Regular Languages, Sizes of Syntactic Monoids, Graph Colouring, State Complexity Results, and How These Topics are Related to Each Other

Markus Holzer
Technische Universität München
holzer@in.tum.de

Barbara König
Universität Stuttgart
koenigba@fmi.uni-stuttgart.de

Abstract

We invite the reader to join our quest for the largest subsemigroup of a transformation monoid on n elements generated by two transformations. Some of the presented results were independently obtained by the authors [6, 7, 8] and Krawetz, Lawrence, and Shallit [12, 13]. In particular, we will see how a surprising connection to graph colouring and chromatic polynomials is very helpful to count the elements of the investigated subsemigroup of transformations. At the end of our search, we will present some applications of these results to state complexity problems for one- and two-way finite automata.

1 Introduction

Our search started after a tutorial for a course on formal languages given by the first author. The students learned about the concept of recognizability, which says that a language $L \subseteq \Sigma^*$ is *recognizable* if and only if there exists a finite monoid M, a morphism $\varphi : \Sigma^* \to M$, and a subset $N \subseteq M$ such that $L = \varphi^{-1}(N)$, which in turn is equivalent to the regularity (acceptance by a finite state machine) of L. For the students it was surprising to see this unexpected relation between the theory of formal languages and algebra. The idea is quite abstract, therefore I used an exercise taken from Pin's book [19] on varieties of formal languages:

Let $A = (Q, \Sigma, \delta, q_0, F)$ be a deterministic finite automaton where Q is the finite set of states, Σ is a finite alphabet, $\delta : Q \times \Sigma \to Q$ denotes the transition function, $q_0 \in Q$ is the initial state, and $F \subseteq Q$ is the set of final states. Its behaviour can be described by a monoid, the so-called transformation monoid. As an example we took the automaton depicted in Figure 1. One argues that each word $w \in \Sigma^*$

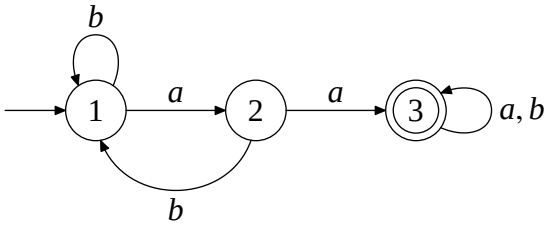

Figure 1: Deterministic finite automaton A.

naturally defines a function from Q into Q, and that the monoid $M(A)$ generated by all these functions, where w varies over Σ^*, is a sub-monoid of $T(Q)$. Here $T(E)$, for a finite set E, denotes the monoid of functions from E into E together with function composition, where we read composition from left to right, i.e., in $\alpha\beta$ first α is applied, then β. Because of this convention, it is natural to write the argument i of a function to the left of a function, i.e., $(i)\alpha\beta = ((i)\alpha)\beta$. In particular, if $E = \{1, \ldots, n\}$, we simply write T_n for the monoid $T(E)$. It holds that $|T(E)| = |E|^{|E|}$.

Back to the transformation monoid: I convinced the students that it is generated by the functions corresponding to the letters of the alphabet, that there is a canonical morphism $\Sigma^* \to M(A)$, and that the set $N \subseteq M(A)$ should be equal to all transformations that map the initial state q_0 to some final state in F.

Thus, I successively calculated the functions that are defined by the words of Σ^*. I started with the letters a and b, followed by all words of length two, where it turns out that b^2 describes the same function as b; one simply remembers the relation $b^2 = b$. This procedure continues with longer words and stops whenever the inferred relations ensure that no new functions can appear. For the example automaton A it turns out that every word of length three induces a function corresponding to a shorter word. Thus, the calculation stops, and we can determine the elements of the transformation monoid. The complete set of functions, together with the computed relations, is shown in Table 1 and is taken from Pin's book as well. Thus, we ended up with the six element transformation monoid

$$M(A) = \{1, a, b, a^2, ab, ba\},$$

where 1 is the neutral element, which is induced by the empty word $\lambda \in \Sigma^*$, and composition is defined via the relations given above. Note, that a^2 acts as a zero

Function table:

	1	2	3
a	2	3	3
b	1	1	3
a^2	3	3	3
ab	1	3	3
ba	2	2	3

Relations:

$$\begin{aligned} b^2 &= b \\ a^3 &= a \\ a^2b &= a \\ aba &= a \\ ba^2 &= a \\ bab &= b \end{aligned}$$

Table 1: Function table and relations induced by the finite automaton A.

element on $M(A)$. It was a tedious exercise since the calculation of the functions took quite some time.

