in which B finishes reading the string w. In addition, denote by (P^0, R^0, f^0) the profile $(\varnothing, \{q_0\}, \textsc{Start} \mapsto q_0)$ for $q_0 \in Q_0$ such that (P^1, R^1, f^1) is reachable from it by the symbol $\vdash$. Accordingly, denote by $(P^{n+2}, R^{n+2}, f^{n+2})$ the profile $(\varnothing, \{q\}, \textsc{Start} \mapsto q)$ for $q \in F$ such that it is reachable from $(P^{n+1}, R^{n+1}, f^{n+1})$ by the symbol $\dashv$. The goal is to reconstruct an accepting computation of A on w from this sequence of profiles.

Construct a graph G with vertices labeled with pairs (q, i), where $q \in Q$, and $i \in \{0, \ldots, n+1\}$. Add to G all arrows checked by the automaton B in the accepting computation. Formally, for each $i \in \{0, \ldots, n+1\}$ and $q \in Q$ such that $q \in R^i \cup P^{i+1}$, let q' be the state corresponding to the state q via the bijections used in the transition from profile (P^i, R^i, f^i) to profile $(P^{i+1}, R^{i+1}, f^{i+1})$ (then, $q' \in R^{i+1} \cup P^i$). Let $s = +1$ if q' was obtained through transition from the state q to the right, and let $s = -1$ otherwise. Then there is an arrow in G from (q, i) to $(q', i+s)$, if $i + s \in \{0, \ldots, n+1\}$. This arrow corresponds to the next step of a computation of A.

The arrows in the graph G represent the valid moves of the automaton A. Hence, paths in G are valid computations for A. An example of such graph G is illustrated in Figure 4. The values of functions f^i are given as green arrows.

In the graph G, all vertices, except $(q_0, 0)$ for $q_0 \in Q_0$ — the only element of R^0 and $(q, n+1)$ for $q \in F$ — the only element of P^{n+2}, either have both indegree and outdegree equal to 0 (for pairs not present in the computation), or have both indegree and outdegree equal to 1 (for pairs present in the computation). Indeed, let (q, i) be a vertex in the graph G. If q is neither in R^i nor in P^{i+1}, then there are no arrows in or out of (q, i). If q is in R^i or P^{i+1} (and q cannot be in both at the same time by the definition of transition between profiles), then there is exactly

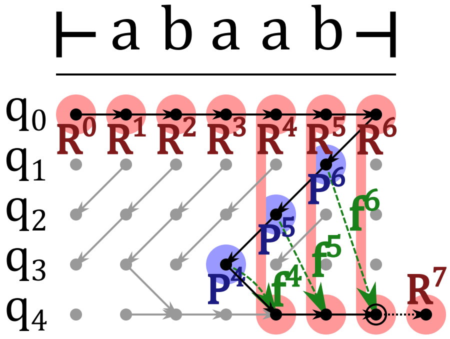


Figure 4: An example of graph built from accepting computation of B.

one arrow into (q, i) (except $(q, i) = (q_0, 0)$) and exactly one arrow out of (q, i) (except $i = n+1$ and $q \in F$).

Hence, the graph G consists of a path from $(q_0, 0)$ to $(q, n+1)$ and, possibly, some disjoint simple cycles. To finish the proof of this claim, the following result is needed:

Claim 2.1. *Let $i \in \{1, \dots, n+1\}$ and $q \in P^i$. Consider the path from $(q, i-1)$ in G. Let (q', i) be the first vertex on this path with the second coordinate equal to i. Then, q' exists and is equal to $f^i(q)$.*

Proof. The claim is proved by induction on i.

Consider the case of $i = 1$ first. Consider a partition $P^0 = P_2 \uplus P_4$, $R^0 = R_1 \uplus R_2$, $P^1 = P'_3 \uplus P'_4$ and $R^1 = R'_1 \uplus R'_3$ with bijections $g_1 \colon R_1 \to R'_1$, $g_2 \colon R_2 \to P_2$, $g_3 \colon P'_3 \to R'_3$ and $g_4 \colon P'_4 \to P_4$ used in the transition from the profile (P^0, R^0, f^0) to the profile (P^1, R^1, f^1) by the symbol $\vdash$. Since P^0 is empty, so are P_2 and P_4. Since g_4 is a bijection from P_4 to P'_4, the set P'_4 is also empty, and $q \in P'_3$. Then, by definition, the graph G contains an arrow from $(q, 0)$ to $(g_3(q), 1)$. Hence, $q' = g_3(q)$. Since $f^1(q) = g_3(q)$ thanks to transition rules, the result of the claim follows.

Suppose now that the claim is proved for $i-1$. As with the case of $i = 1$, consider a partition $P^{i-1} = P_2 \uplus P_4$, $R^{i-1} = R_1 \uplus R_2$, $P^i = P'_3 \uplus P'_4$ and $R^i = R'_1 \uplus R'_3$ with bijections $g_1 \colon R_1 \to R'_1$, $g_2 \colon R_2 \to P_2$, $g_3 \colon P'_3 \to R'_3$ and $g_4 \colon P'_4 \to P_4$ used in the transition from the profile $(P^{i-1}, R^{i-1}, f^{i-1})$ to the profile (P^i, R^i, f^i) by the symbol w_{i-1}. If $q \in P'_3$, then G contains an arrow from $(q, i-1)$ to $(g_3(q), i)$, and $q' = g_3(q) = f^i(q)$.

The remaining case is that $q \in P'_4$. Then, by the transition rules, there exists a sequence $p_1, r_1 = f^{i-1}(p_1), p_2 = g_2(r_1), \dots, p_m = g_2(r_{m-1}), r_m = f^{i-1}(p_m)$ such that $p_1 = g_4(q)$ and $f^i(q) = g_1(r_m)$. Then, the graph G contains the arrows from $(r_k, i-1)$ to $(p_{k+1}, i-2)$ for every $k \in \{1, \dots, m-1\}$, and also an arrow from $(q, i-1)$ to $(p_1, i-2)$ and an arrow from $(r_m, i-1)$ to $(f^i(q), i)$. Furthermore, by the induction assumption, for every $k \in \{1, \dots, m\}$ the point $(r_k, i-1)$ is the first point on the path from $(p_k, i-2)$ with the second coordinate equal to $i-1$. Since the arrows in the graph q connect only vertices which second coordinates differ by 1, said path from $(p_k, i-2)$ to $(r_k, i-1)$ never visits vertices with second coordinate equal to i (since it would need to pass through $i-1$ first). Hence, the whole united path from $(q, i-1)$ to $(r_m, i-1)$ never visits such vertices, and $(f^i(q), i)$ is the first one. Then, $q' = f^i(q)$. □

Claim 2.2. *Graph G does not contain cycles.*

Proof. Assume the opposite. Consider any such cycle C; let i be the maximal second coordinate of a vertex therein. Denote by R the set of states q such that (q, i) is a vertex of C. For every $q \in R$ the path from (q, i) never visits vertices with the second coordinate equal to $i+1$. Therefore, $R \cap P^{i+1} = \varnothing$. Since $R \subseteq R^i \cup P^{i+1}$, it follows that $R \subseteq R^i$.

Denote by P the set of states p such that G contains an arrow from (q,i) to $(p,i-1)$ for some $q \in R$. In this situation, define $g(q) = p$. Since every state of R has one such arrow, and no two arrows have the same endpoint, the sizes of P and R are equal, and g is a bijection from R to P. Also note that $P \subseteq P^i$, since vertices $(p,i-1)$ with $p \in P$ have the arrow pointed at them from (q,i) with $q \in R^i$.

Now, let f be the restriction of the function f^i to P. By Claim 2.1, all values of f lie in R, since the path from $(p,i-1)$ cannot leave the cycle C. The arrows in f represent contractions of paths in C outside of the position i.

Then, the cycle is detected by B in the transition between profiles (P^i, R^i, f^i) and $(P^{i+1}, R^{i+1}, f^{i+1})$, as follows. Consider a partition $P^i = P_2 \uplus P_4$, $R^i = R_1 \uplus R_2$, $P^{i+1} = P'_3 \uplus P'_4$ and $R^{i+1} = R'_1 \uplus R'_3$ with bijections $g_1 \colon R_1 \to R'_1$, $g_2 \colon R_2 \to P_2$, $g_3 \colon P'_3 \to R'_3$ and $g_4 \colon P'_4 \to P_4$ used in the transition from the profile (P^i, R^i, f^i) to the profile $(P^{i+1}, R^{i+1}, f^{i+1})$ by the symbol w_i. Note that g is the restriction of g_2 to R by definition.

Then, it is possible to construct the sequence $p_1, r_1 = f^i(p_1), p_2 = g_2(r_1), \ldots, p_m = g_2(r_{m-1}), r_m = f^i(p_m)$ such that $g_2(r_m) = p_1$, which would mean that this transition is invalid. Indeed, pick $p_1 \in P$ and define the rest of p_k and r_k using functions f and g instead of f^i and g_2. Sunce P and R are finite, then $p_j = p_k$ for some $j < k$. Then the sequence starting at p_j and ending at r_{k-1} would contradict the rules of transition between profiles. □

By Claim 2.2, there are no cycles in G, and it contains only a path from $(q_0, 0)$ to $(q, n+1)$. This path is unique, because it corresponds to an accepting computation of 2UFA A on the string w, and A has no more than one accepting computation on this string. Since different sequences of profiles define different graphs (the graph fixes P^i and R^i; and f^i can be restored through the use of Claim 2.1), the computation of B is also unique. □

Therefore, B is unambiguous and recognizes the same language as A. □

3.3 The asymptotics of the upper bound

Lemma 1. $\sum_{k=1}^{n} \binom{n}{k-1}\binom{n}{k} k! < 2^n \cdot n! < n(n^n - (n-1)^n)$ *for every integer* $n \geqslant 3$.

Proof. The first inequality is proven by the following transformations: $\sum_{k=1}^{n} \binom{n}{k-1}\binom{n}{k} k! = \sum_{k=1}^{n} \binom{n}{k-1} \frac{n! \cdot k!}{(n-k)! \cdot k!} \leqslant \sum_{k=1}^{n} \binom{n}{k-1} \cdot n! = (2^n - 1) \cdot n! < 2^n \cdot n!$.

Then, Stirling's approximation states that $n! \leqslant e \cdot \sqrt{n} \left(\frac{n}{e}\right)^n$, and the above expression is bounded as $2^n \cdot n! \leqslant 2^n \cdot e \cdot \sqrt{n} \cdot \frac{n^n}{e^n} = \frac{e \cdot \sqrt{n}}{\left(\frac{e}{2}\right)^n} \cdot n^n$.

On the other hand, $n(n^n - (n-1)^n) = n(n-(n-1))(n^{n-1} + n^{n-2}(n-1) + \cdots + (n-1)^{n-1}) \geqslant n \cdot 1 \cdot n^{n-1} = n^n$.

It is left to determine, for which n the inequality $\frac{e \cdot \sqrt{n}}{\left(\frac{e}{2}\right)^n} < 1$ holds, and verify the lemma for small values of n.

The latter inequality is rewritten as $e \cdot \sqrt{n} < \left(\frac{e}{2}\right)^n$, which is equivalent to $e^2 n < \left(\frac{e^2}{4}\right)^n$.

If $n = 7$, then $e^2 n < 52$ and $\left(\frac{e^2}{4}\right)^n > 73$, so the inequality is true. When $n \geqslant 7$ is increased by 1, the left-hand side grows by $e^2 < 8$ while the right-hand side grows at least by $\left(\frac{e^2}{4}\right)^8 - \left(\frac{e^2}{4}\right)^7 > 60$. Hence, the inequality holds for all $n \geqslant 7$.

The values of the expressions in the lemma for $n \leqslant 6$ are listed in Table 1. □

By Lemma 1, the number of profiles is less than $2^n \cdot n!$, which is in turn asymptotically less than $n(n^n - (n-1)^n)$. This confirms that the proposed transformation to UFA is more

Table 1: Values of expressions in Lemma 1 for $3 \leqslant n \leqslant 6$.

n	$\sum_{k=1}^{n} \binom{n}{k-1}\binom{n}{k} k!$	$2^n \cdot n!$	$n(n^n - (n-1)^n)$
3	39	48	57
4	292	384	700
5	2 505	3 840	10 505
6	24 306	46 080	186 186

efficient than transforming a 2DFA to a DFA, as per another construction by Kapoutsis [6] (a comparison for small values of n shall be given later on in Table 2).

Note that the number of profiles, $\sum_{k=1}^{n} \binom{n}{k-1}\binom{n}{k} k! \geqslant n!$, is much larger than the number of frontiers, $\sum_{k=1}^{n} \binom{n}{k-1}\binom{n}{k} = \binom{2n}{n-1} \leqslant 4^n$.

4 The preparatory work for lower bound

A lower bound on the state complexity of transforming a 2DFA to a UFA is based on a witness language recognized by a small 2DFA, for which every equivalent UFA would require a substantial number of states.

4.1 The witness language and its 2DFA

The witness language is defined over an alphabet $\Gamma = (\{1, \ldots, n\} \cup \{\, f \mid f \colon \{1, \ldots, n\} \to \{1, \ldots, n\} \text{ is a partial function}\,\}) \times \{l, r\}$, and is recognized by a 2DFA $\mathcal{D}_n$ with the set of states $Q = \{1, \ldots, n\}$, with $q_0 = 1$ and $F = \{1\}$. It uses the following transitions, defined for all $x, y \in Q$ and $f, g \colon \{1, \ldots, n\} \to \{1, \ldots, n\}$.

$$\delta(q_0, \vdash) = (q_0, +1), \quad \delta(q, (x, \mathtt{l})) = (x, +1), \quad \delta(q, (f, \mathtt{l})) = \begin{cases} (f(q), +1), & \text{if } f(q) \text{ is defined} \\ (q_0, -1), & \text{otherwise} \end{cases}$$

$$\delta(q, (g, \mathtt{r})) = \begin{cases} (g(q), -1), & \text{if } g(q) \text{ is defined} \\ (q, +1), & \text{otherwise} \end{cases} \qquad \delta(q, (y, \mathtt{r})) = \begin{cases} (q_0, +1), & \text{if } q = y \\ (q, -1), & \text{otherwise} \end{cases}$$

This is the automaton used by Kapoutsis [6] in his lower bound for the transformation of a 2NFA to an NFA. Following Kapoutsis, the subsequent proof uses four-symbol strings of the form $(x, \mathtt{l})\,(f, \mathtt{l})\,(g, \mathtt{r})\,(y, \mathtt{r})$, with $x, y \in Q$ and with partial functions f, g, where $f(x)$ is defined and $g(y)$ is not. These strings correspond to directed (n, n)-bipartite graphs, with f representing arrows from left to right, and g, from right to left. The automaton $\mathcal{D}_n$ then verifies, whether there is a path from x in the left part to y on the right.

The rest of this work is concerned with proving a lower bound on the size of every UFA recognizing the language $L(\mathcal{D}_n)$. The only known method for proving such lower bounds is the following theorem.

Theorem B (Schmidt [15], see also Leung [10]). *Let L be a regular language, and let (x_1, y_1), …, (x_n, y_n) be pairs of strings. Let M be an integer matrix defined by $M_{i,j} = 1$, if $x_i y_j \in L$, and $M_{i,j} = 0$ otherwise. Then, every UFA for L has at least* $\operatorname{rank} M$ *states.*

Sketch of a proof. Let U be a UFA that recognizes the language L. Construct the following matrix M' with rows corresponding to the states of U and columns corresponding to the strings y_i, with values $M'_{q,y_i} = 1$ if $y_i \in L_q(U)$ (y_i is accepted from state q) and $M'_{q,y_i} = 0$ otherwise.

Note that every row of the matrix M is a sum of some rows of the matrix M'; namely, of those that correspond to states reachable from q_0 after reading x_i. Hence, M' cannot contain fewer than $\operatorname{rank} M$ rows, and the number of states of U is at least $\operatorname{rank} M$. □

4.2 The choice of strings for Schmidt's theorem

In addition to the prefix profiles, describing a computation of an automaton on a prefix, a new type of profile shall be introduced.

Definition 4. A *suffix profile* is a triple (g, P, R), where $P, R \subseteq Q$, $|P| + 1 = |R|$, and $g \colon R \to P \cup \{\textsc{Accept}\}$ is a bijection.

For a given computation of automaton on the string uv, the suffix profile of v complements the prefix profile of u. The function g in the suffix profile is constructed in a similar way as in the prefix profile: for any state q, the state $g(q)$ is the state in which the automaton first

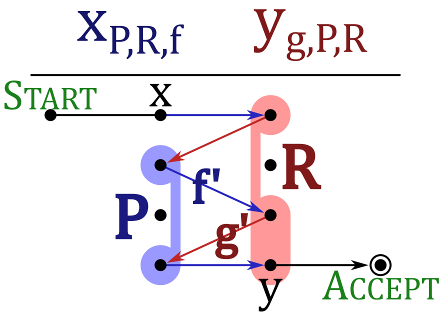


Figure 5: An example of interaction between a prefix profile and a matching suffix profile.

crosses the border between u and v in this computation after visiting the first symbol of v in the state q — or $g(q) = \textsc{Accept}$, if it accepts without crossing this border.

How many suffix profiles are there? For fixed P and R with $|R| = k$, there are $k!$ ways to choose a prefix profile (P, R, f) and $k!$ ways to choose a suffix profile (g, P, R). So, the number of suffix profiles is equal to the number of prefix profiles.

The strings for Schmidt's theorem are chosen as follows. Let (P, R, f) be a prefix profile, where $f \colon P \cup \{\textsc{Start}\} \to R$ is a bijection, and let $x \in \{1, \ldots, n\}$, with $x \notin P$; such an x exists, since $|P| < n$. Let $f' \colon P \cup \{x\} \to R$ be a function defined by $f'(p) = f(p)$ for $p \in P$, and $f'(x) = f(\textsc{Start})$. Then, define $x_{P,R,f} = (x, \mathtt{l})\,(f', \mathtt{l})$. The state x is added to create a bijection.

Next, for a suffix profile (g, S, T), let y be the element of T such that $g(y) = \textsc{Accept}$. Let $g' \colon T \setminus \{y\} \to S$ be a function defined by $g'(r) = g(r)$. Define $y_{g,S,T} = (g', \mathtt{r})\,(y, \mathtt{r})$. Again, the state y is removed to create a bijection.

For the computation of some 2UFA on a string uv, the prefix profile (P, R, f) of u and the suffix profile (g, P, R) of v define strings $x_{P,R,f}$ and $y_{g,P,R}$ that contain functions f' and g' which complement each other to a full path. Figure 5 illustrates this relation.