The tutorial finished with an exercise dealing with another important monoid, the syntactic monoid, which is defined, for a given language $L \subseteq \Sigma^*$, by the *syntactic congruence* $\sim_L$ over Σ^* where $v_1 \sim_L v_2$ if and only if $uv_1w \in L \iff uv_2w \in L$ for every $u, w \in \Sigma^*$. Then the *syntactic monoid* is the quotient monoid $M(L) = \Sigma^*/\sim_L$, where the concatenation of equivalence classes $[u]_{\sim_L} \cdot [v]_{\sim_L} = [uv]_{\sim_L}$ serves as the monoid operation. I repeated one of the theorems treated in the course, which says that the syntactic monoid of a regular language L is the smallest monoid recognizing the language under consideration (with respect to the division relation) and that it is isomorphic to the transformation monoid of the minimal finite automaton accepting L. Thus, since the automaton shown in Figure 1 is minimal, we had actually computed the syntactic monoid (up to isomorphism). A good overview on the algebraic theory of formal languages was written by Pin [20].

On my way back to my office I reflected on the tutorial, and suddenly I realized, that my knowledge concerning the relation between the size of a finite automaton and the size of the syntactic monoid for the accepted language was quite limited. Sure, there is the trivial upper bound given by the size of T_n, and I knew some basic facts about generators of T_n and S_n, where S_n denotes the symmetric group of all permutations on n elements. But what else was known, about this question? So that day I discussed this issue with my colleague Barbara, and suddenly we found ourselves on the journey through "algebra land" discovering the secrets behind the largest monoid generated by a finite number of generators. But first we tried to figure out, whether this straightforward question had already been answered by someone else, which, surprisingly, was not the case. After having finished most of our research on that topic, we discovered, that Krawetz, Lawrence, and Shallit [12, 13] had independently conducted research on that question.

So, for the moment we stop thinking about syntactic monoids and concentrate

on transformation semigroups. The question to answer is the following: If you are able to choose a fixed finite number of generators and you should generate as many transformations as possible, which generators would you choose?

> How large is the maximal subsemigroup of T_n, which is generated by a fixed number of transformations?

The next section recalls some basic known facts about generators for S_n and T_n. These results show that the question for S_n instead of T_n is already answered, and that for T_n only the case of two generators lacks an answer. In Section 3 we describe what can be done with two generators. We consider two subcases: Either one of the generators is a single cycle permutation or a permutation with two (or more) cycles. In the penultimate section we present some applications to state complexity problems for one- and two-way finite automata. Finally, we summarize the work on this topic and state some open problems for future research.

2 The Easy Cases. Some Results for the Literature

It is well known that the symmetric group S_n of size $n!$ can be generated by any cyclic permutation on n elements together with an arbitrary transposition, i.e., for instance

$$\alpha = \begin{pmatrix} 1 & 2 & \dots & n-1 & n \\ 2 & 3 & \dots & n & 1 \end{pmatrix} \quad \text{and} \quad \beta = \begin{pmatrix} 1 & 2 & 3 & 4 & \dots & n \\ 2 & 1 & 3 & 4 & \dots & n \end{pmatrix}$$

generate all of S_n—we write $S_n = \langle \alpha, \beta \rangle$, where the semigroup $\langle \alpha, \beta \rangle$ consists of all elements that can be expressed as finite products of elements using α and β. This answers our question, when restricting to the maximal subsemigroup of S_n, in case of two or more generators. We found the following nice and useful result, stating how to find a complete basis for the symmetric group S_n. The theorem given below was shown by Piccard [18].

Theorem 1 (Piccard). *Given a non-identical element α in S_n, then there exists an element β of S_n such that both generate the symmetric group S_n, provided that it is not the case that $n = 4$ and α is one of the three permutations* (1 2)(3 4), (1 3)(2 4), *and* (1 4)(2 3).

But what about a single generator? Here it is quite clear, that one should split n into (preferably coprime) addends in order to find a maximal subsemigroup. This immediately leads us to Landau's function [14], which is given by

$$g(n) = \max\{ \operatorname{lcm}\{i_1, \dots, i_k\} \mid i_1 + \cdots + i_k = n \},$$

and in turn is equivalent to $\max\{\, \mathrm{ord}(\alpha) \mid \alpha \in S_n \,\}$, where $\mathrm{ord}(\alpha)$ denotes the order of α. The function $g(n)$ is well studied and one of the earliest significant results about it is that

$$\lim_{n\to\infty} \frac{\log g(n)}{\sqrt{n \log n}} = 1.$$

A more elaborate bound was given by Szalay [26], who determines the size of the largest submonoid of S_n generated by a single element. This bound also holds in case of the largest submonoid or subsemigroup of T_n generated by a single transformation.

In order to generate all of T_n it suffices to use the following transformations

$$\alpha = \begin{pmatrix} 1 & 2 & \dots & n-1 & n \\ 2 & 3 & \dots & n & 1 \end{pmatrix}, \quad \beta = \begin{pmatrix} 1 & 2 & 3 & 4 & \dots & n \\ 2 & 1 & 3 & 4 & \dots & n \end{pmatrix},$$

$$\text{and} \quad \gamma = \begin{pmatrix} 1 & 2 & \dots & n-1 & n \\ 1 & 2 & \dots & n-1 & 1 \end{pmatrix}.$$

As in the case of the symmetric group, the generators of the transformation monoid T_n are nicely characterizable. For $n \geq 3$ the following completeness theorem for functions of one argument given by Salomaa [23], shows that at least three transformations are needed for T_n. The completeness result reads as follows. Note that the kernel of a transformation α is the equivalence relation $\equiv$, which is induced by $i \equiv j$ if and only if $(i)\alpha = (j)\alpha$.

Theorem 2 (Salomaa). *Assume $n \geq 3$. Then three elements of T_n generate all transformations of T_n if and only if two of them generate the symmetric group S_n and the third has kernel size $n-1$. Moreover, no less than three elements generate all transformations from T_n.*

It is worth mentioning that the above given theorem was re-discovered several times during the years; for instance see Dénes [3]. This gives us the result, that the largest subsemigroup generated by three or more elements has full size n^n. Thus, it remains to consider the case of two generators in more detail.

Finally, let us remark that there is a gap of at least $\binom{n}{2}$ between the size of T_n and the largest proper subsemigroup of T_n. It is quite straightforward to prove that for $n \geq 1$, if $M \subseteq T_n$ is a subsemigroup such that $|M| > n^n - \binom{n}{2}$, then $M = T_n$. This bound was recently shown by Krawetz [12]. It is independent on the number of generators, and therefore it is almost certainly not tight. In the application section we will come back to this bound.

3 What Can You do With Two Generators? The Search for a Maximal Subsemigroup

3.1 The Single Cycle Case

In order to get a better grasp of the problem, we first study a special case where one of the generators is a permutation α consisting of a single cycle only. Such a permutation is often written as follows: $\alpha = (1\ 2\ \ldots\ n)$. As it turns out, one can not generate very many elements with two non-bijective transformations, so one of the two generators should be a permutation. Now, the second generator β should certainly be a non-bijective transformation, since two permutations can generate at most $n!$ elements, which is far below the maximum. So there are at least two indices i, j with $i \neq j$ and $(i)\beta = (j)\beta$ (we say that β *merges* i and j). It seems intuitive that β should not merge too many indices, so we assume that there is only one pair of such indices and that furthermore $i = 1$ and $j = 2$.

In order to get some intuition about the semigroup $\langle \alpha, \beta \rangle$, we state two conditions such that one of them is satisfied by every transformation γ in this semigroup:

(1) The transformation γ is either a multiple of α *or*

(2) there is an index i such that $(i)\gamma = (i+1)\gamma$. Note that we consider summation modulo n in the set $\{1, \ldots, n\}$, i.e., in this case $n + 1 = 1$.

The second condition can be explained as follows: Every transformation γ that is not a multiple of α is of the form $\gamma = \alpha^k \beta \gamma'$, which means that Condition (2) holds for $i = n + 1 - k$. Using Theorem 1 we can show that β can be chosen in such a way that *all* permutations satisfying either Condition (1) or Condition (2) can be generated. This subsemigroup of T_n is denoted by V_n^1 and can be generated by two transformations only.

However, we still have to count all those transformations. Here we find that a surprising connection to graph colouring and chromatic polynomials is very helpful. Imagine a cyclic graph, i.e., a ring, consisting of n nodes, which should be coloured using up to n colours. Then the number of all invalid colourings of this graph, i.e., all colourings where two neighbouring nodes are assigned equal colours is equal to the number of all transformations satisfying Condition (2). This is easy to see, just regard a transformation γ as a colouring function, assigning colours to nodes.

The chromatic polynomial $\chi(G, \lambda)$ of a graph G is a polynomial in λ giving the number of possibilities to colour G with λ colours (not all colours have to be used). If G is a cyclic graph with n nodes, its chromatic polynomial is known to be $\chi(G, \lambda) = (\lambda - 1)^n + (-1)^n(\lambda - 1)$—see [21]. Using this fact we can easily

compute the number of invalid colourings with n colours. Furthermore there is an interesting asymptotic result, involving Euler's number e.

Theorem 3. *With two generators, where $\alpha = (1\ 2\ \ldots\ n)$ is a single cycle permutation and the other is a transformation β with $(i)\beta = (i+1)\beta$, for some index $1 \leq i \leq n$, one can generate at most*

$$|V_n^1| = n + n^n - (n-1)^n - (-1)^n(n-1)$$

transformations. This bound is tight. Furthermore

$$\lim_{n\to\infty} \frac{|V_n^1|}{n^n} = \frac{1}{e}.$$

Is it possible to do better if we fix α but choose a different β? In fact, we can do better for some n. For instance, for $n = 6$, the formula above yields $|V_n^1| = 31032$, but we can in fact generate more elements. Using the Groups, Algorithms and Programming (GAP) system[1], a very useful tool for computational discrete algebra, we can for example compute the number of transformations generated by our usual α and

$$\beta = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 4 & 1 & 5 & 6 & 2 \end{pmatrix}.$$

Now we obtain 32262 transformations! So, maybe the secret is mapping two indices with difference 2 (in this case 1 and 3) to the same number? But this cannot be turned into a general result. For instance if $n = 4$ we can generate up to 176 transformations by merging indices 1 and 2, but only 116 transformations by merging 1 and 3.