Note that $f'(x)$ is defined and $g'(y)$ is not, otherwise the string $x_{P,R,f}\, y_{g,S,T}$ would have no chance to be accepted by $\mathcal{D}_n$.

The strings $x_{P,R,f}$ and $y_{g,S,T}$ are constructed in such a way, that (P, R, f) is the largest prefix profile of $\mathcal{D}_n$ on the string $x_{P,R,f}$, and (g, S, T) is the largest suffix profile of $\mathcal{D}_n$ on the string $y_{g,S,T}$.

Let $M^{(n)}$ be the square matrix defined for these strings in Schmidt's theorem. Each row corresponds to a prefix profile (P, R, f), each column corresponds to a suffix profile (g, S, T), and the element at their intersection is denoted by $M^{(n)}_{(P,R,f),(g,S,T)}$. The order of the matrix is the total number of profiles, that is, $\sum_{k=1}^{n} \binom{n}{k-1} \binom{n}{k} k!$.

Definition 5. On an input $(x, \mathtt{l})\,(f, \mathtt{l})\,(g, \mathtt{r})\,(y, \mathtt{r})$, the automaton $\mathcal{D}_n$ is said to *use the left-to-right arrow from p to r*, if, at some point in its computation, $\mathcal{D}_n$ is in the state p on the symbol $(f, \mathtt{l})$, and $f(p) = r$. Similarly, $\mathcal{D}_n$ *uses the right-to-left arrow from r to p*, if, at some point, $\mathcal{D}_n$ is in the state r at $(g, \mathtt{r})$, and $g(r) = p$.

In order to estimate the rank of this matrix, it shall first be subjected to a series of rank-preserving transformations.

4.3 Inclusion-exclusion formula for rows

The first transformation considers rows and replaces them with linear combinations of the original matrix' rows in a reversible way, such that the rank is preserved. In order to do that, several lemmata shall be proved first.

Definition 6. A prefix profile (P', R', f') shall be called a *subprofile* of a prefix profile (P, R, f), if $P' \subseteq P$, $R' \subseteq R$ and $f'(p) = f(p)$ for all $p \in P' \cup \{\textsc{Start}\}$. Notation: $(P', R', f') \preccurlyeq (P, R, f)$.

Lemma 2. *Let $(P', R', f') \preccurlyeq (P, R, f)$ be two prefix profiles, and let (g, S, T) be a suffix profile such that $M^{(n)}_{(P',R',f'),(g,S,T)} = 1$. Then $M^{(n)}_{(P,R,f),(g,S,T)} = 1$. Furthermore, the computations of $\mathcal{D}_n$ on strings $x_{P',R',f'}\, y_{g,S,T}$ and $x_{P,R,f}\, y_{g,S,T}$ are the same, except at the second step (when $\mathcal{D}_n$ visits second symbol for the first time).*

In other words: if there already is a path, then the addition of new left-to-right arrows will not change it.

Proof. Let $(x_{f'}, \mathtt{1})$ be the first symbol of the string $x_{P',R',f'}\, y_{g,S,T}$, and let $(x_f, \mathtt{1})$ be the first symbol of the string $x_{P,R,f}\, y_{g,S,T}$.

The lemma is proved by induction on the step number k, starting from $k = 3$. After three steps, the computation on $x_{P',R',f'}\, y_{g,S,T}$ gets to the third symbol in the state $f'(\textsc{Start})$, and the computation on $x_{P,R,f}\, y_{g,S,T}$ gets to the third symbol in the state $f(\textsc{Start})$; since $f(\textsc{Start}) = f'(\textsc{Start})$, the induction base is proven.

After that, if k is even, then the k-th computation step depends only on the third symbol, and third symbols in the strings $x_{P',R',f'}\, y_{g,S,T}$ and $x_{P,R,f}\, y_{g,S,T}$ are equal. If k is odd, then both computations before the k-th step were either on the fourth symbol (and they proceed identically, since the fourth symbols are equal), or on the second symbol. If they were on the second symbol, then the next step is identical for all states except $x_{f'}$ and x_f (if they are not equal), and states from $P \setminus P'$. In the former case, computation on $x_{P',R',f'}\, y_{g,S,T}$ would loop, which is a contradiction. In the latter two cases, it would not be accepting, since the second symbol of the string $x_{P',R',f'}\, y_{g,S,T}$ lacks arrow from x_f, because $x_f \neq x_{f'}$, $x_f \notin P \supseteq P'$ and $(P \setminus P') \cup P' = \varnothing$.

As the computations are equal, their outcomes are also the same. Hence, $M^{(n)}_{(P,R,f),(g,S,T)} = 1$. □

Lemma 3. *Let $(P', R', f') \preccurlyeq (P, R, f)$ be two prefix profiles, and let (g, S, T) be a suffix profile such that $M^{(n)}_{(P,R,f),(g,S,T)} = 0$. Then, $M^{(n)}_{(P',R',f'),(g,S,T)} = 0$.*

In other words: if there is no path, then the deletion of left-to-right arrows will not create a new one.

Proof. Assume otherwise, that $M^{(n)}_{(P',R',f'),(g,S,T)} = 1$. By Lemma 2, then $M^{(n)}_{(P,R,f),(g,S,T)} = 1$, which yields a contradiction. □

Lemma 4. *Let $(P'', R'', f'') \preccurlyeq (P', R', f') \preccurlyeq (P, R, f)$ be prefix profiles, and let (g, S, T) be a suffix profile such that $\mathcal{D}_n$ on the string $x_{P,R,f}\, y_{g,S,T}$ never visits the second symbol in a state from $P' \setminus P''$. Then, $M^{(n)}_{(P',R',f'),(g,S,T)} = M^{(n)}_{(P'',R'',f''),(g,S,T)}$.*

In other words: if path does not use some left-to-right arrows, then their removal will not affect the existence of a path.

Proof. If $M^{(n)}_{(P',R',f'),(g,S,T)} = 0$, then, by Lemma 3, $M^{(n)}_{(P'',R'',f''),(g,S,T)} = 0$, and the lemma is proved.

Suppose then, that $M^{(n)}_{(P',R',f'),(g,S,T)} = 1$. By Lemma 2, the computations of $\mathcal{D}_n$ on strings $x_{P,R,f}\, y_{g,S,T}$ and $x_{P',R',f'}\, y_{g,S,T}$ coincide (with one exception). Hence, $\mathcal{D}_n$ never visits the second symbol in any state from $P' \setminus P''$ on string $x_{P',R',f'}\, y_{g,S,T}$. Then, this computation (with a state change for the second step) is also an accepting computation on the string $x_{P'',R'',f''}\, y_{g,S,T}$. This can be proved by the same induction as in Lemma 2, with a slight modification when k is odd (the next step is not identical for $q \in P' \setminus P''$, but the computation on the string $x_{P',R',f'}\, y_{g,S,T}$ never enters the second symbol in those states). □

Definition 7. Define a new square integer matrix $L^{(n)}$ of the same order as $M^{(n)}$, with rows and columns indexed by prefix and suffix profiles, respectively. Each element $L^{(n)}_{(P,R,f),(g,S,T)}$ is defined as 1, if $\mathcal{D}_n$ accepts the string $x_{P,R,f}\, y_{g,S,T}$ and uses all left-to-right arrows in the corresponding graph. Otherwise, $L^{(n)}_{(P,R,f),(g,S,T)} = 0$.

It turns out that the rows of $L^{(n)}$ are linear combinations of the rows of $M^{(n)}$ expressed by the inclusion–exclusion principle.

Lemma 5. *Let (P, R, f) be a prefix profile and let (g, S, T) be a suffix profile. Then,*

$$\begin{aligned} L^{(n)}_{(P,R,f),(g,S,T)} &= \sum_{(P',R',f')\preccurlyeq(P,R,f)} (-1)^{|P|-|P'|} M^{(n)}_{(P',R',f'),(g,S,T)} \\ M^{(n)}_{(P,R,f),(g,S,T)} &= \sum_{(P',R',f')\preccurlyeq(P,R,f)} L^{(n)}_{(P',R',f'),(g,S,T)} \end{aligned}$$

Accordingly, $\operatorname{rank} L^{(n)} = \operatorname{rank} M^{(n)}$.

Proof. The first equality follows from the second one, because

$$\sum_{(P',R',f')\preccurlyeq(P,R,f)} (-1)^{|P|-|P'|} M^{(n)}_{(P',R',f'),(g,S,T)} =$$

$$= \sum_{(P'',R'',f'')\preccurlyeq(P',R',f')\preccurlyeq(P,R,f)} (-1)^{|P|-|P'|} L^{(n)}_{(P'',R'',f''),(g,S,T)} =$$

$$= L^{(n)}_{(P,R,f),(g,S,T)}$$

The last equality shall be expained in more detail. For any given prefix subprofile $(P'', R'', f'') \preccurlyeq (P, R, f)$, the expression $L^{(n)}_{(P'',R'',f''),(g,S,T)}$ is present in the sum with coefficient $\sum_{(P'',R'',f'')\preccurlyeq(P',R',f')\preccurlyeq(P,R,f)} (-1)^{|P|-|P'|}$. If $(P'', R'', f'') = (P, R, f)$, then necessarily $(P', R', f') = (P, R, f)$, and this coefficient is equal to 1. Suppose then, that $(P'', R'', f'') \neq (P, R, f)$; let $m = |P| - |P''|$ ($m > 0$). There are 2^{m-1} ways to delete odd number of arrows out of m arrows, so there are 2^{m-1} prefix profiles (P', R', f') for which $|P| - |P'|$ is odd, and each of them adds -1 to coefficient. Similarly, there are 2^{m-1} prefix profiles (P', R', f') for which $|P| - |P'|$ is even, and each of them adds 1 to coefficient. Those two groups cancel themselves out, and resulting coefficient is 0.

If $M^{(n)}_{(P,R,f),(g,S,T)} = 0$, then, by Lemma 3, $M^{(n)}_{(P',R',f'),(g,S,T)} = 0$ holds for every prefix subprofile (P', R', f'). Then, $L^{(n)}_{(P',R',f'),(g,S,T)} = 0$ for every prefix subprofile (P', R', f') of prefix profile (P, R, f), and $\sum_{(P',R',f')\preccurlyeq(P,R,f)} L^{(n)}_{(P',R',f'),(g,S,T)} = 0$. Hence, the second equality is true in this case.

If $M^{(n)}_{(P,R,f),(g,S,T)} = 1$, let P_0 be the set of all states in which $\mathcal{D}_n$ visits the second symbol of the string $x_{P,R,f}\, y_{g,S,T}$, except for the first visit.

If $L^{(n)}_{(P',R',f'),(g,S,T)} = 1$ for some subprofile (P', R', f') of profile (P, R, f), then $M^{(n)}_{(P',R',f'),(g,S,T)} = 1$, and, by Lemma 2, the computations of $\mathcal{D}_n$ on the strings $x_{P',R',f'}\, y_{g,S,T}$ and $x_{P,R,f}\, y_{g,S,T}$ coincide (with one exception related to the first visit of the second symbol). Since $\mathcal{D}_n$ uses all left-to-right arrows on the string $x_{P',R',f'}\, y_{g,S,T}$, then $P' = P_0$. Then the subprofile (P', R', f') with $P' = P_0$ is uniquely defined by restricting f to $P_0 \cup \{\textsc{Start}\}$. Therefore, $\sum_{(P',R',f')\preccurlyeq(P,R,f)} L^{(n)}_{(P',R',f'),(g,S,T)} \leqslant 1$.

For the subprofile (P', R', f'), with $P' = P_0$, Lemma 4 asserts that $M^{(n)}_{(P',R',f'),(g,S,T)} = 1$, since P_0 and $P \setminus P_0$ have no common elements. Then, $L^{(n)}_{(P',R',f'),(g,S,T)} = 1$ for that subprofile

(P', R', f'), and $\sum_{(P',R',f') \preccurlyeq (P,R,f)} L^{(n)}_{(P',R',f'),(g,S,T)} \geqslant 1$. Hence, the second equality holds true in this case as well.

Concerning the rank, the rows of $L^{(n)}$ are linear combinations of the rows of $M^{(n)}$, and vice versa. Therefore, $\operatorname{rank} L^{(n)} = \operatorname{rank} M^{(n)}$. □

4.4 Inclusion-exclusion formula for columns

The columns of the matrix shall now be transformed by the same method.

Definition 8. A suffix profile (g', S', T') shall be called a *subprofile* of a suffix profile (g, S, T), if $S' \subseteq S$, $T' \subseteq T$ and $g'(q) = g(q)$ for all $q \in T'$. Notation: $(g', S', T') \preccurlyeq (g, S, T)$.

Lemma 6. *Let (P, R, f) be a prefix profile, and let $(g', S', T') \preccurlyeq (g, S, T)$ be two suffix profiles such that $M^{(n)}_{(P,R,f),(g',S',T')} = 1$. Then $M^{(n)}_{(P,R,f),(g,S,T)} = 1$. Furthermore, the computations of $\mathcal{D}_n$ on strings $x_{P,R,f}\, y_{g',S',T'}$ and $x_{P,R,f}\, y_{g,S,T}$ are the same.*

In other words: if there already is a path, then the addition of new right-to-left arrows will not change it.

Proof. Let $(x_f, \mathtt{1})$ be the first symbol of the strings $x_{P,R,f}\, y_{g',S',T'}$, and $x_{P,R,f}\, y_{g,S,T}$.

The lemma is proved by induction on the step number k, starting from $k = 2$. After two steps, computations on $x_{P,R,f}\, y_{g',S',T'}$ and $x_{P,R,f}\, y_{g,S,T}$ get to the second symbol in the state x_f. Hence, the induction base is proven.

After that, if k is odd, then before the k-th step both computations were either on the fourth symbol (and they proceed identically, since the fourth symbols are equal), or on the second symbol, which are also equal. If k is even, then both computations were on the third symbol, and the next step is identical for all states except those from $T \setminus T'$, but then the computation on $x_{P,R,f}\, y_{g',S',T'}$ would not be accepting.

As the computations are equal, their outcomes are also the same. □

Lemma 7. *Let (P, R, f) be a prefix profile, and let $(g', S', T') \preccurlyeq (g, S, T)$ be two suffix profiles such that $L^{(n)}_{(P,R,f),(g',S',T')} = 1$. Then $L^{(n)}_{(P,R,f),(g,S,T)} = 1$.*

In other words: if a path uses all left-to-right arrows, then the addition of new right-to-left arrows would not change that.

Proof. Since $L^{(n)}_{(P,R,f),(g',S',T')} = 1$, then $M^{(n)}_{(P,R,f),(g',S',T')} = 1$. By Lemma 6, $M^{(n)}_{(P,R,f),(g,S,T)} = 1$ and the computations of $\mathcal{D}_n$ on strings $x_{P,R,f}\, y_{g',S',T'}$ and $x_{P,R,f}\, y_{g,S,T}$ coincide. Then the sets of used left-to-right arrows also coincide. Since $\mathcal{D}_n$ uses all left-to-right arrows on the string $x_{P,R,f}\, y_{g',S',T'}$, the same holds for the computation on $x_{P,R,f}\, y_{g,S,T}$, and $L^{(n)}_{(P,R,f),(g,S,T)} = 1$. □

Lemma 8. *Let (P, R, f) be a prefix profile, and let $(g', S', T') \preccurlyeq (g, S, T)$ be two suffix profiles such that $L^{(n)}_{(P,R,f),(g,S,T)} = 0$. Then, $L^{(n)}_{(P,R,f),(g',S',T')} = 0$.*

In other words: if there is no path that uses all left-to-right arrows, then the deletion of right-to-left arrows would not create a new one.

Proof. Assume otherwise, that $L^{(n)}_{(P,R,f),(g',S',T')} = 1$. By Lemma 7, then $L^{(n)}_{(P,R,f),(g,S,T)} = 1$, which yields a contradiction. □

Lemma 9. *Let (P, R, f) be a prefix profile, and let $(g'', S'', T'') \preccurlyeq (g', S', T') \preccurlyeq (g, S, T)$ be suffix profiles such that $\mathcal{D}_n$ on the string $x_{P,R,f}\, y_{g,S,T}$ never visits the third symbol in a state from $T' \setminus T''$. Then, $M^{(n)}_{(P,R,f),(g',S',T')} = M^{(n)}_{(P,R,f),(g'',S'',T'')}$.*

In other words: if path does not use some right-to-left arrows, then their removal would not affect the existence of a path.

Proof. If $M^{(n)}_{(P,R,f),(g',S',T')} = 0$, then, by Lemma 6, $M^{(n)}_{(P,R,f),(g'',S'',T'')}$ should also be 0, and the lemma is proved.

Suppose then, that $M^{(n)}_{(P,R,f),(g',S',T')} = 1$. By Lemma 6, the computations of $\mathcal{D}_n$ on strings $x_{P,R,f}\, y_{g,S,T}$ and $x_{P,R,f}\, y_{g',S',T'}$ coincide. Hence, $\mathcal{D}_n$ never visits the third symbol in any state from $T' \setminus T''$ on the string $x_{P,R,f}\, y_{g',S',T'}$. Then, this computation is also an accepting computation on the string $x_{P,R,f}\, y_{g'',S'',T''}$. This can be proved by the same induction as in Lemma 6, with a slight modification for even k (the next step is not identical for $q \in T' \setminus T''$, but the computation on the string $x_{P,R,f}\, y_{g',S',T'}$ never enters the third symbol in those states). □

Lemma 10. *Let (P, R, f) be a prefix profile, and let (g_1, S_1, T_1) and (g_2, S_2, T_2) be two suffix profiles. Suppose that $M^{(n)}_{(P',R',f'),(g_1,S_1,T_1)} = M^{(n)}_{(P',R',f'),(g_2,S_2,T_2)}$ for every profile $(P', R', f') \preccurlyeq (P, R, f)$. Then, $L^{(n)}_{(P,R,f),(g_1,S_1,T_1)} = L^{(n)}_{(P,R,f),(g_2,S_2,T_2)}$.*

Proof. By Lemma 5, $L^{(n)}_{(P,R,f),(g_1,S_1,T_1)} = \sum_{(P',R',f') \preccurlyeq (P,R,f)} (-1)^{|P|-|P'|} M^{(n)}_{(P',R',f'),(g_1,S_1,T_1)} = \sum_{(P',R',f') \preccurlyeq (P,R,f)} (-1)^{|P|-|P'|} M^{(n)}_{(P',R',f'),(g_2,S_2,T_2)} = L^{(n)}_{(P,R,f),(g_2,S_2,T_2)}$. □

Lemma 11. *Let (P, R, f) be a prefix profile, and let $(g'', S'', T'') \preccurlyeq (g', S', T') \preccurlyeq (g, S, T)$ be suffix profiles such that $\mathcal{D}_n$ on the string $x_{P,R,f}\, y_{g,S,T}$ never visits the third symbol in a state from $T' \setminus T''$. Then, $L^{(n)}_{(P,R,f),(g',S',T')} = L^{(n)}_{(P,R,f),(g'',S'',T'')}$.*

Proof. In order to prove that using Lemma 10, it shall be shown that $M^{(n)}_{(P',R',f'),(g',S',T')} = M^{(n)}_{(P',R',f'),(g'',S'',T'')}$ for every prefix profile $(P', R', f') \preccurlyeq (P, R, f)$.