Again, the connection to graph colouring is very helpful. Let d be a divisor of n (in the example above we chose $d = 2$ and $n = 6$). Now if we choose a transformation β with $(1)\beta = (1+d)\beta$ (again summation is taken modulo n) we can generate all transformations γ satisfying Condition (1) above and Condition (3) below. This subsemigroup is called V_n^d.

(3) There is an index i such that $(i)\gamma = (i+d)\gamma$.

What kind of graph do we have to choose such that the number of its invalid colourings corresponds to the number of all transformations satisfying Condition (3)? The answer is: d disjoint cycles, each of length $\frac{n}{d}$. For instance in the case $n = 6$, $d = 2$, the transformation β above can be represented as the invalid colouring a system of two cycles, both of length three—see Figure 2. Note that the colours are written inside the circles, whereas the number of a node is written beside it.

[1] `http://www.math.rwth-aachen.de/~GAP/`

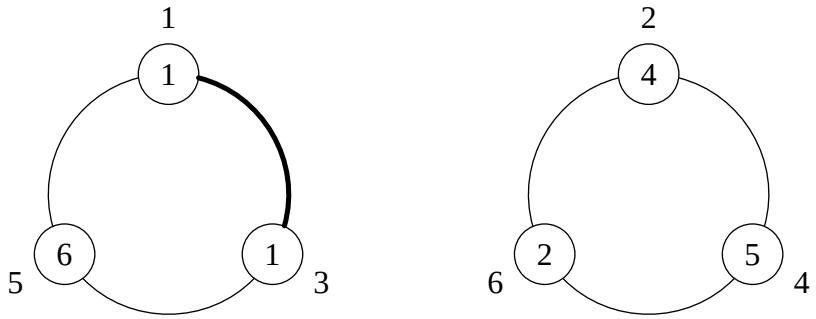

Figure 2: A invalid colouring of two disjoint cycles.

If a graph G is the disjoint union of two graphs G_1 and G_2, it can be easily seen that $\chi(G, \lambda) = \chi(G_1, \lambda) \cdot \chi(G_2, \lambda)$. So the chromatic polynomial of a system of d cycles, each of length $\frac{n}{d}$ corresponds to

$$\chi(G, \lambda) = \left((\lambda - 1)^{\frac{n}{d}} + (-1)^{\frac{n}{d}}(\lambda - 1)\right)^d.$$

This can be used to give a closed formula for the maximal size of a subsemigroup generated by a single cycle and a second non-bijective transformation. Note that above we have only considered divisors of n for a good reason: All other cases can be reduced to that case.

Theorem 4. *Let $\alpha = (1\ 2\ \ldots\ n)$ and let β be a transformation such that $(i)\beta = (i + d')\beta$. Let $d = \gcd\{d', n\}$.*

(1) Then α and β can generate at most

$$|V_n^d| = n + n^n - \left((n - 1)^{\frac{n}{d}} + (-1)^{\frac{n}{d}}(n - 1)\right)^d$$

transformations. This bound is tight.

(2) If n is fixed, we can maximize $|V_n^d|$ by choosing for d the largest divisor of n for which $\frac{n}{d}$ is odd. If $\frac{n}{d'}$ is even for all divisors d' of n we choose $d = 1$. In this case we set $V_n = V_n^d$.

The second part of the theorem explains why $d = 2$ is a good choice if $n = 6$ but not if $n = 4$. From an asymptotic point of view nothing has changed. The fraction $\frac{|V_n|}{n^n}$ still converges to $\frac{1}{e}$ when n goes to infinity.

3.2 The Two Cycle Case

So maybe now we have found the most reproductive pair of generators? No, not yet. For instance if $n = 7$ we can generate at most $|V_7^1| = 543620$ transformations with the method described above. But setting

$$\alpha = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 2 & 1 & 4 & 5 & 6 & 7 & 3 \end{pmatrix} \quad \text{and} \quad \beta = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 3 & 4 & 5 & 6 & 2 & 1 \end{pmatrix}$$

and letting GAP compute (this takes quite a long time) gives us even 610871 transformations!

What is the secret of this new pair of generators? The first transformation α is a permutation that can be written as $\alpha = (1\ 2)(3\ 4\ 5\ 6\ 7)$. The lengths of the two cycles are coprime, which means that they can be turned "independently," i.e., for every pair of numbers q_1 and q_2 one can find a number q such that $\alpha^q = (1\ 2)^{q_1}(3\ 4\ 5\ 6\ 7)^{q_2}$. So, if we let α do some preprocessing and then merge indices 1 and 7 (as the transformation β above does), we can in fact merge every index of the first cycle of α (choose 1 or 2) with every index of the second cycle (choose 3, 4, 5, 6 or 7). This gives us a total of $2 \cdot 5 = 10$ pairs of indices to merge, which looks better than the 7 pairs in the case of V_7^1.

However, it is not possible to generate *every* transformation merging an index of the first and an index of the second cycle. Consider for instance

$$\gamma = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 3 & 4 & 5 & 6 & 7 & 1 \end{pmatrix},$$

a transformation which contains all elements of the second cycle of α (which are 3, 4, 5, 6, 7) in its image. This transformation cannot be of the form $\gamma'\beta$, since the index 7 is missing in the image of β. Similarly, it cannot be of the form $\gamma'\beta\alpha$, since 3 is missing in the image of $\beta\alpha$. Continuing, one can argue that at least one index of the second cycle has to be missing from the image of $\gamma'\beta\alpha^q$.

So, all elements generated by a permutation

$$\alpha = (1\ 2\ \ldots\ k)(k+1\ k+2\ \ldots\ k+\ell),$$

where $\gcd\{k, \ell\} = 1$ and $n = k + \ell$, and a non-bijective transformation β merging elements of different cycles such that at least one element of the second cycle is missing from the image of β, satisfy one of the two conditions below. Let $img(\alpha) = \{(i)\alpha \mid 1 \leq i \leq n\}$, if $\alpha \in T_n$.

(4) The transformation γ is either a multiple of α *or*

(5) there are indices $i \in \{1, 2, \ldots, k\}$, $j \in \{k+1, k+2, \ldots, k+\ell\}$ such that $(i)\gamma = (j)\gamma$ and furthermore there is an index $h \in \{k+1, k+2, \ldots, k+\ell\}$ such that $h \notin img(\gamma)$.

Without loss of generality we assume that $k < \ell$. Note that it is always better to choose h from the larger cycle since in this case there are more possibilities to choose from and we obtain more transformations. Again, by using Theorem 1 we can show that whenever k and ℓ are coprime, transformation β can be chosen in such a way that we can actually generate all transformations satisfying either Condition (4) or Condition (5). Thus, the described semigroup can be defined by two generators only. We call this subsemigroup $U_{k,\ell}$ in the following.

Since the correspondence to graph colouring has led to good results for earlier cases, we can try again to find a connection. In this case it turns out that the number of transformations satisfying Condition (5) is equal to the number of invalid colourings of the complete bipartite graph $K_{k,\ell}$, where the first set contains k nodes and the second set contains ℓ nodes. We may use up to n colours, but at least one colour from the set $\{k+1, k+2, \ldots, k+\ell\}$ has to stay unused. Figure 3 shows that transformation $\beta \in T_7$ given above is indeed an invalid colouring of the complete bipartite graph $K_{2,5}$.

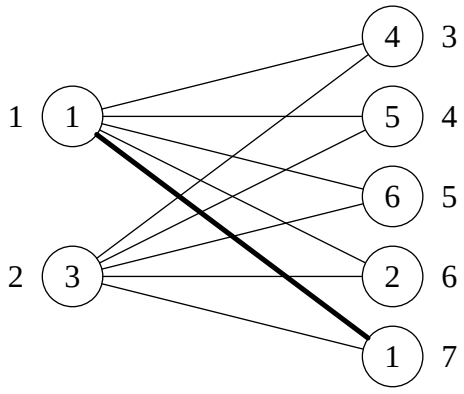

Figure 3: An invalid colouring of a complete bipartite graph.

This second requirement as well as the fact that the chromatic polynomials of complete bipartite graphs are nowhere near as nice as the ones of cycles complicates the computation of the number of elements of $U_{k,\ell}$. Before we proceed with the computation, we first state the following central result.

Theorem 5. *If n is a prime greater than or equal to 7, a subsemigroup of the form $U_{k,\ell}$ with $n = k + \ell$ is maximal in size among all semigroups that can be generated by two generators.*

The theorem says, among other things, that using a permutation with more than two cycles does not help in generating more elements—observe, that the

cases where (i) two arbitrary permutations or (ii) two non-bijective transformations are used as generators cannot do better than the single cycle subsemigroup V_n from the previous subsection. So far we were only able to prove Theorem 5 for prime n. This case is simpler, since k and ℓ are then automatically coprime. For n smaller than 7, one of the semigroups of the form V_n^d is the winner.

We still have to describe how to compute the size of $U_{k,\ell}$ and give a result concerning asymptotics. In order to do so we regard the following formula for a chromatic polynomial of a graph G.

$$\chi(G,\lambda) = \sum_{i=1}^{\lambda} \binom{n}{i} p(G,i) i!,$$

where $p(G,i)$ is the number of possibilities to partition the set of nodes of G into i non-empty independent sets. After having partitioned the nodes, one chooses i colours—$\binom{n}{i}$ possibilities—and assigns them to the i node sets—$i!$ possibilities.