If $M^{(n)}_{(P',R',f'),(g,S,T)} = 0$, then, by Lemma 6, $M^{(n)}_{(P',R',f'),(g',S',T')}$ and $M^{(n)}_{(P',R',f'),(g'',S'',T'')}$ should also be 0, and, in particular, $M^{(n)}_{(P',R',f'),(g',S',T')} = M^{(n)}_{(P',R',f'),(g'',S'',T'')}$.

Suppose then, that $M^{(n)}_{(P',R',f'),(g,S,T)} = 1$. By Lemma 2, the computations of $\mathcal{D}_n$ on strings $x_{P,R,f}\, y_{g,S,T}$ and $x_{P',R',f'}\, y_{g,S,T}$ coincide (with one exception, which does not affect the third symbol). Hence, $\mathcal{D}_n$ never visits the third symbol in any state from $T' \setminus T''$ on the string $x_{P',R',f'}\, y_{g,S,T}$. Then, by Lemma 9, $M^{(n)}_{(P',R',f'),(g',S',T')} = M^{(n)}_{(P',R',f'),(g'',S'',T'')}$.

Since $M^{(n)}_{(P',R',f'),(g',S',T')} = M^{(n)}_{(P',R',f'),(g'',S'',T'')}$ for every prefix profile $(P', R', f') \preccurlyeq (P, R, f)$, then, by Lemma 10, $L^{(n)}_{(P,R,f),(g',S',T')} = L^{(n)}_{(P,R,f),(g'',S'',T'')}$. □

Definition 9. Define yet another integer matrix $K^{(n)}$ of the same dimensions as $M^{(n)}$ and $L^{(n)}$, with its rows and columns again indexed by prefix profiles and suffix profiles respectively. Let $K^{(n)}_{(P,R,f),(g,S,T)}$ be 1, if $\mathcal{D}_n$ accepts $x_{P,R,f}\, y_{g,S,T}$ and uses all left-to-right and right-to-left arrows in the corresponding graph. Otherwise, let this element be 0.

Lemma 12. *Let (P, R, f) be a prefix profile, and let (g, S, T) be a suffix profile. Then,*

$$K^{(n)}_{(P,R,f),(g,S,T)} = \sum_{(g',S',T') \preccurlyeq (g,S,T)} (-1)^{|T|-|T'|} L^{(n)}_{(P,R,f),(g',S',T')}$$

$$L^{(n)}_{(P,R,f),(g,S,T)} = \sum_{(g',S',T') \preccurlyeq (g,S,T)} K^{(n)}_{(P,R,f),(g',S',T')}$$

In particular, $\operatorname{rank} K^{(n)} = \operatorname{rank} L^{(n)}$.

Proof. The first equality follows from the second one, because

$$\sum_{(g',S',T')\preccurlyeq(g,S,T)} (-1)^{|T|-|T'|} L^{(n)}_{(P,R,f),(g',S',T')} =$$

$$= \sum_{(g'',S'',T'')\preccurlyeq(g',S',T')\preccurlyeq(g,S,T)} (-1)^{|T|-|T'|} K^{(n)}_{(P,R,f),(g'',S'',T'')} =$$

$$= K^{(n)}_{(P,R,f),(g,S,T)}$$

The last equality shall be expained in more detail. For any given suffix subprofile $(g'', S'', T'') \preccurlyeq (g, S, T)$, the expression $K^{(n)}_{(P,R,f),(g'',S'',T'')}$ is present in the sum with coefficient $\sum_{(g'',S'',T'')\preccurlyeq(g',S',T')\preccurlyeq(g,S,T)} (-1)^{|T|-|T'|}$. If $(g'', S'', T'') = (g, S, T)$, then necessarily $(g', S', T') = (g, S, T)$, and this coefficient is equal to 1. Suppose then, that $(g'', S'', T'') \neq (g, S, T)$; let $m = |T| - |T''|$ $(m > 0)$. There are 2^{m-1} ways to delete odd number of arrows out of m arrows, so there are 2^{m-1} suffix profiles (g', S', T') for which $|T| - |T'|$ is odd, and each of them adds -1 to coefficient. Similarly, there are 2^{m-1} suffix profiles (g', S', T') for which $|T| - |T'|$ is even, and each of them adds 1 to coefficient. Those two groups cancel themselves out, and resulting coefficient is 0.

If $L^{(n)}_{(P,R,f),(g,S,T)} = 0$, then, by Lemma 8, $L^{(n)}_{(P,R,f),(g',S',T')} = 0$ holds for every suffix subprofile (g', S', T'). Then, $K^{(n)}_{(P,R,f),(g',S',T')} = 0$ for every suffix subprofile (g', S', T') of suffix profile (g, S, T), and $\sum_{(g',S',T')\preccurlyeq(g,S,T)} K^{(n)}_{(P,R,f),(g',S',T')} = 0$. Hence, the second equality is true in this case.

If $L^{(n)}_{(P,R,f),(g,S,T)} = 1$, let T_0 be the set of all states in which $\mathcal{D}_n$ visits the third symbol of the string $x_{P,R,f}\, y_{g,S,T}$.

If $K^{(n)}_{(P,R,f),(g',S',T')} = 1$ for some suffix subprofile (S', T', g') of profile (S, T, g), then $M^{(n)}_{(P,R,f),(g',S',T')} = 1$, and, by Lemma 6, the computations of $\mathcal{D}_n$ on the strings $x_{P,R,f}\, y_{g',S',T'}$ and $x_{P,R,f}\, y_{g,S,T}$ coincide. Since $\mathcal{D}_n$ uses all right-to-left arrows on the string $x_{P,R,f}\, y_{g',S',T'}$, then $T' = T_0$. There is exactly one suffix subprofile (g', S', T') with $T' = T_0$, hence $\sum_{(g',S',T')\preccurlyeq(g,S,T)} K^{(n)}_{(P,R,f),(g',S',T')} \leqslant 1$.

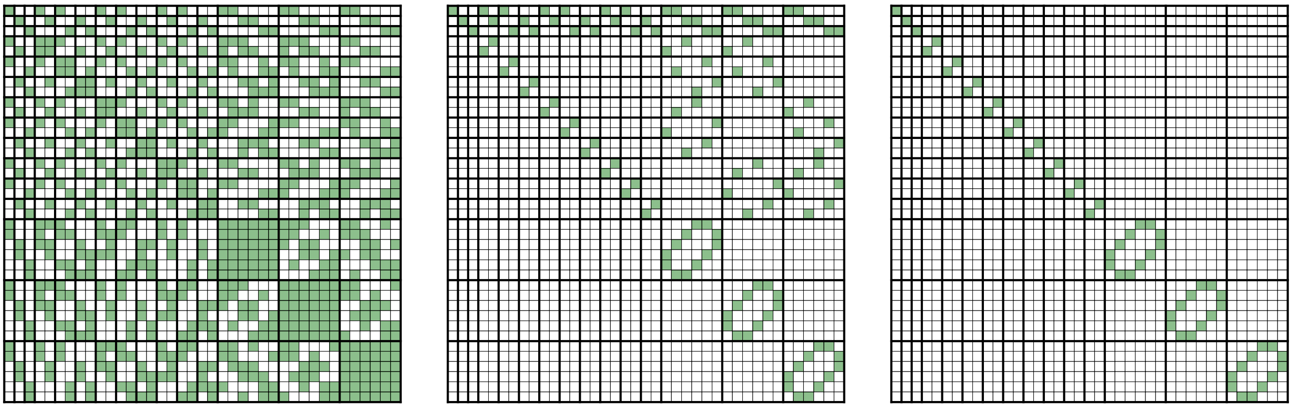
For the subprofile (g', S', T'), with $T' = T_0$, Lemma 9 asserts that $M^{(n)}_{(P',R',f'),(g,S,T)} = M^{(n)}_{(P',R',f'),(g',S',T')}$ for every prefix profile $(P', R', f') \preccurlyeq (P, R, f)$, since T_0 and $T \setminus T_0$ have no common elements. By Lemma 10, then $L^{(n)}_{(P,R,f),(g,S,T)} = L^{(n)}_{(P,R,f),(g',S',T')}$, and $L^{(n)}_{(P,R,f),(g',S',T')} = 1$. Then, $K^{(n)}_{(P,R,f),(g',S',T')} = 1$ for that suffix subprofile (g', S', T'), and $\sum_{(g',S',T')\preccurlyeq(g,S,T)} K^{(n)}_{(P,R,f),(g',S',T')} \geqslant 1$. Hence, the second equality holds true in this case as well.

Concerning the rank, the rows of $K^{(n)}$ are linear combinations of the rows of $L^{(n)}$, and vice versa. Therefore, $\operatorname{rank} K^{(n)} = \operatorname{rank} L^{(n)}$. □

The above transformations of the matrix $M^{(3)}$, which is of size 39 × 39, are given in Figure 6. The prefix profiles (P, R, f) are enumerated by ordering them first by $|P|$, and then lexicographically by P, by R and finally by the values of f. The suffix profiles (g, S, T) are similarly ordered first by $|S|$, and then lexicographically by S, by T and finally by the values of g.

4.5 The structure of the matrix after transformations

The figure suggests that **the matrix $K^{(3)}$ is block diagonal.** This is proved as follows.



$M^{(3)}$ $L^{(3)}$ $K^{(3)}$

Figure 6: Transformation of $M^{(3)}$. Each filled square contains 1, each empty square has 0.

Lemma 13. *Let (P, R, f) be a prefix profile, and let (g, S, T) be a suffix profile with $(P, R) \neq (S, T)$. Then, $K^{(n)}_{(P,R,f),(g,S,T)} = 0$.*

Proof. Consider the string $x_{P,R,f}\, y_{g,S,T} = (x, \mathtt{l})(f', \mathtt{l})(g', \mathtt{r})(y, \mathtt{r})$. Assume the contrary, that $K^{(n)}_{(P,R,f),(g,S,T)} = 1$. Then, $\mathcal{D}_n$ uses all left-to-right arrows in the corresponding graph. In particular, the set of states in which $\mathcal{D}_n$ arrives to $(g', \mathtt{r})$ from $(f', \mathtt{l})$ is R, since these are the heads of all left-to-right arrows, and the set of states in which $\mathcal{D}_n$ arrives to $(f', \mathtt{l})$ from $(g', \mathtt{r})$ is P: the tails of all left-to-right arrows, except the arrow from x to $f(\text{Start})$.

At the same time, $\mathcal{D}_n$ uses all right-to-left arrows, and the set of states in which $\mathcal{D}_n$ arrives to $(g', \mathtt{r})$ from $(f', \mathtt{l})$ is T, these are the tails of all right-to-left arrows with the addition of y, for which $g(y) = \text{Accept}$; the states in which $\mathcal{D}_n$ arrives to $(f', \mathtt{l})$ from $(g', \mathtt{r})$ is S, these are the heads of right-to-left arrows. Therefore, $P = S$ and $R = T$, which contradicts the assumption. □

Thus the matrix $K^{(n)}$ is organized into blocks corresponding to different pairs (P, R).

4.6 The matrix for permutations

The next important observation is that the **blocks corresponding to pairs (P_1, R_1) and (P_2, R_2), with $|P_1| = |P_2|$, are identical up to permutations of rows and columns.** Indeed, let $g \colon P_1 \to P_2$ and $h \colon R_1 \to R_2$ be fixed bijections. Then, for a prefix profile (P_1, R_1, f), set $g(\text{Start}) = \text{Start}$, and let the corresponding prefix profile be $(P_2, R_2, h \circ f \circ g^{-1})$. Accordingly, for a suffix profile (f, P_1, R_1), set $g(\text{Accept}) = \text{Accept}$, and let the corresponding suffix profile be $(g \circ f \circ h^{-1}, P_2, R_2)$. The existence of a path in a bipartite graph is invariant to such permutations of vertices. This block is denoted by $P^{(k)}$, where $k = |R_1| = |R_2|$.

Definition 10. Let $1 \leqslant k \leqslant n$. The *matrix for permutations* $P^{(k)}$ is a $k! \times k!$ submatrix of $K^{(n)}$ that consists of rows and columns corresponding to profiles (P, R, f) with $P = \{1, \ldots, k{-}1\}$ and $R = \{1, \ldots, k\}$. Its rows and columns are still indexed by prefix and suffix profiles prespectively, that is, the element corresponding to the functions f and g is denoted by $P^{(k)}_{(P,R,f),(g,P,R)}$.

The form of the matrix $P^{(k)}$ for $k = 2, 3, 4$ is presented in Figure 10. White squares represent zeroes, the rest of the squares contain 1.

Lemma 14. $\operatorname{rank} K^{(n)} = \sum_{k=1}^{n} \binom{n}{k-1}\binom{n}{k} \operatorname{rank} P^{(k)}$

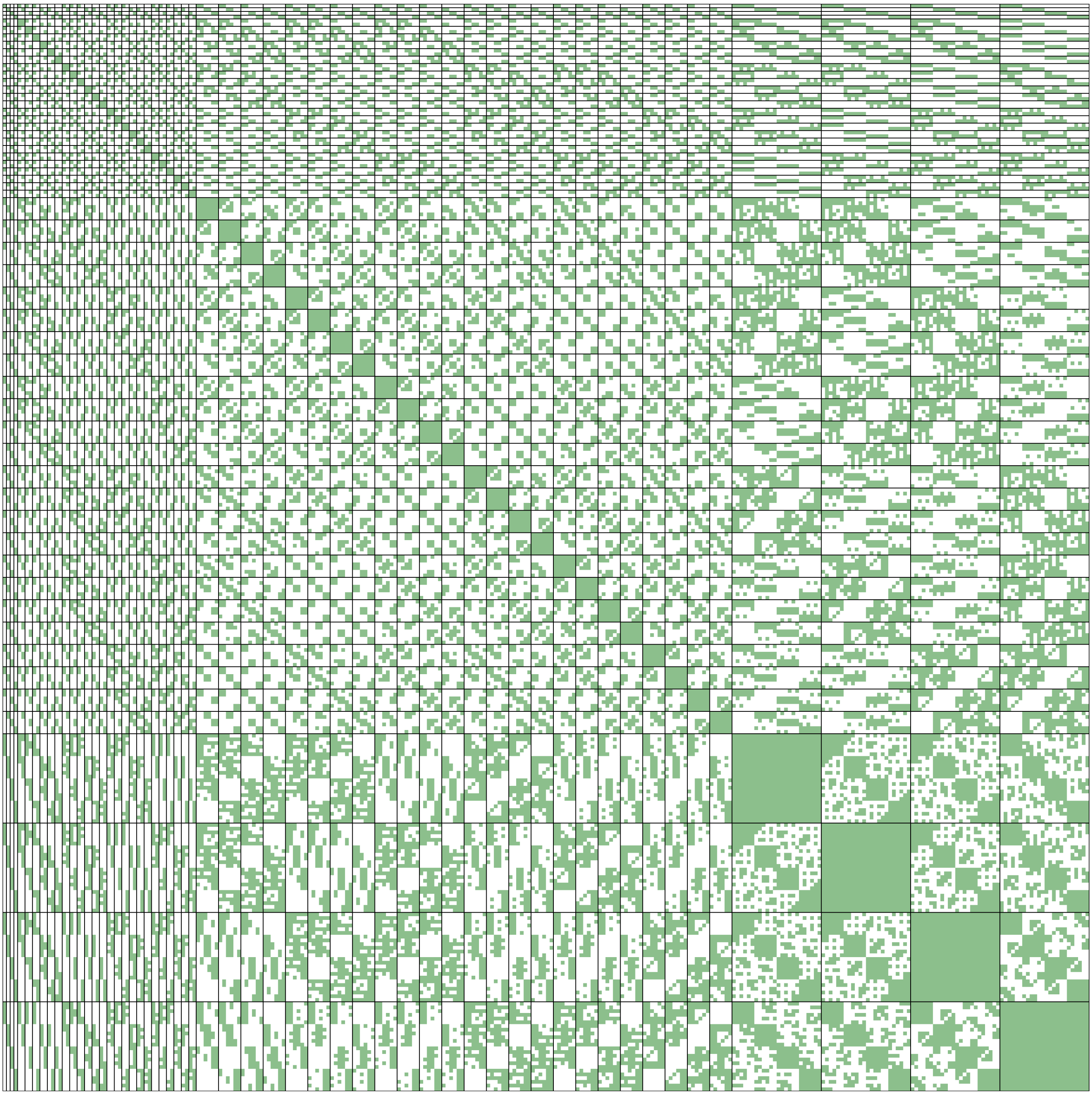


Figure 7: $M^{(4)}$.

Proof. By Lemma 13, the matrix $K^{(n)}$ is block diagonal, and hence rank $K^{(n)}$ is a sum of ranks of independent blocks. Since the blocks with the same $|P|$ are equivalent, each block is equivalent to the matrix for permutations $P^{(k)}$ with $k-1=|P|$. There are $\binom{n}{k-1}\binom{n}{k}$ different ways to choose a pair (P,R) so that $|P|=k-1$, hence for every k from 1 to n there are $\binom{n}{k-1}\binom{n}{k}$ blocks in $K^{(n)}$ that are equivalent to $P^{(k)}$. This gives the formula. □

Thus, it is sufficient to estimate the rank of the matrix for permutations.

Definition 11. Let (P,R,f) be a prefix profile with $P=\{1,\dots,k-1\}$ and $R=\{1,\dots,k\}$. Then, *the permutation corresponding to* (P,R,f) is a function $g\colon\{1,\dots,k\}\to\{1,\dots,k\}$ defined by $g(p)=f(p)$ for $p\in P$, and $g(k)=f(\textsc{Start})$.