For a complete bipartite graph, $p(K_{k,\ell}, i)$ can be determined using Stirling numbers of the second kind. A Stirling number of the second kind is written $\left\{ {n \atop i} \right\}$ and denotes the number of possibilities to partition an n-element set into i non-empty subsets. It holds that

$$p(K_{k,\ell}, i) = \sum_{r=1}^{i} \left\{ {k \atop r} \right\} \left\{ {\ell \atop i-r} \right\}$$

(for every r, first partition the nodes of the first set into r subsets, then partition the nodes of the second set into $i-r$ subsets). So, in analogy to the formula above, when we attempt to produce an invalid coloring of $K_{k,\ell}$ using exactly i colours, such that at least one colour from the set $\{k+1, k+2, \ldots, k+\ell\}$ is missing, we first choose i colours—$\binom{n}{i}$ possibilities, but have to take into account that we must not use all colours between $k+1$ and $k+\ell$—subtract $\binom{k}{i-\ell}$. Then we choose a partitioning of the nodes which contains at most one non-independent set—there are $\left\{ {n \atop i} \right\} - p(K_{k,\ell}, i)$ possibilities. Everything combined we obtain the following theorem.

Theorem 6. *Let $k > 1$ and let* $\gcd\{k,\ell\} = 1$.

(1) It holds that

$$|U_{k,\ell}| = k\ell + \sum_{i=1}^{n} \left(\binom{n}{i} - \binom{k}{i-\ell} \right) \left(\left\{ {n \atop i} \right\} - \sum_{r=1}^{i} \left\{ {k \atop r} \right\} \left\{ {\ell \atop i-r} \right\} \right) i!.$$

(2) If we choose, for every n, indices $k(n)$ and $\ell(n)$ close to $\frac{n}{2}$, it holds that

$$\lim_{n\to\infty} \frac{|U_{k(n),\ell(n)}|}{n^n} = 1.$$

The second part of the theorem can be shown by doing an under-estimation of the size of $U_{k,\ell}$ and using Stirling's approximation of the factorial [5, 22], which says that

$$n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n \left(1 + \Theta\left(\frac{1}{n}\right)\right),$$

in order to approximate binomial coefficients. This shows that, in an asymptotic sense, one can generate "almost all" transformations using only two generators.

4 Applications

It's now time to give some applications, but let us first summarize the results so far. The original motivation of our journey was to gain more knowledge on the size of syntactic monoids for regular languages. Combining the results stated in the previous two subsections we find the following theorem.

Theorem 7. *Let A be a n-state deterministic finite automaton with input alphabet Σ.*

(1) If Σ is a singleton set, then a monoid of size n is sufficient to recognize the language $L(A)$. This bound is tight.

(2) If $|\Sigma| = 2$ and $n \geq 3$, then a monoid of size $n^n - n! + g(n)$ is sufficient to recognize the language $L(A)$ and a monoid of size at least $n^n(1 - \frac{2}{\sqrt{n}})$ is necessary in the worst case to recognize the language $L(A)$.

(3) In all other cases a monoid of size n^n is sufficient to recognize the language $L(A)$. This bound is tight.

Observe, that in the first case above, the linear bound on the monoid size stems from the fact that a transition graph of a deterministic finite automaton with unary input alphabet consists of a path, which starts at the initial state, followed by a cycle of one or more states. Therefore, the upper bound of Landau's function is not reachable in this case. The upper bound of the second statement is trivial, while the lower bound $n^n(1 - \frac{2}{\sqrt{n}})$ can be obtained from Theorem 6. A slightly better lower bound of $n^n(1 - \frac{4}{n})$ for odd $n \geq 70$ was presented by Krawetz, Lawrence, and Shallit [13], by using a more elaborate counting argument for the number of proper vertex colourings of a bipartite graph, which is due to Lazebnik [15].

The following two applications concern state complexity results for one- and two-way finite automata and were obtained by Krawetz, Lawrence, and Shallit [12, 13]. Since regular languages have many representations in the world

of finite automata, it is natural to investigate the succinctness of their representation by different types of automata. For conversion results we refer to, e.g., [1, 16, 17, 24, 25]. We come back to this issue in the remainder of this section.

Related to these questions are the costs (in terms of states) of operations on regular languages with regard to their representing devices. For example, complementing a language accepted by a given nondeterministic finite automaton can result in an exponential blowup of states. In this case, conversion to an equivalent deterministic automaton and subsequent complementation gives an upper bound. In recent years, results for many operations have been obtained; we refer to, e.g, [2, 4, 9, 10, 27, 28]. Krawetz, Lawrence, and Shallit have studied the deterministic state complexity of the root operation, which is defined by

$$root(L) = \{\, w \in \Sigma^* \mid \text{there is an } n \geq 1 \text{ such that } w^n \in L \,\}$$

for any language $L \subseteq \Sigma^*$. Observe, that this operation is not the same as the operation studied by Horváth, Leupold, and Lischke [11], since in their case the root of a language contains only primitive words (a word is *primitive* if it is not the power of any other word). The idea is to use the close relationship between the root of a regular language and the transformation monoid, which can be stated as follows: If L is accepted by a minimal deterministic finite automaton A, then $root(L)$ is accepted by a deterministic finite automaton with at most $|M(A)|$ states—this can be easily seen by using a similar technique by Zhang [29], who characterized regularity preserving operations. As the following theorem shows, this bound is not tight in general.

Theorem 8 (Krawetz, Lawrence, and Shallit). *Let $L \subseteq \Sigma^*$ be a regular language accepted by a minimal deterministic finite automaton with n states.*

(1) If Σ is a singleton set, then n states are sufficient to accept the language $root(L)$.

(2) If $|\Sigma| = 2$ and $n \geq 7$ a prime number, then there exist integers k and ℓ with $k + \ell = n$, such that $|U_{k,\ell}| - \binom{n}{2}$ states are sufficient to accept the language $root(L)$.

(3) If $|\Sigma| \geq 3$, then $n^n - \binom{n}{2}$ states are sufficient to accept the language $root(L)$.

All the above given bounds are tight.

The problem with the automaton, that accepts the root of the language and which is based on the transformation monoid, is that some of the transformation monoid elements turn out to be equivalent. To be more precise, it turns out, that only certain transformations of kernel size two are equivalent. For a more detailed discussion we refer to Krawetz, Lawrence, and Shallit [12, 13].

Our favourite application comes next. Krawetz, Lawrence, and Shallit applied their result concerning the state complexity of the root operation to the problem of converting a two-way deterministic finite automaton into a one-way deterministic finite automaton; let's call this problem the 2DFA-DFA problem. Birget [1] improved Sheperdson's [25] upper bound from $n(n+1)^n$ to n^n, and previous results giving a bound $n^{\Theta(n)}$ for 2DFA-DFA have been obtained by Meyer and Fischer [16], and Moore [17]. Since, for a regular language accepted by an n-state deterministic automaton, it is possible to accept $root(L)$ by a two-way deterministic finite automaton with end-markers, having $2n$ states, the above theorem gives yet another example of a language with an $n^{\Theta(n)}$ blowup in the number of states.

Theorem 9 (Krawetz, Lawrence, and Shallit). *For sufficiently large n, there exists an n-state two-way deterministic finite automaton with end-markers such that the equivalent one-way deterministic finite automaton has at least*

$$\left(\frac{n}{2}\right)^{\frac{n}{2}} - 4 \cdot \left(\frac{n}{2}\right)^{\frac{n}{2}-1} - \frac{1}{8}n^2$$

states.

For the proof of this theorem the lower bound $n^n(1 - \frac{4}{n})$ mentioned in the previous section comes into play. Nevertheless, a closer look reveals that this lower bound provides a significant improvement of

$$\Theta\left(n^{\frac{5}{2}}\right) \quad \text{and} \quad \Theta\left(\left(\frac{25}{32}\right)^{\frac{1}{10}n} n^{\frac{3}{10}n}\right)$$

over the results by Moore respectively Meyer and Fisher. Finally, note that one cannot use $root(L)$ to prove that the upper bound of n^n on the 2DFA-DFA-problem is tight.

5 Conclusions

We hope that you have enjoyed our journey through all the topics mentioned in the title of this paper. Although our quest for the maximal subsemigroup of T_n generated by two transformations was quite successful, it still lacks a complete answer. First of all, what about the case when $n \geq 7$ is not a prime number? We conjecture that Theorem 5 holds in this case as well, but we have no proof yet. Also, the question of how to best choose k and ℓ remains unanswered. In order to maximize the size of $U_{k,\ell}$ one has to minimize the number of valid colourings, which is minimal if k and ℓ are close to $\frac{n}{2}$. This clashes with the observation that the cycle α from which an element in the image of β is missing should be as large

as possible. Nevertheless, to maximize the size of $U_{k,\ell}$ we conjecture that for large enough n both k and ℓ should be as close to $\frac{n}{2}$ as permitted by the condition that k and ℓ have to be coprime. Again a proof of this statement is still missing. The very nature of the question is relevant to more than just formal language theory, in fact it applies to semigroup in general.

References

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

[2] C. Câmpeanu, K. Culik II, K. Salomaa, and S. Yu. State complexity of basic operations on finite languages. In O. Boldt and H. Jürgensen, editors, *Proceedings of the 4th International Workshop on Implementing Automata*, number 2214 in LNCS, pages 60–70, Potsdam, Germany, July 1999. Springer.