Definition 12. Let (f,P,R) be a suffix profile with $P=\{1,\dots,k-1\}$ and $R=\{1,\dots,k\}$. Then, *the permutation corresponding to* (f,P,R) is a function $g\colon\{1,\dots,k\}\to\{1,\dots,k\}$ de-

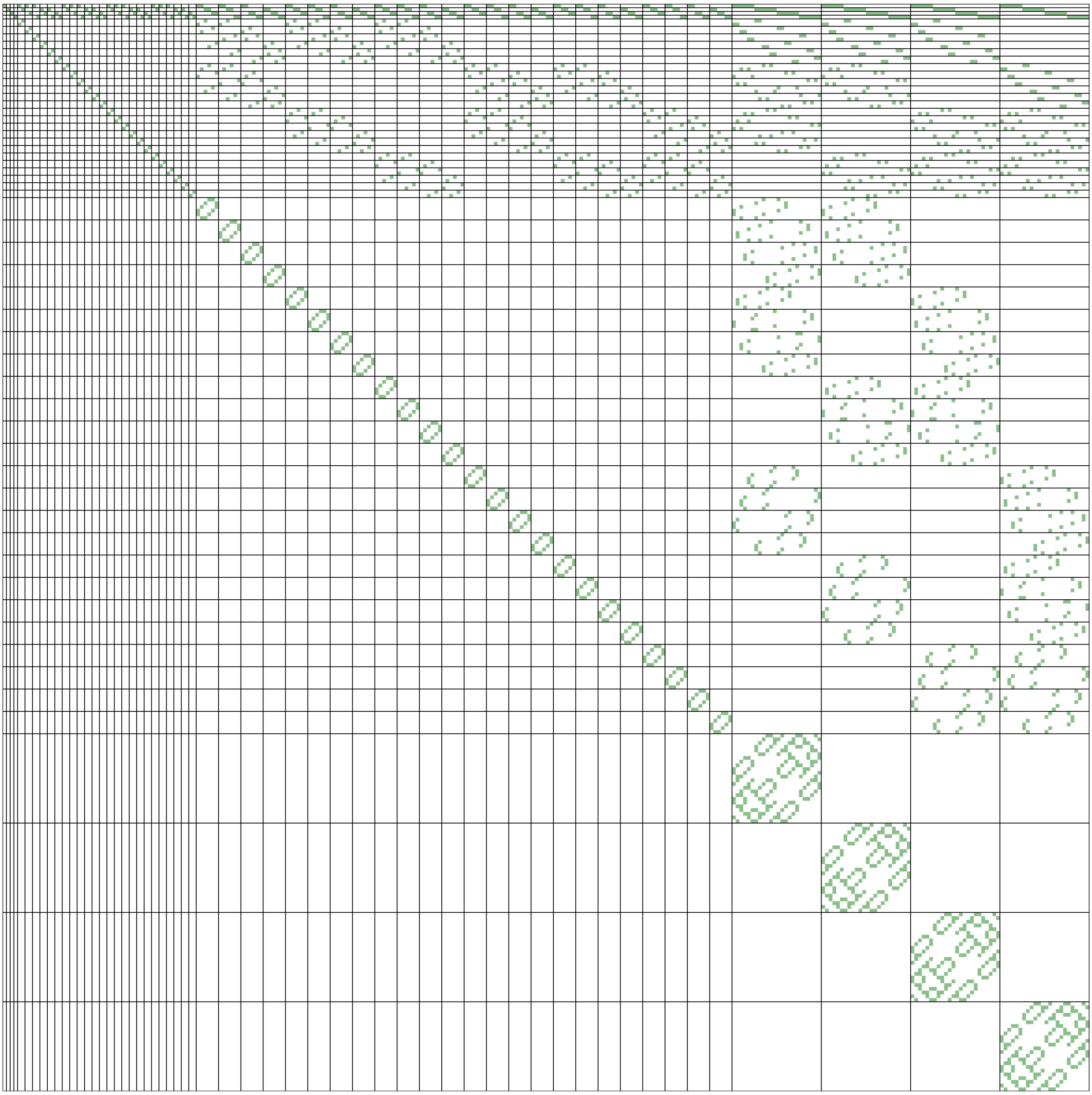


Figure 8: The matrix $L^{(4)}$, obtained after transformation of $M^{(4)}$.

fined by $g(p) = f(p)$ for $p \in R \setminus \{y\}$, and $g(y) = k$, where $y \in R$ is the state such that $f(y) = \textsc{Accept}$.

Note that, conversely, each permutation has a unique corresponding prefix profile, and an unique corresponding suffix profile of this form.

The elements of $P^{(k)}$ are characterized entirely in terms of permutations as follows. Let (P, R, f_1) be a prefix profile, and let (f_2, P, R) be a suffix profile, with $P = \{1, \ldots, k-1\}$ and $R = \{1, \ldots, k\}$. Let g_1 and g_2 be the corresponding permutations. Denote $P^{(k)}_{g_1,g_2} = P^{(k)}_{(P,R,f_1),(f_2,P,R)}$.

Lemma 15. $P^{(k)}_{g_1,g_2} = 1$ *if and only if the permutation* $g_2 \circ g_1$ *is cyclic.*

Proof. Let (P, R, f_1) and (f_2, P, R) be profiles corresponding to g_1 and to g_2, respectively.

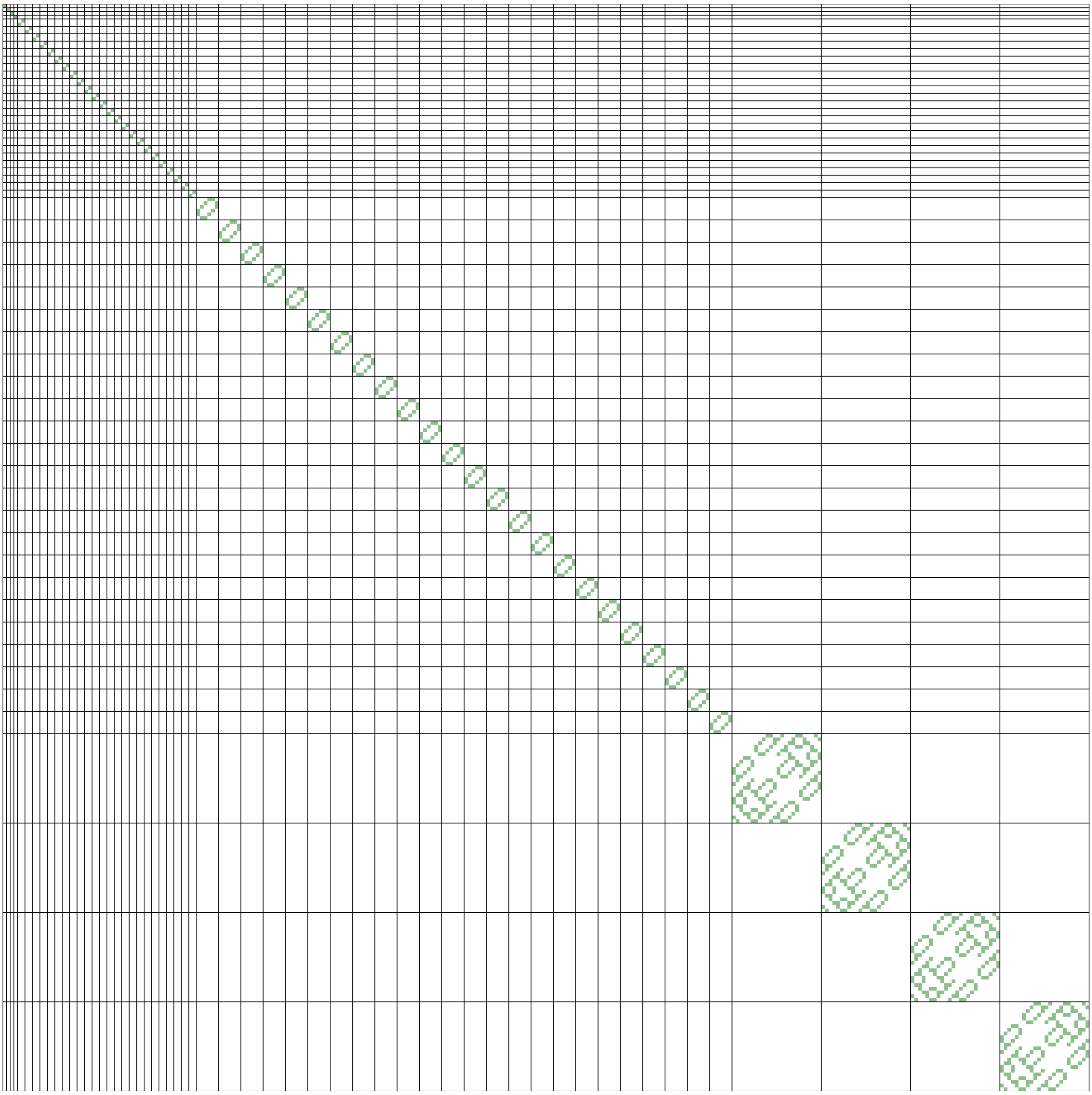


Figure 9: The matrix $K^{(4)}$, obtained after transformation of $L^{(4)}$.

Consider the computation of $\mathcal{D}_k$ on the string $x_{P,R,f_1}\, y_{f_2,P,R} = (x, \mathtt{l})(f_1', \mathtt{l})(f_2', \mathtt{r})(y, \mathtt{r})$. Then $x = k$, because $P = \{1, \ldots, k-1\}$, and $y \in R$ is the state such that $f_2(y) = \textsc{Accept}$, and $g_2(y) = k$. The automaton first moves to $(f_1', \mathtt{l})$ in the state k, and then alternates between the second and the third symbols. This computation is depicted on Figure 12.

Consider the sequence of states, in which it visits the second symbol $(f_1', \mathtt{l})$. The sequence begins with k. In a state q, the automaton moves to the third symbol in the state $g_1(q)$, and then immediately returns to the second symbol in the state $g_2 \circ g_1(q)$, as long as $g_1(q)$ is not equal to $y = g_2^{-1}(k)$, which is equivalent to $g_2 \circ g_1(q) \neq k$.

$\ominus\!\!\!\!\Rightarrow$ If $P^{(k)}_{g_1,g_2} = 1$, then the element $P^{(k)}_{(P,R,f_1),(f_2,P,R)}$ is 1. As $P^{(k)}$ is a submatrix of $K^{(k)}$, the element $K^{(k)}_{(P,R,f_1),(f_2,P,R)}$ is 1 as well. By definition, this means that $\mathcal{D}_k$ accepts $x_{P,R,f_1}\, y_{f_2,P,R}$, using all arrows in both directions in its computation.

The above sequence of states must contain all states, since the automaton uses all left-to-

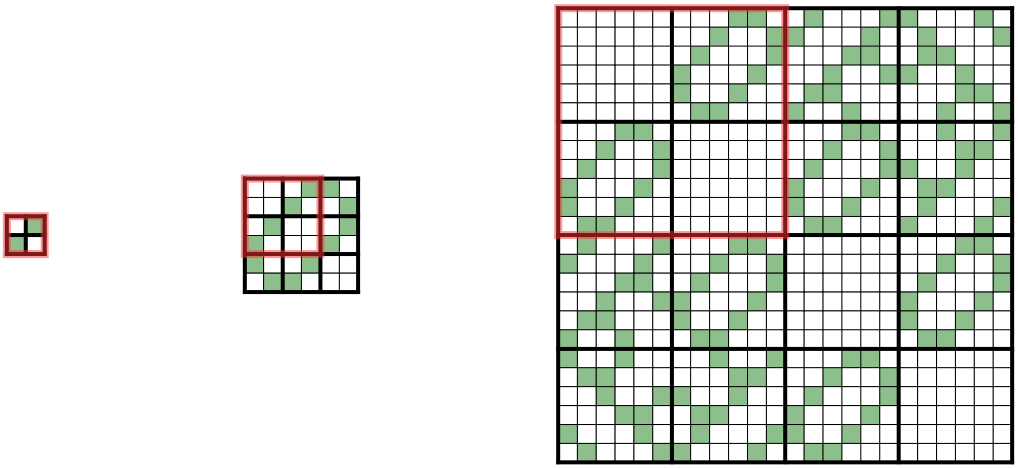


Figure 10: The matrices $P^{(2)}$, $P^{(3)}$ and $P^{(4)}$.

right arrows in its computation. Therefore, one can reach all states by applying $g_2 \circ g_1$ starting from k, and this exactly means that this permutation is cyclic.

$\Leftarrow$ Assuming that the permutation $g_2 \circ g_1$ is cyclic, the above sequence must contain all states, which means that all arrows in both directions are used. The sequence is concluded with a state q satisfying $g_2 \circ g_1(q) = k$, and then $\mathcal{D}_k$ accepts. Therefore, $P^{(k)}_{g_1,g_2} = 1$. □

The results obtained in this section can be summed up in the following theorem:

Theorem 2. *For every $n \geqslant 1$, there exists a language recognized by an n-state 2DFA, for which every UFA requires at least $\sum_{k=1}^{n} \binom{n}{k-1}\binom{n}{k} \operatorname{rank} P^{(k)}$ states, where $P^{(k)}$ is a $k! \times k!$ matrix, with its rows and columns corresponding to pertumations, and $P^{(k)}_{g_1,g_2} = 1$ if and only if $g_2 \circ g_1$ is a cyclic permutation.*

Proof. Consider the language described in the Section 4.1. It is recognized by an n-state 2DFA. By Theorem B, any UFA that recognizes this language should have at least rank $M^{(n)}$ states. By Lemma 5, the ranks of matrices $M^{(n)}$ and $L^{(n)}$ are equal. By Lemma 12, the same is true for $L^{(n)}$ and $K^{(n)}$. By Lemma 14, the rank of $K^{(n)}$ is equal to $\sum_{k=1}^{n} \binom{n}{k-1}\binom{n}{k} \operatorname{rank} P^{(k)}$. Finally, Lemma 15 provides an alternate definition of $P^{(k)}$. The statement of the theorem follows. □

The only thing left is to determine the rank of one particular matrix, $P^{(k)}$.

5 Estimating the rank of the matrix for permutations

5.1 Simple estimation of the rank

First, an easy lower bound can be obtained using purely combinatorial observations. The proof is based on an argument that $P^{(k)}$ has a submatrix $\left(\begin{smallmatrix} 0 & P^{(k-1)} \\ P^{(k-1)} & 0 \end{smallmatrix}\right)$, which can be observed in Figure 10.

Lemma 16. *Let k be an integer greater than 1. Let $g_1, g_2 : \{1, \ldots, k\} \to \{1, \ldots, k\}$ be two permutations such that $g_1(k) = g_2(k)$. Then, $P^{(k)}_{g_1,g_2^{-1}} = 0$.*

Proof. Assume otherwise, that $P^{(k)}_{g_1,g_2^{-1}} = 1$. Then, by Lemma 15, $g_2^{-1} \circ g_1$ is a cyclic permutation. However, $g_2^{-1} \circ g_1(k) = k$, so $g_2^{-1} \circ g_1$ contains a cycle of length $1 \neq k$, which leads to a contradiction. □

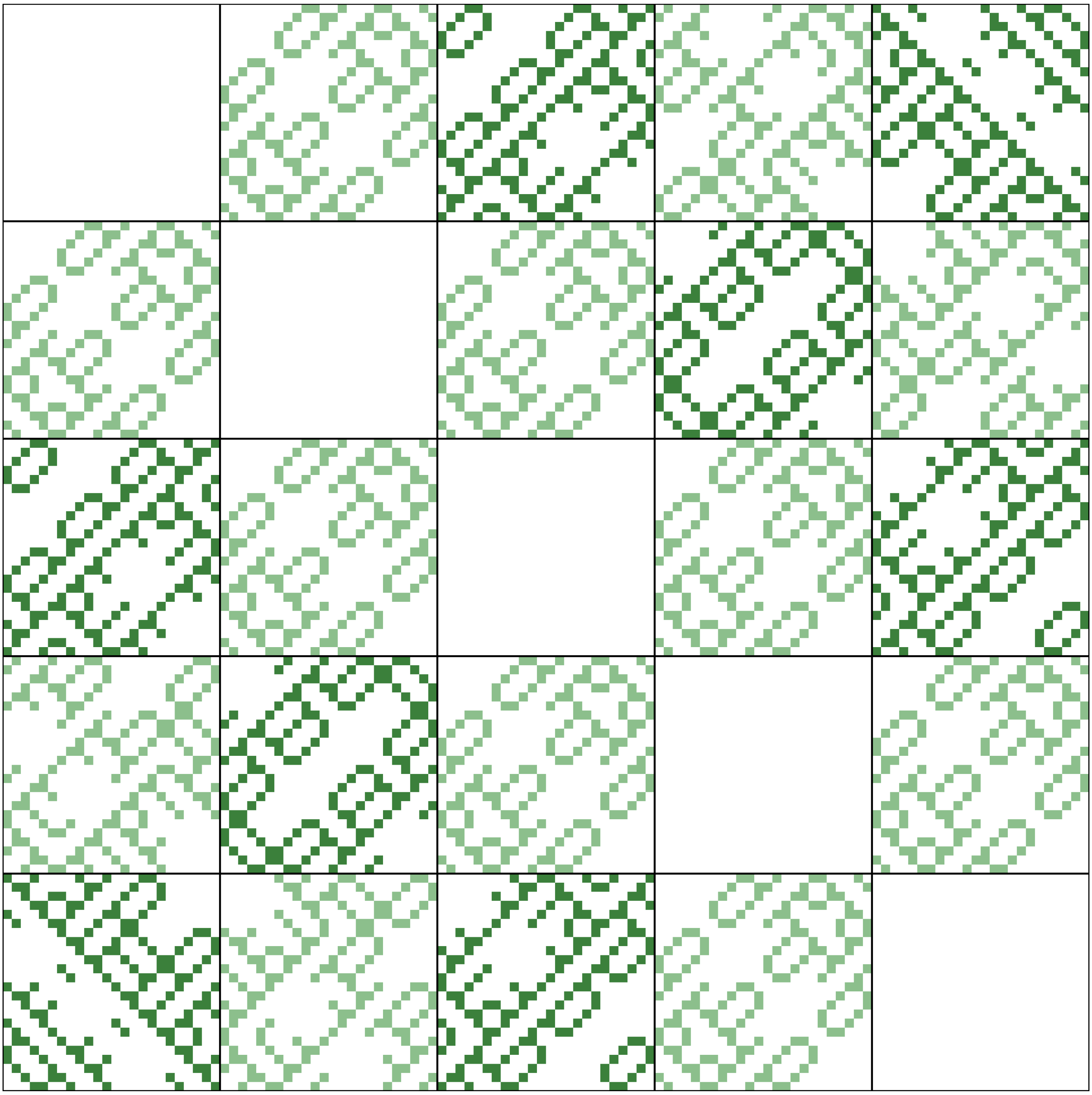


Figure 11: $P^{(5)}$.