[3] J. Dénes. On transformations, transformation-semigroups and graphs. In *Theory of Graphs: Proceedings of the Colloqium on Graph Theory*, pages 65–75, Tihany, Hungary, 1968. Academic Press.

[4] K. Ellul. Descriptional complexity measures of regular languages. Master thesis, Computer Science, University of Waterloo, Ontario, Canada, 2002.

[5] W. Feller. Stirling's formula. In *An Introduction to Probability Theory and Its Applications*, volume 1, chapter 2.9, pages 50–53. Wiley, 3rd edition, 1968.

[6] M. Holzer and B. König. On deterministic finite automata and syntactic monoid size. *Theoretical Computer Science*. Accepted for publication.

[7] M. Holzer and B. König. On deterministic finite automata and syntactic monoid size. In M. Ito and M. Toyama, editors, *Proceedings of the 6th International Conference on Developments in Language Theory*, number 2450 in LNCS, pages 258–269, Kyoto, Japan, September 2002. Springer.

[8] M. Holzer and B. König. Deterministic finite automata and syntactic monoid size, continued. In Z. Ésik and Z. Fülöp, editors, *Proceedings of the 7th International Conference on Developments in Language Theory*, number 2710 in LNCS, pages 349–360, Szeged, Hungary, July 2003. Springer.

[9] M. Holzer and M. Kutrib. Unary language operations and their nondeterministic state complexity. In M. Ito and M. Toyama, editors, *Proceedings of the 6th International Conference on Developments in Language Theory*, number 2450 in LNCS, pages 162–172, Kyoto, Japan, September 2002. Springer.

[10] M. Holzer and M. Kutrib. Nondeterministic descriptional complexity of regular languages. *International Journal of Foundations of Computer Science*, 14(6):1087–1102, December 2003.

[11] S. Horváth, P. Leupold, and G. Lischke. Roots and powers of regular languages. In M. Ito and M. Toyama, editors, *Proceedings of the* 6*th International Conference on Developments in Language Theory*, number 2450 in LNCS, pages 269–281, Kyoto, Japan, September 2002. Springer.

[12] B. Krawetz. Monoids and the state complexity of the operation *root*(L). Master thesis, Computer Science, University of Waterloo, Ontario, Canada, 2003.

[13] B. Krawetz, J. Lawrence, and J. Shallit. State complexity and the monoid of transformations of a finite set. http://arxiv.org/abs/math.gr/0306416, June 2003.

[14] E. Landau. Über die Maximalordnung der Permutationen gegebenen Grades. *Archiv der Mathematik und Physik*, 3:92–103, 1903.

[15] F. Lazebnik. New upper bounds for the greatest number of proper vertex colorings of a (V, W)-graph. *Journal of Graph Theory*, 14(1):25–29, 1990.

[16] A. R. Meyer and M. J. Fischer. Economy of description by automata, grammars, and formal systems. In *Proceedings of the* 12*th Annual Symposium on Switching and Automata Theory*, pages 188–191. IEEE Computer Society Press, 1971.

[17] F. R. Moore. On the bounds for state-set size in the proofs of equivalence between deterministic, nondeterministic, and two-way finite automata. *IEEE Transaction on Computing*, C-20:1211–1219, 1971.

[18] S. Piccard. *Sur les bases du groupe symétrique et les couples de substitutions qui engendrent un groupe régulier*. Librairie Vuibert, Paris, 1946.

[19] J.-E. Pin. *Varieties of formal languages*. North Oxford, 1986.

[20] J.-E. Pin. Syntactic semigroups. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 1, pages 679–746. Springer, 1997.

[21] R. C. Read. An introduction to chromatic polynomial. *Journal of Combinatorial Theory*, 4:52–71, 1968.

[22] H. Robbins. A remark of Stirling's formula. *American Mathematical Monthly*, 62:26–29, 1955.

[23] A. Salomaa. On the composition of functions of several variables ranging over a finite set. *Annales Universitatis Turkuensis*, 41, 1960. Series AI.

[24] K. Salomaa and S. Yu. NFA to DFA transformation for finite language over arbitrary alphabets. *Journal of Automata, Languages and Combinatorics*, 2:177–186, 1997.

[25] J. C. Shepherdson. The reduction of two-way automata to one-way automata. In E. F. Moore, editor, *Sequential Machines. Selected Papers*, Computer-Science and Information Processing, pages 63–91. Addison-Wesley, 1964.

[26] M. Szalay. On the maximal order in S_n and S_n^*. *Acta Arithmetica*, 37:321–331, 1980.

[27] S. Yu and Q. Zhuang. On the state complexity of intersection of regular languages. *SIGACT News*, 22(3):52–54, 1991.

[28] S. Yu, Q. Zhuang, and K. Salomaa. The state complexity of some basic operations on regular languages. *Theoretical Computer Science*, 125:315–328, 1994.

[29] G.-Q. Zhang. Automata, Boolean matrices, and ultimate periodicity. *Information and Computation*, 152(1):138–154, July 1999.