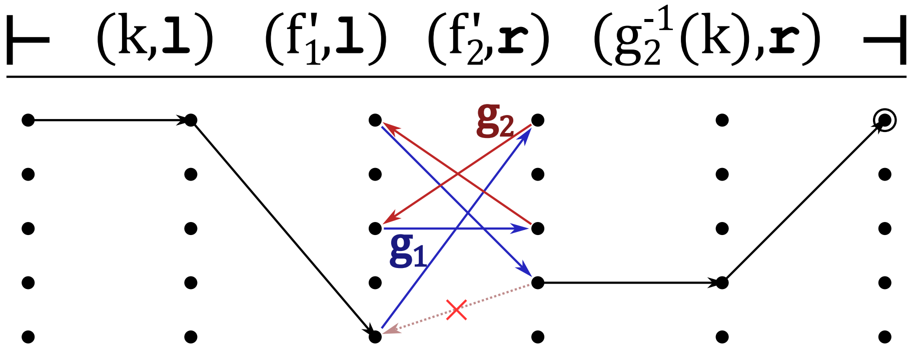


Figure 12: A computation of $\mathcal{D}_k$ on the string $x_{P,R,f_1}\, y_{P,R,f_2}$, where profiles (P,R,f_1) and (P,R,f_2) correspond to permutations g_1 and g_2^{-1}, respectively.

Lemma 17. *Let k be an integer. Let $g_1, g_2 : \{1,\ldots,k\} \to \{1,\ldots,k\}$ be two permutations. Then, $P^{(k)}_{g_1,g_2^{-1}} = P^{(k)}_{g_2,g_1^{-1}}$* (symmetry of the matrix for permutations).

Proof. An inverse of cyclic permutation is also cyclic. Hence, $g_2^{-1} \circ g_1$ is cyclic if and only if $(g_2^{-1} \circ g_1)^{-1} = g_1^{-1} \circ g_2$ is cyclic. Then, by Lemma 15, $P^{(k)}_{g_1,g_2^{-1}} = 1$ if and only if $P^{(k)}_{g_2,g_1^{-1}} = 1$, which means that $P^{(k)}_{g_1,g_2^{-1}} = P^{(k)}_{g_2,g_1^{-1}}$. □

Lemma 18. *Let k be an integer greater than 1. Let $g_1, g_2 : \{1,\ldots,k\} \to \{1,\ldots,k\}$ be two permutations such that $g_1(k) = k$, $g_2(k) = k-1$. Define two permutations $\tilde{g}_1, \tilde{g}_2 : \{1,\ldots,k-1\} \to \{1,\ldots,k-1\}$ as follows:*

$$\tilde{g}_1(p) = g_1(p)$$

$$\tilde{g}_2(p) = \begin{cases} g_2(p), & g_2(p) \neq k \\ k-1, & g_2(p) = k \end{cases}$$

Then, $P^{(k)}_{g_1,g_2^{-1}} = P^{(k-1)}_{\tilde{g}_1,\tilde{g}_2^{-1}}$.

Proof. The goal is to prove that $g_2^{-1} \circ g_1$ is cyclic if and only if $\tilde{g}_2^{-1} \circ \tilde{g}_1$ is cyclic. For that, the values of $\tilde{g}_2^{-1} \circ \tilde{g}_1$ shall be expressed through the values of $g_2^{-1} \circ g_1$.

Let $p \in \{1,\ldots,k-1\}$. Then, $g_1(p) = \tilde{g}_1(p)$, and $g_2^{-1}(p) = \tilde{g}_2^{-1}(p)$ if $p \neq k-1$. Hence, for every p such that $p \neq g_1^{-1}(k-1)$ it holds that $g_2^{-1} \circ g_1(p) = \tilde{g}_2^{-1} \circ \tilde{g}_1(p)$.

If $p = g_1^{-1}(k-1)$, then $\tilde{g}_2^{-1} \circ \tilde{g}_1(p) = \tilde{g}_2^{-1}(k-1) = g_2^{-1}(k) = g_2^{-1} \circ g_1(k) = g_2^{-1} \circ g_1 \circ g_2^{-1}(k-1) = g_2^{-1} \circ g_1 \circ g_2^{-1} \circ g_1(g_1^{-1}(k-1)) = g_2^{-1} \circ g_1 \circ g_2^{-1} \circ g_1(p)$, and two transitions of $g_2^{-1} \circ g_1$ collapse into one transition of $\tilde{g}_2^{-1} \circ \tilde{g}_1$.

Therefore, $g_2^{-1} \circ g_1$ is cyclic if and only if $\tilde{g}_2^{-1} \circ \tilde{g}_1$ is cyclic. Then, by Lemma 15, $P^{(k)}_{g_1,g_2^{-1}} = 1$ if and only if $P^{(k-1)}_{\tilde{g}_1,\tilde{g}_2^{-1}} = 1$, which means that $P^{(k)}_{g_1,g_2^{-1}} = P^{(k-1)}_{\tilde{g}_1,\tilde{g}_2^{-1}}$. □

Note that different permutations g_2 yield different $\tilde{g}_2$, since g_2 can be restored from $\tilde{g}_2$ by changing the value $k-1$ to k and adding $g_2(k) = k-1$.

The results are illustrated in Figure 10. By Lemma 16, the main diagonal contains k blocks of zeroes; by Lemma 17, the matrix is symmetric; and, by Lemma 18, the matrix $P^{(k)}$ contains blocks that are the same as $P^{(k-1)}$.

Lemma 19. *Let k be an integer greater than 1. Then,* $\operatorname{rank} P^{(k)} \geqslant 2 \cdot \operatorname{rank} P^{(k-1)}$.

Proof. Let M_k be a submatrix of the matrix $P^{(k)}$ that contains only rows corresponding to permutations g with $g(k) = k-1$ or $g(k) = k$, and columns corresponding to permutations g^{-1} with $g(k) = k-1$ or $g(k) = k$. Rows and columns for which $g(k) = k-1$, shall be called *small*, and others, *big*.

By Lemma 16, the matrix M_k consists of two blocks, since if a row and a column are both small or big, then their intersection contains 0. By Lemma 18, those two blocks are the same up to the permutation of rows and columns. Since those blocks have no rows or columns in common, the rank of M_k is a sum of ranks of those two blocks, which, by Lemma 18, are equal to $\operatorname{rank} P^{(k-1)}$. Hence, $\operatorname{rank} M_k = 2 \cdot \operatorname{rank} P^{(k-1)}$. Since $\operatorname{rank} M_k \leqslant \operatorname{rank} P^{(k)}$, the lemma is proved. □

Theorem 3. $\operatorname{rank} P^{(k)} \geqslant 2^{k-1}$.

Proof. The theorem is proved by induction on k. Base is $\operatorname{rank} P^{(1)} = 1$, and Lemma 19 gives the induction step. □

Corollary 1. *For every n, there is a language recognized by an n-state 2DFA, for which every UFA requires at least $\sum_{k=1}^{n} \binom{n}{k-1}\binom{n}{k} 2^{k-1} = \Omega\big(\frac{(4\sqrt{2})^n}{\sqrt{n}}\big)$ states.*

Proof. By Theorem B, for the $\mathcal{D}_n$ with n states, every UFA recognizing this language has at least $\operatorname{rank} M^{(n)}$ states. By Lemma 5, $\operatorname{rank} M^{(n)} = \operatorname{rank} L^{(n)}$. By Lemma 12, $\operatorname{rank} L^{(n)} = \operatorname{rank} K^{(n)}$. By Lemma 14, $\operatorname{rank} K^{(n)} = \sum_{k=1}^{n} \binom{n}{k-1}\binom{n}{k} \operatorname{rank} P^{(k)}$. Finally, by Theorem 3, $\operatorname{rank} P^{(k)} \geqslant 2^{k-1}$. Putting these together yields a lower bound in the form of a sum.

The desired asymptotic lower bound is obtained by casting away all summands with $k < \lceil \frac{n}{2} \rceil$, and by replacing 2^{k-1} with $2^{\lceil \frac{n}{2} \rceil - 1}$ in the remaining summands.

$$\sum_{k=1}^{n} \binom{n}{k-1}\binom{n}{k} 2^{k-1} \geq 2^{\lceil \frac{n}{2} \rceil - 1} \cdot \sum_{k=\lceil \frac{n}{2} \rceil}^{n} \binom{n}{k-1}\binom{n}{k}$$

Next, note that $\sum_{k=\lceil \frac{n}{2} \rceil}^{n} \binom{n}{k-1}\binom{n}{k} \geqslant \frac{1}{2} \sum_{k=1}^{n} \binom{n}{k-1}\binom{n}{k}$, because every k-th element of the latter sum is equal to its $(n-k+1)$-th element, as $\binom{n}{k-1}\binom{n}{k} = \binom{n}{n-k+1}\binom{n}{n-k}$. Then, the above sum is further estimated as follows.

$$2^{\lceil \frac{n}{2} \rceil - 1} \cdot \sum_{k=\lceil \frac{n}{2} \rceil}^{n} \binom{n}{k-1}\binom{n}{k} \geq 2^{\lceil \frac{n}{2} \rceil - 2} \cdot \sum_{k=1}^{n} \binom{n}{k-1}\binom{n}{k} =$$

$$= 2^{\lceil \frac{n}{2} \rceil - 2} \cdot \sum_{k=1}^{n} \binom{n}{k-1}\binom{n}{n-k} = 2^{\lceil \frac{n}{2} \rceil - 2} \binom{2n}{n-1}.$$

The asymptotics of the resulting expression are finally determined by Stirling's approximation.

$$2^{\lceil \frac{n}{2} \rceil - 2} \binom{2n}{n-1} = 2^{\lceil \frac{n}{2} \rceil - 2} \frac{n}{n+1} \binom{2n}{n} = \Omega\left(\sqrt{2}^n \cdot \frac{(2n)^{2n}\sqrt{2n}}{n^{2n}\sqrt{n}^2}\right) =$$

$$= \Omega\left(\sqrt{2}^n \cdot \frac{4^n}{\sqrt{n}}\right) = \Omega\left(\frac{(4\sqrt{2})^n}{\sqrt{n}}\right).$$

□

However, the actual rank of $P^{(k)}$ is higher. A better estimation is obtained using algebraic methods, by reformulating the problem in terms of group representation theory. A short background is presented below, and for more details, a reader is directed to a monograph by Sagan [16].

5.2 The matrix for permutations as an action on group algebra

For convenience, $P^{(k)}$ shall be replaced with another $k! \times k!$ matrix $Q^{(k)}$, which has its rows and columns indexed by permutations. The new matrix is defined by $Q^{(k)}_{g_1,g_2} = P^{(k)}_{g_1^{-1},g_2}$, for every two permutations g_1 and g_2; that is, $Q^{(k)}_{g_1,g_2} = 1$ if $g_2 \circ g_1^{-1}$ is cyclic. In order to determine its rank, the matrix shall be represented as an action on a group algebra.

Let S_k be the group of permutations on k elements. Consider the *group algebra* $\mathbb{C}[S_k] = \mathbb{C}^{S_k}$: this is a linear space over the complex numbers, with coordinates corresponding to permutations, and with the following sum and composition operations.

$$\Big(\sum_{p \in S_k} a_p p\Big) + \Big(\sum_{q \in S_k} b_q q\Big) = \sum_{p \in S_k} (a_p + b_p) p$$
$$\Big(\sum_{p \in S_k} a_p p\Big) \circ \Big(\sum_{q \in S_k} b_q q\Big) = \sum_{p,q \in S_k} a_p b_q (p \circ q)$$

Let $C_k \subseteq S_k$ be the set of cyclic permutations, and denote their sum by $q_k = \sum_{p \in C_k} p$.

Lemma 20. $Q^{(k)}_{g_1,g_2} = 1$ *if and only if there exists a cyclic permutation* r *such that* $r \circ g_1 = g_2$.

Proof. By the definition of $Q^{(k)}$, the equality $Q^{(k)}_{g_1,g_2} = 1$ is equivalent to $P^{(k)}_{g_1^{-1},g_2} = 1$. By Lemma 15, $P^{(k)}_{g_1^{-1},g_2} = 1$ is equivalent to existence of such cyclic permutation r, that $g_2 \circ g_1^{-1} = r$. This is equivalent to $r \circ g_1 = g_2$. □

Lemma 21. *Let* g_1, g_2 *be permutations on* k *elements. If there exists a cyclic permutation* r *such that* $r \circ g_1 = g_2$, *then* r *is unique.*

Proof. Since $r \circ g_1 = g_2$, then $r = g_2 \circ g_1^{-1}$. Therefore, r is unique. □

Lemma 22. *The matrix* $Q^{(k)}$ *is a matrix of left multiplication by the sum of cyclic permutations* q_k *in* $\mathbb{C}[S_k]$.

Proof. By Lemma 20, $Q^{(k)}_{g_1,g_2} = 1$ if and only if there exists a cyclic permutation r such that $r \circ g_1 = g_2$. By Lemma 21, such permutation r is unique. □

Multiplying by the sum of cyclic permutations on the left is an *action on* $\mathbb{C}[S_k]$. Its image forms a subspace, and its dimension is the rank of the matrix $Q^{(k)}$, which is equal to the rank of $P^{(k)}$. It remains to determine this dimension.

5.3 Representations and modules

The argument is based on representations of the group $G = S_k$. In this work, representations shall be given in the notation of vector spaces, and are known as *modules*.

Definition 13. For a group G, a G-module is any vector space V over the complex numbers, with a homomorphism $\varphi \colon G \to GL(V)$, where $GL(V)$ is the set of all invertible linear mappings $V \to V$ with a composition operation defined on them. The function φ then describes the action of the elements of G on the elements of V.

For a G-module (V, φ), the function φ is naturally extended to the group algebra $\mathbb{C}[G]$ by setting $\varphi\big(\sum_{g \in G} a_g g\big)(v) = \sum_{g \in G} a_g \varphi(g)(v)$ for every vector $v \in V$. Note that $\varphi\big(\sum_{g \in G} a_g g\big)$ is not necessarily invertible. In the following, this extension of φ shall always be assumed.

Definition 14. A G-module (V, φ) is called *reducible*, if there exists a non-trivial proper subspace $W \subset V$ invariant to the action of elements of G. Otherwise, the module is *irreducible.*

Theorem C (Maschke, see Sagan [16, Thm. 1.5.3])**.** *Let G be a group, and let (V, φ) be a G-module. Then there is a number d and irreducible modules $W^{(1)}, \dots, W^{(d)}$, such that V is a direct sum of these vector spaces: $V = W^{(1)} \oplus \dots \oplus W^{(d)}$.*

By Maschke's theorem, the group algebra $\mathbb{C}[S_k]$ is decomposed into a direct sum of irreducible modules.

Theorem D ([16, Prop. 1.10.1])**.** *Every irreducible S_k-module occurs in the decomposition of the group algebra $\mathbb{C}[S_k]$ as many times, up to isomorphism, as its dimension.*

Therefore, it is sufficient to consider the product by the sum of cyclic permutations in *irreducible* S_k-modules.

5.4 The action of the sum of cyclic permutations on irreducible modules

An element g of a ring R is called *central*, if $gh = hg$ for every $h \in R$. Since $\mathbb{C}[S_k]$ is a ring, it may have central elements.

Lemma 23. *The sum of cyclic permutations, q_k, is a central element of $\mathbb{C}[S_k]$.*

Proof. Note that it is enough to prove that $p \circ q_k = q_k \circ p$ for every permutation $p \in S_k$, since the elements of $\mathbb{C}[S_k]$ are linear combinations of those permutations.

Let $p \in S_k$ be a permutation, and let $g \in C_k$ be a cyclic permutation. Define $h = p^{-1} \circ g \circ p$. Then, $p \circ h = g \circ p$ by definition. Since g and h are conjugate in the group S_k, their cycle types are the same, and h is also a cyclic permutation. Note that different g yield different h, since there exists a reverse representation $g = p \circ h \circ p^{-1}$.

Then, $q_k \circ p = \sum_{g \in C_k} g \circ p = \sum_{h \in C_k} p \circ h = p \circ q_k$, and the lemma is proved. □

Lemma 24. *Let G be a group. Let $g \in G$ be a central element of group algebra $\mathbb{C}[G]$. Let (V, φ) be an irreducible G-module. Then, there is a constant $\lambda \in \mathbb{C}$, such that $\varphi(g)v = \lambda v$ for every element $v \in V$.*

Proof. Let λ be an eigenvalue of $\varphi(g)$. Define $V_\lambda = \{v \in V \mid \varphi(g)v = \lambda v\}$. Since λ is an eigenvalue of $\varphi(g)$, then V_λ is non-trivial. If $V_\lambda = V$, then the lemma is proved.

Suppose then, that V_λ is a proper subspace of V. The goal is to prove that V_λ is invariant to the action of the elements of G: that is, for every $h \in G$ and for every $v \in V_\lambda$ the product $\varphi(h)v$ is also in V_λ.

For $h \in G$ and $v \in V_\lambda$, $\varphi(g) \cdot \varphi(h)v = [\varphi \text{ is a homomorphism}] = \varphi(gh)v = [g \text{ is central}] = \varphi(hg)v = \varphi(h)\varphi(g)v = \varphi(h)\lambda v = \lambda \cdot \varphi(h)v$, and $\varphi(h)v \in V_\lambda$.

Since V_λ is a non-trivial proper subspace of V that is invariant to the action of the elements of G, the module (V, φ) is reducible, which leads to a contradiction. □

Lemma 25. *Let (V, φ) be an irreducible S_k-module. Then, there is a constant $\lambda \in \mathbb{C}$, such that $\varphi(q_k)v = \lambda v$ for every element $v \in V$.*

In other words, in irreducible modules, the sum of cyclic permutations acts as multiplication by a number. If that number is non-zero, the action is full-rank; if it is zero, the rank shall be zero as well.

Proof. By Lemma 23, the element q_k is central in $\mathbb{C}[S_k]$. The result of the lemma follows from Lemma 24 for $G = S_k$ and $g = q_k$. □

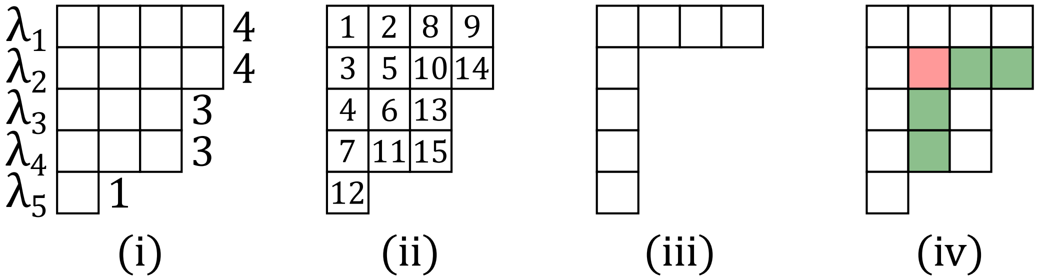


Figure 13: (i) Young diagram for $\lambda = (4, 4, 3, 3, 1)$; (ii) a standard Young tableau; (iii) a Young diagram of a hook shape, with $\lambda = (4, 1, 1, 1, 1)$; (iv) a hook of length 5 in a Young diagram.

5.5 Specht modules

The form of irreducible S_k-modules is well-studied, they are known as **Specht modules**.

Let $k \geqslant 1$ be an integer, and let $\lambda = (\lambda_1, \ldots, \lambda_m)$ be a partition, with $k = \lambda_1 + \ldots + \lambda_m$ and $\lambda_1 \geqslant \ldots \geqslant \lambda_m \geqslant 1$. Every partition has a corresponding *Young diagram* $Y(\lambda)$, as in Figure 13(i), which consists of m rows, with each i-th row of length λ_i.

Definition 15. A *Young tableau* of size k is a Young diagram, with its boxes filled with the elements $\{1, \ldots, k\}$, with each element occurring exactly once. A tableau is *standard*, if the numbers in every row and the numbers in every column increase.

A standard Young tableau is illustrated in Figure 13(ii).

The action of a permutation in S_k on a Young tableau of size k is defined by applying the permutation to the element in every box.

Definition 16. Two Young tableaux are *equivalent*, if the sets of values in the corresponding rows are the same. The equivalence class of a tableau T is called a *tabloid* and is denoted by $\{T\}$. The action of a permutation on a tabloid is defined as $g \cdot \{T\} = \{gT\}$.

To establish the correctness of the action of permutations on tabloids, the following condition needs to be satisfied.

Lemma 26. *Let g be a permutation. Let T and T' be two equivalent Young tableaux. Then, gT and gT' are also equivalent.*

Proof. Fix a row of Young tableaux T and T'. Since these two tableaux are equivalent, they contain the same set of elements S in this row. Then, gT and gT' also contain the same set of elements $g(S) = \{g(s) \mid s \in S\}$ in this row.

Since this holds for every row, gT and gT' are equivalent. □

For a Young tableau T, let $C(T)$ be the set of permutations that preserve the set of values in each column of T.

Let Y_λ be a vector space, in which the coordinates correspond to tabloids.

Definition 17. A *polytabloid* corresponding to a Young tableau T is a linear combination $e_T = \sum_{\pi \in C(T)} \operatorname{sgn}(\pi) \cdot \{\pi T\}$ in the vector space Y_λ.

The irreducible S_k-modules under concern are the *Specht modules.*

Definition 18. A *Specht module* for a partition λ of a number k is the module $S^{(\lambda)} = \{ \sum c_T e_T \mid c_T \in \mathbb{C} \}$, where the sum is taken over all Young tableaux T corresponding to the partition λ. A Specht module is a subspace of Y_λ.

Theorem E ([16, Thm. 2.4.6])**.** *Specht modules are irreducible, every two distinct Specht modules are non-isomorphic, and every irreductible S_k-module is isomorphic to some Specht module.*

5.6 The action of the sum of cyclic permutations on Specht modules

What is **the action of the sum of cyclic permutations on Specht modules?** The question is, for which Young tableaux T the action of the sum $q_k e_T$ is non-trivial, in the sense that it is not constant zero. This action is expressed as follows.

Lemma 27. *$q_k e_T = \sum_{r \in C_k} \left(\sum_{\pi \in C(T)} \operatorname{sgn}(\pi) \cdot \{\pi(rT)\} \right)$ for every Young tableau T.*

Proof. By Lemma 23, q_k is a central element of $\mathbb{C}[S_k]$. In particular, $q_k \circ \pi = \pi \circ q_k$ for every permutation $\pi \in S_k$. Using that, the action of the sum of cyclic permutations on the polytabloid e_T is calculated as follows.

$$q_k e_T = q_k \cdot \sum_{\pi \in C(T)} \operatorname{sgn}(\pi) \cdot \{\pi T\} = \sum_{\pi \in C(T)} \operatorname{sgn}(\pi) q_k \circ \pi\{T\} = \sum_{\pi \in C(T)} \operatorname{sgn}(\pi) \pi \circ q_k\{T\} =$$
$$= \sum_{\pi \in C(T)} \operatorname{sgn}(\pi)\pi \left(\sum_{r \in C_k} r \cdot \{T\} \right) = \sum_{\pi \in C(T)} \operatorname{sgn}(\pi)\pi \left(\sum_{r \in C_k} \{rT\} \right) =$$
$$= \sum_{r \in C_k} \left(\sum_{\pi \in C(T)} \operatorname{sgn}(\pi)\pi \cdot \{rT\} \right) = \sum_{r \in C_k} \left(\sum_{\pi \in C(T)} \operatorname{sgn}(\pi) \cdot \{\pi(rT)\} \right).$$

□

This is a sum of expressions of the form $\sum_{\pi \in C(T)} \operatorname{sgn}(\pi) \cdot \{\pi(rT)\}$, akin to the expression in the definition of the polytabloid. It turns out that, if the same tabloid $\{rT\}$ could be obtained using a column-preserving permutation $g \in C(T)$, then an expression of the above form is actually equal to the polytabloid, up to the sign.

Lemma 28. *Let T be a Young tableau, and let $r \in S_k$ be a permutation. Then, if there is a permutation $g \in C(T)$ satisfying $\{gT\} = \{rT\}$, then $\sum_{\pi \in C(T)} \operatorname{sgn}(\pi) \cdot \{\pi(rT)\} = \operatorname{sgn}(g) e_T$. If there is no such g, then $\sum_{\pi \in C(T)} \operatorname{sgn}(\pi) \cdot \{\pi(rT)\} = 0$.*

Proof. Denote $T' = rT$.

1. Suppose that there is a permutation $g \in C(T)$ satisfying $\{gT\} = \{rT\}$. Then, the following is true:

$$\begin{aligned}
\sum_{\pi \in C(T)} \operatorname{sgn}(\pi) \cdot \{\pi T'\} &= \sum_{\pi \in C(T)} \operatorname{sgn}(\pi)\pi \cdot \{T'\} \\
&= \sum_{\pi \in C(T)} \operatorname{sgn}(\pi)\pi \cdot \{gT\} \\
&= \sum_{\pi \in C(T)} \operatorname{sgn}(\pi \circ g \circ g^{-1})\pi \circ g\{T\} \\
&= \operatorname{sgn}(g^{-1}) \sum_{\pi \in C(T)} \operatorname{sgn}(\pi \circ g)\pi \circ g\{T\} \\
&= [\pi \in C(T) \Leftrightarrow \pi \circ g \in C(T), \\
&\text{since } C(T) \text{ is closed under composition}] = \\
&= \operatorname{sgn}(g) \sum_{\pi \circ g \in C(T)} \operatorname{sgn}(\pi \circ g)\pi \circ g\{T\} \\
&= \operatorname{sgn}(g) e_T.
\end{aligned}$$

2. Suppose that there is no permutation $g \in C(T)$ such that $\{gT\} = \{rT\}$.

   Then, there exists a pair of numbers (x_1, x_2), that are located in the same column of T and in the same row of T'. Suppose, for the sake of a contradiction, that there is no such pair. Then, all elements of the first column of T must belong to pairwise distinct rows of T'. Construct a permutation of the first column of T that would place all elements into their rows in T'. Next, in T', these elements are shifted into the first column, forming an equivalent tableau with its first column identical to that in the modified T. This process is repeated with every next column, and overall, it affects the tableaux as follows: elements in columns of T are permuted by some permutation g, whereas elements in rows of T' are permuted by some g'. Then, $gT = g'T'$, and since g' preserves the sets of elements in the rows of T', the tableaux T' and $g'T'$ are equivalent. Therefore, $\{gT\} = \{g'T'\} = \{T'\} = \{rT\}$, which contradicts the assumption.

   It has thus been proved that there is a pair of elements (x_1, x_2) that share the same column in T and the same row in T'. Denote by $(x_1\,x_2)$ the permutation that swaps x_1 and x_2, and leaves other elements on their places. Note that $\{(x_1\,x_2)T'\} = \{T'\}$, since x_1 and x_2 are located in the same row of T', and $(x_1\,x_2) \in C(T)$, since x_1 and x_2 are located in the same column of T. Then, the following is true:

$$\begin{aligned}
\sum_{\pi \in C(T)} \operatorname{sgn}(\pi) \cdot \{\pi T'\} &= \sum_{\pi \circ (x_1\,x_2) \in C(T)} \operatorname{sgn}(\pi \circ (x_1\,x_2)) \cdot \{(\pi \circ (x_1\,x_2))T'\} \\
&= [\pi \in C(T) \Leftrightarrow \pi \circ (x_1\,x_2) \in C(T), \\
&\text{since } C(T) \text{ is closed under composition}] = \\
&= \sum_{\pi \in C(T)} \operatorname{sgn}(\pi \circ (x_1\,x_2)) \cdot \{(\pi \circ (x_1\,x_2))T'\} \\
&= \sum_{\pi \in C(T)} \operatorname{sgn}(\pi) \operatorname{sgn}((x_1\,x_2))\pi \cdot \{(x_1\,x_2)T'\} \\
&= -\sum_{\pi \in C(T)} \operatorname{sgn}(\pi)\pi \cdot \{T'\} \\
&= -\sum_{\pi \in C(T)} \operatorname{sgn}(\pi) \cdot \{\pi T'\}.
\end{aligned}$$

   Since this sum is equal to its additive inverse, it is equal to 0.

□

Accordingly, all non-zero summands in the expression in Lemma 27 are of the form $\pm e_T$, and the question is, how the number of positive summands compares to the number of negative summands.

5.7 The sum of cyclic permutations on hook-shaped tableaux

The overall sum in Lemma 27 turns out to be non-zero for Young tableaux of a special **hook shape**, illustrated in Figure 13(iii), in which all rows, possibly except the first row, are of length 1.

Lemma 29. *Let T be a hook-shaped Young tableau of shape λ. Let $r \in C_k$ and $g \in C(T)$ be such that $\{gT\} = \{rT\}$. Then g is a cycle on the first column of T. Accordingly, as there are $|\lambda|$ elements in the first column,* $\operatorname{sgn}(g) = (-1)^{|\lambda|-1}$.

Proof. As $\{gT\} = \{rT\}$, then $\{T\} = \{(g^{-1} \circ r)T\}$ by Lemma 26. Then, by the definition of equivalence of Young tableaux, $g^{-1} \circ r$ acts as a permutation on each row of T, and, in particular, $g^{-1}(r(x)) = x$ for each number x not in the first row of T. This means that $r(x) = g(x)$.

Suppose that g is not a cycle on the first column of T. Then, since g is in $C(T)$, it acts on the first column as multiple cycles, and, accordingly, must have a cycle on a subset of the first column that does not contain the corner element. Since this cycle contains no elements of the first row, the equality $r(x) = g(x)$ holds for each element of the cycle. Then the same cycle is also in r, and this contradicts the assumption that r is cyclic. □

Lemma 30. *Let T be a Young tableau of shape $\lambda = (k - m + 1, 1, \dots, 1)$. Then, $q_k e_T = c_\lambda \cdot e_T$ for some complex constant $c_\lambda \neq 0$ that depends on λ only.*

Proof. By Lemma 25, the sum of cyclic permutations acts on irreducible S_k-modules as multiplication by a constant, and it is left to prove that c_λ is non-zero.

The expression on the left is transformed by Lemma 27 as $q_k e_T = \sum_{r \in C_k} \left(\sum_{\pi \in C(T)} \operatorname{sgn}(\pi) \cdot \{\pi(rT)\} \right)$. By Lemma 28 expressions of the form $\sum_{\pi \in C(T)} \operatorname{sgn}(\pi) \cdot \{\pi(rT)\}$, with $r \in C_k$, evaluate either to zero or to $\operatorname{sgn}(g) e_T$, where $g \in C(T)$ is a permutation satisfying $\{gT\} = \{rT\}$. Since T is hook-shaped, in the latter case, by Lemma 29, the value of the expression is $\operatorname{sgn}(g) e_T = (-1)^{|\lambda|-1} e_T$, and it does not depend on the permutation r.

Accordingly, all non-zero summands in $\sum_{r \in C_k} \left(\sum_{\pi \in C(T)} \operatorname{sgn}(\pi) \cdot \{\pi(rT)\} \right)$ are of the same form $(-1)^{|\lambda|-1} e_T$. In order to prove that the entire sum is non-zero, it is sufficient to find a single r, for which the corresponding summand is non-zero.

Denote by $T_{i,j}$ the j-th element of the i-th row. A cyclic permutation r is constructed as follows, with the value listed in the order of the cycle traversal. It shifts all elements of the first row to the right ($r(T_{1,j}) = T_{1,j+1}$ for $j \in \{1, \dots, \lambda_1 - 1\}$) and maps the last element to the second row ($r(T_{1,|\lambda_1|}) = T_{2,1}$). The rest of the first column is shifted downwards ($r(T_{i,1}) = T_{i+1,1}$ for $i \in \{2, \dots, |\lambda| - 1\}$), and the bottom element returns to the corner ($r(T_{|\lambda|,1}) = T_{1,1}$).

It remains to find such a permutation $g \in C(T)$, that $\{gT\} = \{rT\}$. It is defined as a cycle on the first column, with $g(T_{i,1}) = T_{i+1,1}$ for $i \in \{1, \dots, |\lambda| - 1\}$ and $g(T_{|\lambda|,1}) = T_{1,1}$, and as a loop on each of the remaining elements: $g(T_{1,j} = T_{1,j})$ for $j \in \{2, \dots, \lambda_1\}$. In order to show that $\{gT\} = \{rT\}$, by Definition 16, one has to verify that, for each row of T, the sets of values of g and r on that row are the same. Beginning with the second row, this is true, since g and r have the same value on those elements. Since all remaining elements are in the first row, the sets of values are the same also for the first row.

Since a permutation $g \in C(T)$ satisfying $\{gT\} = \{rT\}$ has been found, by Lemma 28, the summand $\sum_{\pi \in C(T)} \operatorname{sgn}(\pi) \cdot \{\pi(rT)\}$ is non-zero, and the entire sum is non-zero as a multiple of this summand. □

These special shapes contribute to the rank of $P^{(k)}$ along with other possible partitions of k, as proved in the following lemma.

Lemma 31. *Let Λ be the set of all partitions λ of the number k, for which the action of q_k on $S^{(\lambda)}$ is a multiplication by a non-zero constant. Then,* $\operatorname{rank} P^{(k)} = \sum_{\lambda \in \Lambda} (\dim S^{(\lambda)})^2$.

Proof. By Theorem D, every irreducible module of dimension d occurs in the decomposition of $\mathbb{C}[S_k]$ into a direct sum d times up to isomorphism. It follows from Lemma 25 that the action of the sum of cyclic permutations q_k on an irreducible module either has full rank d, or its rank is zero. Hence, every irreducible module of dimension d, on which the action of q_k is non-trivial, contributes $d \cdot d$ to the sum of the ranks. According to Theorem E, irreducible modules are exactly the Specht modules. □

5.8 Dimensions of Specht modules

The exact contribution of hook-shaped partitions to the rank of $P^{(k)}$ shall be determined using the following known result on **the dimension of Specht modules**.

Theorem F ([16, Thm. 2.6.5]). *Polytabloids corresponding to standard Young tableaux for a partition λ form the basis of the Specht module $S^{(\lambda)}$.*

Accordingly, the dimension of a Specht module coincides with the number of standard Young tableaux for a given partition. This number is given by *the hook formula*.

For a box x in a Young diagram, its *hook length*, $\operatorname{hook}(x)$, is the number of squares to the right of x in the same row, below x in the same column, plus one (for x itself). An example of a hook is given in Figure 13(iv).

Theorem G ([16, Thm. 3.10.2]). *For a Young diagram $Y(\lambda)$ of size k, the number of its standard fillings is $\frac{k!}{\prod_{x \in Y(\lambda)} \operatorname{hook}(x)}$.*

Corollary 2. $\dim(S^{(\lambda)}) = \frac{k!}{\prod_{x \in Y(\lambda)} \operatorname{hook}(x)}$

5.9 Lower bound on the rank of the matrix for permutations

Using the above results, the rank of the desired matrix is estimated as follows.

Theorem 4. *The rank of the $k! \times k!$ matrix $P^{(k)}$ is at least $\binom{2k-2}{k-1}$.*

Proof. By Lemma 31, the rank of $P^{(k)}$ is the sum of squares of dimensions of Specht modules S^{λ}, on which the action of q_k is non-trivial. It follows from Lemma 30 that the action of q_k on hooks is non-trivial. With k fixed, a hook is determined by the size λ_1 of its first row. Then, there are $m = k + 1 - \lambda_1$ elements in the first column.

In order to determine the dimension of the representation corresponding to a hook-shaped Specht module, by Corollary 2, one should determine the lengths of all hooks contained therein. For the box at the intersection of the first row and the first column, the hook length is k. For an i-th box in the first row, with $i > 1$, the hook length is $\lambda_1 - i + 1$, and the product of all these expressions is $(\lambda_1 - 1)!$. Similarly, for a j-th box in the first column, with $j > 1$, the hook length is $m - j + 1 = k + 1 - \lambda_1 - j + 1$ and their product is $(k - \lambda_1)!$.

Overall, $\dim(S^{(\lambda)}) = \frac{k!}{k \cdot (\lambda_1 - 1)! \cdot (k - \lambda_1)!} = \binom{k-1}{\lambda_1 - 1}$. Then, $\operatorname{rank} P^{(k)} \geqslant \sum_{\lambda_1 = 1}^{k} \binom{k-1}{\lambda_1 - 1}^2 = \sum_{a=0}^{k-1} \binom{k-1}{a}^2$. The latter sum gives the number of ways to choose a among $k-1$ elements and $k - 1 - a$ among $k - 1$ other elements. This is the same as choosing $k - 1$ among $2k - 2$ elements, and thus the lower bound obtained is $\binom{2k-2}{k-1}$. □

5.10 The asymptotics of the lower bound

Theorem 5. *For every n, there is a language recognized by a 2DFA with n states, such that every UFA for the same language requires at least $\sum_{k=1}^{n} \binom{n}{k-1}\binom{n}{k}\binom{2k-2}{k-1} = \Omega\big(\frac{9^n}{n^{3/2}}\big)$ states.*

Proof. By Theorem B, for the $\mathcal{D}_n$ with n states, every UFA recognizing this language has at least $\operatorname{rank} M^{(n)}$ states. By Lemma 5, $\operatorname{rank} M^{(n)} = \operatorname{rank} L^{(n)}$. By Lemma 12, $\operatorname{rank} L^{(n)} = \operatorname{rank} K^{(n)}$. By Lemma 14, $\operatorname{rank} K^{(n)} = \sum_{k=1}^{n} \binom{n}{k-1}\binom{n}{k} \operatorname{rank} P^{(k)}$. Finally, by Theorem 4, $\operatorname{rank} P^{(k)} \geqslant \binom{2k-2}{k-1}$. The above facts are combined into the desired estimation in the form of sum.

An asymptotic bound on this sum is obtained by estimating a single term, assuming $k = \alpha n$, for a fixed α with $0 < \alpha < 1$.

$$\sum_{k=1}^{n} \binom{n}{k-1}\binom{n}{k}\binom{2k-2}{k-1} \geqslant \binom{n}{\alpha n-1}\binom{n}{\alpha n}\binom{2\alpha n-2}{\alpha n-1} =$$

$$= \frac{n!}{(\alpha n-1)!((1-\alpha)n+1)!} \cdot \frac{n!}{(\alpha n)!((1-\alpha)n)!} \cdot \frac{(2\alpha n-2)!}{(\alpha n-1)!(\alpha n-1)!} =$$

$$= \frac{\alpha n \cdot \alpha n \cdot \alpha n}{((1-\alpha)n+1)\cdot(2\alpha n-1)\cdot 2\alpha n} \cdot \frac{n!n!(2\alpha n)!}{(\alpha n)!((1-\alpha)n)!(\alpha n)!((1-\alpha)n)!(\alpha n)!(\alpha n)!} \sim$$

$$\sim \frac{\alpha}{4(1-\alpha)} \cdot \frac{n!n!(2\alpha n)!}{(\alpha n)!((1-\alpha)n)!(\alpha n)!((1-\alpha)n)!(\alpha n)!(\alpha n)!}.$$

The factorials are then estimated using Stirling's approximation.

$$\frac{n!n!(2\alpha n)!}{(\alpha n)!((1-\alpha)n)!(\alpha n)!((1-\alpha)n)!(\alpha n)!(\alpha n)!} =$$

$$= \Omega\left(\frac{n^{2n}\cdot(2\alpha n)^{2\alpha n}}{(\alpha n)^{4\alpha n}\cdot((1-\alpha)n)^{2(1-\alpha)n}} \cdot \frac{\sqrt{n}^2\cdot\sqrt{2\alpha n}}{\sqrt{\alpha n}^4\cdot\sqrt{(1-\alpha)n}^2}\right) = \Omega\left(\frac{(2\alpha)^{2\alpha n}}{\alpha^{4\alpha n}\cdot(1-\alpha)^{2(1-\alpha)n}}\cdot\frac{n^{\frac{3}{2}}}{n^3}\right) =$$

$$= \Omega\left(\frac{2^{2\alpha n}}{\alpha^{2\alpha n}\cdot(1-\alpha)^{2(1-\alpha)n}}\cdot\frac{1}{n^{\frac{3}{2}}}\right) = \Omega\left(\left(\frac{2^{\alpha}}{\alpha^{\alpha}\cdot(1-\alpha)^{1-\alpha}}\right)^{2n}\cdot\frac{1}{n^{\frac{3}{2}}}\right).$$

The function $f(\alpha) = \frac{2^{\alpha}}{\alpha^{\alpha}\cdot(1-\alpha)^{1-\alpha}}$ has its maximum at $\alpha = \frac{2}{3}$, and its value is calculated as follows.

$$\frac{2^{\alpha}}{\alpha^{\alpha}\cdot(1-\alpha)^{1-\alpha}} = \frac{2^{\frac{2}{3}}}{\left(\frac{2}{3}\right)^{\frac{2}{3}}\cdot\left(\frac{1}{3}\right)^{\frac{1}{3}}} = \left(\frac{2^2}{\left(\frac{2}{3}\right)^2\cdot\left(\frac{1}{3}\right)}\right)^{\frac{1}{3}} = \left(\frac{2^2\cdot 3^3}{2^2}\right)^{\frac{1}{3}} = 3$$

Finally, this bound is substituted into the entire sum, leading to the desired lower bound.

$$\sum_{k=1}^{n} \binom{n}{k-1}\binom{n}{k}\binom{2k-2}{k-1} \geqslant \frac{\alpha}{4(1-\alpha)}\cdot\Omega\left(\left(\frac{2^{\alpha}}{\alpha^{\alpha}\cdot(1-\alpha)^{1-\alpha}}\right)^{2n}\cdot\frac{1}{n^{\frac{3}{2}}}\right) =$$

$$= \Omega\left(3^{2n}\cdot\frac{1}{n^{\frac{3}{2}}}\right) = \Omega\left(\frac{9^n}{n^{\frac{3}{2}}}\right)$$

□

6 Optimality of the lower bound

There is a companion result that it is not possible to achieve a better lower bound via Schmidt's theorem by choosing another 2DFA and different pairs of strings. In fact, even stronger result is true:

Theorem 6. *Let $\mathcal{D}$ be a 2UFA over an alphabet Γ with n states that recognizes a regular language L. Let $X = \{x_1, \ldots, x_\ell\}$ and $Y = \{y_1, \ldots, y_m\}$ be sets of strings over the alphabet Γ. Let M be an $\ell \times m$ matrix defined by $M_{i,j} = 1$ if $x_i y_j \in L$, and $M_{i,j} = 0$ otherwise. Then,* $\operatorname{rank} M \leqslant \sum_{k=1}^{n} \binom{n}{k-1}\binom{n}{k} \operatorname{rank} P^{(k)}$.

The proof of this theorem consists of several parts. Firstly, matrix M is defined in terms of profiles. Secondly, linear combinations based on the inclusion-exclusion principle are applied, yielding a new matrix with the same rank. Finally, it is shown that the rank of this new matrix does not exceed the rank of $K^{(n)}$, concluding the proof.

6.1 Conversion to extended profiles

Let $Q = \{q_1, \ldots, q_n\}$ be the set of states of the 2UFA $\mathcal{D}$. Let $Q_0 \subseteq Q$ be the set of starting states of $\mathcal{D}$.

In order to analyze the computations of $\mathcal{D}$ on all concatenations $x_i y_j$, it is necessary to consider all possible computations on prefixes x_i and on suffixes y_j. This requires a variant of the notion of a profile, called an *extended profile*, which represents all possible computations on a prefix or on a suffix beginning *in all states*, rather than only in the states used in an actual computation, as per the definition of a profile.

For each prefix x of a possible input string, the *left extended profile* on x represents the computations of $\mathcal{D}$ on the input $\vdash x$.

Definition 19. Let $x \in \Gamma^*$ be a string. A left extended profile on x is a function $f_x \colon \{1, \ldots, n\} \cup \{\textsc{Start}\} \to 2^{\{1,\ldots,n\}}$ representing the computations of $\mathcal{D}$ on the input $\vdash x$ as follows. For each $1 \leqslant i \leqslant n$ and for each $1 \leqslant j \leqslant n$, the element j is present in $f_x(i)$ if and only if there exists a computation that starts on the last symbol of $\vdash x$ in the state q_i, and first moves to the right beyond this last symbol in the state q_j. Also, for each $1 \leqslant j \leqslant n$, the element j is present in $f_x(\textsc{Start})$ if and only if there exists a computation that starts on the first symbol of $\vdash x$ in the state from Q_0, and first moves to the right beyond the last symbol of this string in the state q_j.

Similarly, the computations of $\mathcal{D}$ on a suffix $y\dashv$ are represented by a *right extended profile* on y.

Definition 20. Let $y \in \Gamma^*$ be a string. A right extended profile on y is a function $g_y \colon \{1, \ldots, n\} \to 2^{\{1,\ldots,n\} \cup \{\textsc{Accept}\}}$ representing the computations of $\mathcal{D}$ on the input $y\dashv$ as follows. For all $1 \leqslant i \leqslant n$ and for all $1 \leqslant j \leqslant n$, the element j is present in $g_y(i)$ if and only if there exists a computation that starts on the first symbol of $y\dashv$ in the state q_i, and first moves to the left beyond that symbol in the state q_j. Also, for all $1 \leqslant i \leqslant n$, the element $\textsc{Accept}$ is present in $g_y(i)$ if and only if there exists a computation that starts on the first symbol of $y\dashv$ in the state q_i, and accepts the string.

The constructions below are given for arbitrary left and right extended profiles $f \colon \{1, \ldots, n\} \cup \{\textsc{Start}\} \to 2^{\{1,\ldots,n\}}$ and $g \colon \{1, \ldots, n\} \to 2^{\{1,\ldots,n\} \cup \{\textsc{Accept}\}}$, which do not necessarily correspond to some actual prefixes and suffixes.

For a left extended profile f and a right extended profile g, define a directed bipartite graph $G_{f,g}$ with $V(G_{f,g}) = (\{1, \ldots, n\} \times \{0, 1\}) \cup \{\textsc{Start}, \textsc{Accept}\}$ with two parts and two additional vertices $\textsc{Start}$ and $\textsc{Accept}$ as follows:

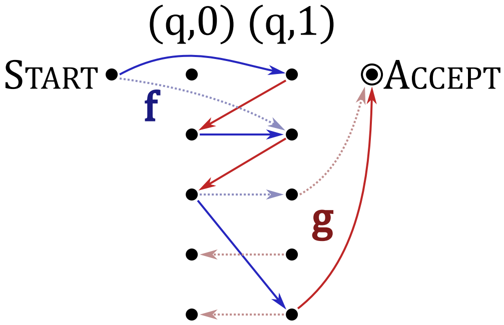


Figure 14: A computation graph, with a path from Start to Accept.

- for all $1 \leqslant i \leqslant n$ and all $j \in f(i)$ there is an arrow from $(i, 0)$ to $(j, 1)$;
- for all $1 \leqslant j \leqslant n$ and all $i \in g(j)$ there is an arrow from $(j, 1)$ to $(i, 0)$;
- for all $1 \leqslant j \leqslant n$ such as $j \in f(\textsc{Start})$ there is an arrow from Start to $(j, 1)$;
- for all $1 \leqslant j \leqslant n$ such as $\textsc{Accept} \in g(j)$ there is an arrow from $(j, 1)$ to Accept.

The resulting graph $G_{f,g}$ shall be called the *computation graph* for a pair of extended profiles (f, g). The form of such a graph is illustrated in Figure 14.

As shown in the following lemma, the matrix M actually depends only on the extended profiles on the strings in X and in Y, rather than on the strings themselves. For the strings in X, it is enough to know only the left extended profiles on those strings, and similarly, only right extended profiles on the strings in Y.

Lemma 32. *Let $\mathcal{D}$ be a 2UFA over an alphabet Γ. Let x and y be strings over Γ. Let f_x be the left extended profile on x, and let g_y be the right extended profile on y. Let G be the computation graph for the pair of extended profiles (f_x, g_y). Then, $\mathcal{D}$ accepts the string xy if and only if there is a path from* Start *to* Accept *in the graph G. Furthermore, if such a path exists, it is unique.*

Proof. The string $\vdash xy \dashv$, on which 2UFA $\mathcal{D}$ operates, can be divided into two parts $\vdash x$ and $y \dashv$. For any computation of $\mathcal{D}$ on this string let $r_0, p_0, r_1, p_1, \ldots$ be the sequence of numbers of states in which $\mathcal{D}$ crosses the boundary between x and y, in the order they appear in the computation.

By the definition of extended profiles, $r_0 \in f_x(\textsc{Start})$, $r_k \in f_x(p_{k-1})$, and $p_k \in g_y(r_k)$; furthermore, the string xy is accepted by $\mathcal{D}$ if and only if for at least one computation the sequence ends with r_k with $\textsc{Accept} \in g_y(r_k)$.

This means that $\textsc{Start}, (r_0, 1), (p_0, 0), (r_1, 1), \ldots$ is a valid path in G. If $\mathcal{D}$ accepts the string xy, then this path for accepting computation ends with a vertex $(r_k, 1)$ such that $\textsc{Accept} \in g_y(r_k)$. Hence, there is an arrow from $(r_k, 1)$ to Accept, which concludes a path from Start to Accept in G.

Conversely, if there is a path $\textsc{Start}, (r_0, 1), (p_0, 0), (r_1, 1), \ldots, (r_k, 1), \textsc{Accept}$ in G, then one can construct a computation with the sequence $r_0, p_0, r_1, p_1, \ldots, r_k$ (by the definition of extended profiles it is possible to get from the state r_i on the symbol directly to the right from the border to the state p_i on the symbol directly next to the border on the left, and from p_i to r_{i+1} as well; also, it is possible to cross the border for the first time in the state r_0). This sequence ends with r_k such that $\textsc{Accept} \in g_y(r_k)$. Therefore, $\mathcal{D}$ accepts xy.

If there are two paths leading from Start to Accept in G, then it is possible to construct two accepting computations on the string $\vdash xy \dashv$ that have different sequences of states in which they crossed the boundary between x and y. However, 2UFA cannot have two different accepting computations on the same string, which leads to a contradiction. □

Corollary 3. *If there are strings x_i and x_j with equal left extended profiles, then either of them can be removed from X without affecting the rank of the matrix M. Similarly, if there are strings y_i and y_j with equal right extended profiles, then either of them can be removed from Y without affecting the rank of M.*

Proof. By Lemma 32, if there are strings x_i and x_j with equal left extended profiles, then rows i and j of the matrix M are equal, since x_i can be replaced with x_j in the string $x_i y$ without affecting the acceptance status. The removal of duplicate rows preserves the rank.

The case or right extended profiles is handled in the same way, by removing duplicate columns. □

Exclude all unnecessary strings, as per Corollary 3, in any order. Now for any left extended profile there is at most one string in X with this profile, and for any right extended profile there is at most one string in Y with this profile, and the rank of matrix M has not changed.

Define a matrix $\mathfrak{M}^{(n)}$, with rows labeled with left extended profiles and columns labeled with right extended profiles, as follows: for a left extended profile f and a right extended profile g, define $\mathfrak{M}^{(n)}_{f,g}$ as the number of paths from $\textsc{Start}$ to $\textsc{Accept}$ in the computation graph $G_{f,g}$ that never visit the same vertex twice.

Lemma 33. $\operatorname{rank} M \leqslant \operatorname{rank} \mathfrak{M}^{(n)}$.

Proof. Matrix M is a submatrix of matrix $\mathfrak{M}^{(n)}$. Indeed, by Lemma 32, matrices M and $\mathfrak{M}^{(n)}$ are constructed based on the same rule (strings in the definition of M are replaced with their extended profiles in the definition of $\mathfrak{M}^{(n)}$): for a left extended profile f for string x_i and a right extended profile g for string y_j, if there is no path in the corresponding computation graph $G_{f,g}$, then $M_{i,j} = \mathfrak{M}^{(n)}_{f,g} = 0$; if there is a path, then this path is unique, and it never visits the same vertex twice (or it would be possible to construct another path by omitting the part between visits), so $M_{i,j} = \mathfrak{M}^{(n)}_{f,g} = 1$. The only difference is that matrix $\mathfrak{M}^{(n)}$ contains rows *for all* left extended profiles and columns *for all* right extended profiles, while matrix M contains only a subset of those.

Hence, $\operatorname{rank} M \leqslant \operatorname{rank} \mathfrak{M}^{(n)}$. □

Then, it is sufficient to prove the desired upper bound on the rank of $\mathfrak{M}^{(n)}$. The latter matrix does not depend on the automaton $\mathcal{D}$.

6.2 Inclusion-exclusion formulas

In order to provide a link between matrices $M^{(n)}$ and $\mathfrak{M}^{(n)}$, several new notions shall be introduced.

Definition 21. Let f and f' be two left extended profiles. The profile f' shall be called a *subprofile* of the profile f, if $f'(x) \subseteq f(x)$ for every x. Notation: $f' \preccurlyeq f$.

Definition 22. Let g and g' be two right extended profiles. The profile g' shall be called a *subprofile* of the profile g, if $g'(x) \subseteq g(x)$ for every x. Notation: $g' \preccurlyeq g$.

Definition 23. Let $G_{f,g}$ be a computation graph for a pair of extended profiles (f, g). The set of vertices of $G_{f,g}$, is split into the left half and the right half, as follows.

$$L(G_{f,g}) = \{\textsc{Start}\} \cup (\{1, \ldots, n\} \times \{0\})$$
$$R(G_{f,g}) = \{\textsc{Accept}\} \cup (\{1, \ldots, n\} \times \{1\})$$

Then, the arrows in $G_{f,g}$ shall be called *left-to-right* and *right-to-left*. Furthermore, arrows from $\{1, \ldots, n\} \times \{1\}$ to Accept, by convention, shall be regarded as right-to-left.

Definition 24. Define a new integer matrix $\mathfrak{L}^{(n)}$, with rows labeled with left extended profiles and columns labeled with right extended profiles, as follows: for a left extended profile f and a right extended profile g, define $\mathfrak{L}^{(n)}_{f,g}$ as the number of paths from Start to Accept in the computation graph $G_{f,g}$ that never visit the same vertex twice and use all left-to-right arrows.

For an extended profile f, left or right, denote by $|f|$ the sum of $|f(x)|$ for all x.

Lemma 34. *Let f be a left extended profile. Let g be a right extended profile. Then,*

$$\mathfrak{M}^{(n)}_{f,g} = \sum_{f' \preccurlyeq f} \mathfrak{L}^{(n)}_{f',g}$$

Proof. It is enough to prove that every path is counted the same number of times on both sides. Note that paths counted on the right-hand side must be counted on the left-hand side as well, since for $f' \preccurlyeq f$ the computation graph $G_{f',g}$ is a subgraph of $G_{f,g}$. Therefore, it is sufficient to check only the paths that are present in $G_{f,g}$.

Consider a path from Start to Accept in $G_{f,g}$, that does not visit any vertex twice. It will be counted in exactly one of $\mathfrak{L}^{(n)}_{f',g}$; namely, the one with the left extended profile corresponding to the set of all left-to-right arrows used by the path (indeed, then any other subprofiles would either generate an additional left-to-right arrow, or exclude an important one). □

Lemma 35. $\operatorname{rank} \mathfrak{L}^{(n)} = \operatorname{rank} \mathfrak{M}^{(n)}$.

Proof. By Lemma 34, the rows of $\mathfrak{M}^{(n)}$ are linear combinations of the rows of $\mathfrak{L}^{(n)}$, and hence $\operatorname{rank} \mathfrak{L}^{(n)} \geqslant \operatorname{rank} \mathfrak{M}^{(n)}$. The ranks are actually the same, since there exists a reverse representation

$$\mathfrak{L}^{(n)}_{f,g} = \sum_{f' \preccurlyeq f} (-1)^{|f|-|f'|} \mathfrak{M}^{(n)}_{f',g}$$

□

The same operation can be done with columns as well.

Definition 25. Define a new integer matrix $\mathfrak{K}^{(n)}$, with rows labeled with left extended profiles and columns labeled with right extended profiles, as follows: for a left extended profile f and a right extended profile g, define $\mathfrak{K}^{(n)}_{f,g}$ as the number of paths from Start to Accept in the computation graph $G_{f,g}$ that never visit the same vertex twice and use all left-to-right arrows and all right-to-left arrows.

Lemma 36. *Let f be a left extended profile. Let g be a right extended profile. Then,*

$$\mathfrak{L}^{(n)}_{f,g} = \sum_{g' \preccurlyeq g} \mathfrak{K}^{(n)}_{f,g'}$$

Proof. The proof is the same as for Lemma 34. □

Lemma 37. $\operatorname{rank} \mathfrak{K}^{(n)} = \operatorname{rank} \mathfrak{L}^{(n)}$.

Proof. By Lemma 36, the rows of $\mathfrak{L}^{(n)}$ are linear combinations of the rows of $\mathfrak{K}^{(n)}$, and hence $\operatorname{rank} \mathfrak{K}^{(n)} \geqslant \operatorname{rank} \mathfrak{L}^{(n)}$. The ranks are actually the same, since there exists a reverse representation

$$\mathfrak{K}^{(n)}_{f,g} = \sum_{g' \preccurlyeq g} (-1)^{|g|-|g'|} \mathfrak{L}^{(n)}_{f,g'}$$

□

6.3 Internal structure of matrix $\mathfrak{K}^{(n)}$

Definition 26. A left extended profile f is *normal*, if the following is true:

- $|f(x)| \leqslant 1$ for every x;
- $f(\textsc{Start}) \neq \varnothing$;
- $f(x) \cap f(y) = \varnothing$ for every $x \neq y$.

For every normal left extended profile f, there is a corresponding prefix profile. Let $P \cup \{\textsc{Start}\}$ be the set of arguments x for which $|f(x)| = 1$, and let $R = \bigcup_{x \in P \cup \{\textsc{Start}\}} f(x)$. Define the function $f^\circ \colon P \cup \{\textsc{Start}\} \to R$ as follows: for $x \in P \cup \{\textsc{Start}\}$, the value of $f^\circ(x)$ is equal to the only element of $f(x)$. Thanks to the normality of f, the function f° is a bijection. Then, (P, R, f°) is a prefix profile corresponding to f.

Definition 27. A right extended profile g is *normal*, if the following is true:

- $|g(x)| \leqslant 1$ for every x;
- $\textsc{Accept} \in g(x)$ for some x;
- $g(x) \cap g(y) = \varnothing$ for every $x \neq y$.

For a normal right extended profile g, there also is a corresponding suffix profile. Let T be the set of arguments x for which $|g(x)| = 1$, and let $S \cup \{\textsc{Accept}\} = \bigcup_{x \in R} g(x)$. Define the function $g^\circ \colon T \to S \cup \{\textsc{Accept}\}$ as follows: for $x \in T$, the value of $g^\circ(x)$ is equal to the only element of $g(x)$. Thanks to the normality of g, the function g° is a bijection. Then, (g°, S, T) is the suffix profile corresponding to g.

For a normal left extended profile f and a normal right extended profile g, define a new graph $H_{f,g}$ as follows. Let (P, R, f°) be the prefix profile corresponding to f, and let (g°, S, T) be the suffix profile corresponding to g. Let $x_{P,R,f^\circ}\, y_{g^\circ,S,T} = (x, \mathtt{l})\,(f', \mathtt{l})\,(g', \mathtt{r})\,(y, \mathtt{r})$. Then, $H_{f,g}$ is constructed from $G_{f,g}$ by merging the vertices $\textsc{Start}$ and $(x, 0)$ into one vertex $(x, 0)$ and by merging the vertices $\textsc{Accept}$ and $(y, 1)$ into $(y, 1)$, followed by removing the resulting loop at $(y, 1)$.

Also, define another graph $D_{f,g}$ with the set of vertices $\{1, \ldots, n\} \times \{0, 1\}$, as follows: for every i with $f'(i)$ defined, there is an arrow from $(i, 0)$ to $(f'(i), 1)$; for every i with $g'(i)$ defined, there is an arrow from $(i, 1)$ to $(g'(i), 0)$; and there are no other arrows. This graph represents all possible computations of $\mathcal{D}_n$ on the second and the third symbols of the string $(x, \mathtt{l})\,(f', \mathtt{l})\,(g', \mathtt{r})\,(y, \mathtt{r})$.

Lemma 38. *Let f be a normal left extended profile. Let g be a normal right extended profile. Then, $D_{f,g} = H_{f,g}$.*

Proof. Let (P, R, f°) be the prefix profile corresponding to f, and let (g°, S, T) be the suffix profile corresponding to g. Let $x_{P,R,f^\circ}\, y_{g^\circ,S,T} = (x, \mathtt{l})\,(f', \mathtt{l})\,(g', \mathtt{r})\,(y, \mathtt{r})$.

Let there be an arrow from $(i, 0)$ to $(j, 1)$ in $D_{f,g}$. That means that $f'(i) = j$. By the construction of f', either $i = x$ and $f_0(\textsc{Start}) = j$, or $i \neq x$ and $f^\circ(i) = j$. In the former case, $j \in f(\textsc{Start})$ and there is an arrow from $\textsc{Start}$ to $(j, 1)$ in the graph $G_{f,g}$. Then, there is an arrow from $(i, 0)$ to $(j, 1)$ in the graph $H_{f,g}$, since $i = x$. In the other case, if $i \neq x$ and $f^\circ(i) = j$, then $j \in f(i)$ and there is an arrow from $(i, 0)$ to $(j, 1)$ in the graph $G_{f,g}$. Then, this arrow is present in the graph $H_{f,g}$ as well.

Let there be an arrow from $(i, 1)$ to $(j, 0)$ in $D_{f,g}$. That means that $g'(i) = j$. Then $g^\circ(i) = j$ by the construction of g', and $j \in g(i)$. Hence, there is an arrow from $(i, 1)$ to $(j, 0)$ in the graph $G_{f,g}$. Then, the graph $H_{f,g}$ contains this arrow as well.

Overall, there are $|f| + |g| - 1$ arrows in the graph $H_{f,g}$ (one arrow to Accept is excluded), which is equal to $|P| + |T|$ (since $|P| = |f| - 1$ and $|T| = |g|$ due to the fact that f and g are normal). The latter is the number of arrows in $D_{f,g}$, because f' defines $|P| + 1$ arrows (since f' is defined on $P \cup \{x\}$) and g' defines $|T| - 1$ arrows (since g' is defined on $T \setminus \{y\}$). Hence, there are no additional arrows in $H_{f,g}$. □

Lemma 39. *Let f be a normal left extended profile. Let g be a normal right extended profile. Let (P, R, f°) be the prefix profile corresponding to f, and let (g°, S, T) be the suffix profile corresponding to g. Then, $\mathfrak{K}^{(n)}_{f,g} = K^{(n)}_{(P,R,f^\circ),(g^\circ,S,T)}$.*

Proof. Note that $\mathfrak{K}^{(n)}_{f,g}$ can be equal only to 0 or 1. Indeed, since f and g are normal, out-degree of every vertex of the corresponding computation graph $G_{f,g}$ does not exceed 1 (since $|f(x)| \leqslant 1$ for every x and $|g(x)| \leqslant 1$ for every x). Furthermore, outdegree of Accept vertex is zero (by construction of computation graph). Therefore, if path from Start to Accept exists, then it is uniquely determined by picking the only possible outgoing arrow until arriving in Accept.

By definition, $\mathfrak{K}^{(n)}_{f,g} = 1$ is equivalent to the existence of a path from Start to Accept that uses all arrows in the graph $G_{f,g}$.

The existence of a path from Start to Accept in the graph $G_{f,g}$ that uses all arrows is equivalent to the existence of a path from $(x, 0)$ to $(y, 1)$ in the graph $H_{f,g}$ that uses all arrows (since Start is replaced by $(x, 0)$ and Accept is replaced by $(y, 1)$, and there were no other arrows from $(x, 0)$ before merging).

By Lemma 38, graph $H_{f,g}$ corresponds to string $x_{P,R,f^\circ}\, y_{g^\circ,S,T}$. Hence, $K^{(n)}_{(P,R,f^\circ),(g^\circ,S,T)} = 1$ is equivalent to the existence of a path from $(x, 0)$ to $(y, 1)$ in $H_{f,g}$ that uses all arrows.

The statement of this lemma is derived from these three equivalences by combining them. □

However, profiles **that are not normal** can be ignored:

Lemma 40. *Let f be a left extended profile which is not normal. Then, $\mathfrak{K}^{(n)}_{f,g} = 0$ for every right extended profile g.*

Similarly, let g be a right extended profile which is not normal. Then, $\mathfrak{K}^{(n)}_{f,g} = 0$ for each left extended profile f.

Proof. It is enough to check that if any condition from the definition of normal profiles is not satisfied, then $\mathfrak{K}^{(n)}_{f,g} = 0$ for every choice of other extended profile.

If there exists an argument x such that $|f(x)| > 1$, then there is no path from Start to Accept that uses all arrows and does not visit any vertex twice. Indeed, if such x exists, the out-degree of corresponding vertex ($(x, 0)$ if $x \neq$ Start, and Start otherwise) is equal to $|f(x)| > 1$. For any path counted in $\mathfrak{K}^{(n)}_{f,g}$, all of those arrows are used in a path. Therefore, said path visits the corresponding vertex more than once, which leads to a contradiction. Hence, $\mathfrak{K}^{(n)}_{f,g} = 0$. The same is true if $|g(x)| > 1$ for some x (for a vertex $(x, 1)$).

If $f(\text{Start}) = \varnothing$, then for all extended profiles g there are no arrows from Start in $G_{f,g}$. Therefore, there are no paths from Start to Accept even without additional conditions, and $\mathfrak{K}^{(n)}_{f,g} = 0$. Similarly, if $\text{Accept} \notin g(x)$ for every x, then there are no arrows to Accept, and paths from Start to Accept cannot exist as well.

Suppose then that the third condition does not hold, and there exists an element j such that for two arguments $x \neq y$, $j \in f(x)$ and $j \in f(y)$. Then, in-degree of the vertex corresponding to j (either $(j,1)$ if $j \neq$ Accept, or Accept otherwise) is at least 2, and any path that uses all arrows visits this vertex twice. Therefore, $\mathfrak{K}^{(n)}_{f,g} = 0$. The same is true for right extended profiles as well (for a vertex $(j,0)$). □

Theorem 7. $\operatorname{rank} \mathfrak{K}^{(n)} = \operatorname{rank} K^{(n)}$

Proof. By Lemma 40, rows and columns of $\mathfrak{K}^{(n)}$, labeled with extended profiles that are not normal, contain only zeroes, and their exclusion would not affect the rank of $\mathfrak{K}^{(n)}$.

By Lemma 39, matrix $\mathfrak{K}^{(n)}$ (with said rows and columns excluded) is identical to $K^{(n)}$ up to permutation of rows and columns. Hence, ranks are also equal. □

Proof of Theorem 6. By Lemma 33, $\operatorname{rank} M \leqslant \operatorname{rank} \mathfrak{M}^{(n)}$. By Lemma 35, $\operatorname{rank} \mathfrak{L}^{(n)} = \operatorname{rank} \mathfrak{M}^{(n)}$. By Lemma 37, $\operatorname{rank} \mathfrak{K}^{(n)} = \operatorname{rank} \mathfrak{L}^{(n)}$. By Theorem 7, $\operatorname{rank} \mathfrak{K}^{(n)} = \operatorname{rank} K^{(n)}$. By Lemma 14, $\operatorname{rank} K^{(n)} = \sum_{k=1}^{n} \binom{n}{k-1}\binom{n}{k} \operatorname{rank} P^{(k)}$ Hence, $\operatorname{rank} M \leqslant \sum_{k=1}^{n} \binom{n}{k-1}\binom{n}{k} \operatorname{rank} P^{(k)}$. □

7 Conclusion

The bounds on the state complexity of transforming 2DFA and 2UFA to UFA established in this work put it asymptotically between $\Omega\big(\frac{9^n}{n^{3/2}}\big)$ and $O(2^n \cdot n!)$, which shows that this is actually a new function different from the known tradeoffs [6]. For small values of n, the bounds proved in this work are compared in Table 2 (the known upper bound on the 2UFA-to-UFA tradeoff, obtained from 2NFA-to-DFA transformation, is not present in the table for shortness). All lower bounds, including the precise bounds by Kapoutsis [6], rely on using an alphabet of exponential size, if the size of the alphabet is subexponential, the upper bounds are improved [4].

It would be interesting to determine the 2DFA-to-UFA and 2UFA-to-UFA tradeoffs precisely. However, as shown in Theorem 6, the lower bound methods based on Schmidt's theorem have virtually been exhausted: it remains to establish the exact rank of the matrix $P^{(k)}$, which is conjectured to be the same as the lower bound in Theorem 4. New methods would be needed for any further improvements.

Table 2: The bounds established in this work for small values of n, compared to the known tradeoffs from two-way to one-way finite automata.

n	2DFA → NFA	**2DFA → UFA**	**2DFA → UFA**	2DFA → DFA
	2UFA → NFA	**2UFA → UFA**	**2UFA → UFA**	
	2NFA → NFA	(lower bound)	(upper bound)	
	$\binom{2n}{n+1}$	$\sum_{k=1}^{n}\binom{n}{k-1}\binom{n}{k}\binom{2k-2}{k-1}$	$\sum_{k=1}^{n}\binom{n}{k-1}\binom{n}{k}k!$	$n(n^n-(n-1)^n)$
1	1	1	1	1
2	4	6	6	6
3	15	39	39	57
4	56	276	292	700
5	210	2 055	2 505	10 505
6	792	15 798	24 306	186 186
7	3 003	124 173	263 431	3 805 249
8	11 440	992 232	3 154 824	88 099 230
9	43 758	8 030 943	41 368 977	2 278 824 849
10	167 960	65 672 850	589 410 910	65 132 155 990

References

[1] J.-C. Birget, “State-complexity of finite-state devices, state compressibility and incompressibility”, *Mathematical Systems Theory*, 26:3 (1993), 237–269.

[2] V. Geffert, C. Mereghetti, G. Pighizzini, “Converting two-way nondeterministic unary automata into simpler automata”, *Theoretical Computer Science*, 295:1–3 (2003), 189–203.

[3] V. Geffert, C. Mereghetti, G. Pighizzini, “Complementing two-way finite automata”, *Information and Computation*, 205:8 (2007), 1173–1187.

[4] V. Geffert, A. Okhotin, “One-way simulation of two-way finite automata over small alphabets”, *NCMA 2013* (Umeå, Sweden, 13–14 August 2013).

[5] J. Jirásek Jr., G. Jirásková, J. Šebej, “Operations on unambiguous finite automata”, *International Journal of Foundations of Computer Science*, 29:5 (2018), 861–876.

[6] C. A. Kapoutsis, “Removing bidirectionality from nondeterministic finite automata”, *Mathematical Foundations of Computer Science* (MFCS 2005, Gdansk, Poland, 29 August–2 September 2005), LNCS 3618, 544–555.

[7] C. A. Kapoutsis, *Algorithms and Lower Bounds in Finite Automata Size Complexity*, Ph. D. thesis, Massachusetts Institute of Technology, 2006.

[8] C. A. Kapoutsis, “Two-way automata versus logarithmic space”, *Theory of Computing Systems*, 55:2 (2014), 421–447.

[9] M. Kunc, A. Okhotin, “Describing periodicity in two-way deterministic finite automata using transformation semigroups”, *Developments in Language Theory* (DLT 2011, Milan, Italy, 19–22 July 2011), LNCS 6795, 324–336.

[10] H. Leung, “Descriptional complexity of NFA of different ambiguity”, *International Journal of Foundations of Computer Science*, 16:5 (2005), 975–984.

[11] C. Mereghetti, G. Pighizzini, “Optimal simulations between unary automata”, *SIAM Journal on Computing*, 30:6 (2001), 1976–1992.

[12] F. R. Moore, “On the bounds for state-set size in the proofs of equivalence between deterministic, nondeterministic, and two-way finite automata”, *IEEE Transactions on Computers*, 20 (1971), 1211–1214.

[13] A. Okhotin, “Unambiguous finite automata over a unary alphabet”, *Information and Computation*, 212 (2012), 15–36.

[14] M. Raskin, “A superpolynomial lower bound for the size of non-deterministic complement of an unambiguous automaton”, *45th International Colloquium on Automata, Languages, and Programming* (ICALP 2018, Prague, Czech Republic, July 9–13, 2018), LIPIcs 107.

[15] E. M. Schmidt, *Succinctness of Description of Context-Free, Regular and Unambiguous Languages*, Ph. D. thesis, Cornell University, 1978.

[16] B. E. Sagan, *The Symmetric Group*, Springer-Verlag, New York, 2001.

[17] W. J. Sakoda, M. Sipser, "Nondeterminism and the size of two way finite automata", *10th ACM Symposium on Theory of Computing* (STOC 1978), 275–286.

[18] J. C. Shepherdson, "The reduction of two-way automata to one-way automata", *IBM Journal of Research and Development*, 3 (1959), 198–200.

[19] M. Vardi, "A note on the reduction of two-way automata to one-way automata", *Information Processing Letters*, 30:5 (1989), 261–264.