Chrobak normal form revisited, with applications

Paweł Gawrychowski*

Institute of Computer Science,
University of Wrocław,
ul. Joliot-Curie 15, 50–383 Wroclaw, Poland
gawry@cs.uni.wroc.pl

Abstract. It is well known that any nondeterministic finite automata over a unary alphabet can be represented in a certain *normal form* called the Chrobak normal form [1]. We present a very simple conversion procedure working in $\mathcal{O}(n^3)$ time. Then we extend the algorithm to improve two trade-offs concerning conversions between different representations of unary regular languages. Given an n-state NFA, we are able to find a regular expression of size $\mathcal{O}(\frac{n^2}{\log n})$ describing the same language (which improves the previously known $\mathcal{O}(n^2)$ size bound [8]) and a context-free grammar in Chomsky normal form with $\mathcal{O}(\sqrt{n \log n})$ nonterminals (which improves the previously known $\mathcal{O}(n^{2/3})$ bound [3]).
As a byproduct of our conversion procedure, we get an alternative proof of the Chrobak normal form theorem. We believe that its efficiency and simplicity make the effort of reproving an already known result worthwhile.
Key-words: unary automata, descriptional complexity

1 Introduction

Finite automata are a simple yet particularly ubiquitous and useful model of computation. There exists a vast amount of research devoted to studying trade-offs between different methods of describing a language recognized by such devices, starting with the classic conversions between deterministic and nondeterministic finite automata [10]. In this paper we focus on the cost of converting an automaton to a regular expression. If there are no restrictions on the size of the input alphabet, the conversion might require an exponential blow-up [4], even if the alphabet is binary [7]. On the other hand, if the alphabet consists of just one letter, it turns out that such exponential blow-up is not necessary. Additionally, it turns out that nondeterministic automata over such an alphabet can be converted into the so-called Chrobak normal form, meaning that there exists a nondeterministic automaton M' such that $L(M) = L(M')$ and M' consists of a path of $\mathcal{O}(n^2)$ states followed by a single nondeterministic choice to a set of disjoint cycles, where the cycles contain at most n states altogether [1]

* Supported by MNiSW grant number N N206 492638, 2010–2012

(also see the errata to the original article [2]). The original proof did not address the computational complexity of finding such M' given M. Martinez showed [8] that this conversion requires polynomial time, or more precisely, $\mathcal{O}(n^5)$. This has been improved by Sawa to $\mathcal{O}(n^2(n+m))$ [12]. Both the original proof and the Martinez's improvement contained a minor flaw observed and corrected in [13]. In the next section we give a more efficient version of the construction and then show how to extend the method to construct a regular expression of size $\mathcal{O}(\frac{n^2}{\log n})$ describing the same language. While the improvement might seem minor, it requires combining a few ideas and refutes a conjecture of Martinez who asked for $\Omega(n^2)$ lower bound. Furthermore, we give an evidence that a more substantial improvement would require dramatically different ideas: we show that for some automata converting to Chrobak normal form involves a quadratic blow-up. Then we show that using a similar technique we can construct a context-free grammar in Chomsky normal form with $\mathcal{O}(\sqrt{n \log n})$ nonterminals thus improving the previously known bound $\mathcal{O}(n^{2/3})$ [3].

2 Preliminaries

We are given a nondeterministic finite automaton $M = \langle \Sigma, Q, q_0, \delta, F \rangle$ over a unary alphabet $\Sigma = \{a\}$. Because the automaton is nondeterministic, without loss of generality there is exactly one final state q_f. Similarly, we can assume that there are no edges incoming into q_0. As the alphabet is unary, we can (and will) view the automaton as a directed graph on $n = |Q|$ vertices and $m = |\delta|$ edges, where $m \leq n^2$. The Chrobak normal form of such automaton consists of a path of length $\mathcal{O}(n^2)$ followed by a single nondeterministic choice to a set of disjoint cycles of lengths $c_1, c_2, \ldots, c_\ell$, with $\sum_i c_i \leq n$, see Figure 1.

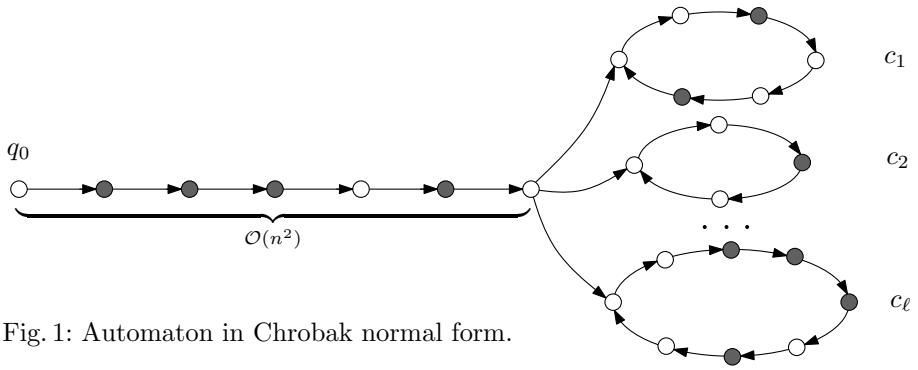

Fig. 1: Automaton in Chrobak normal form.

A *strongly connected component* of a directed graph is a maximal subset of vertices $\{v_1, v_2, \ldots, v_s\}$ such that for any i, j there is a path from v_i to v_j. We call such component nontrivial if there exists at least one edge $v_i \to v_j$ inside, i.e., either $s > 1$ or there is a loop from v_1 to v_1. The *girth* of a directed graph

is the length of its shortest cycle. We will use g to denote the girth of the graph corresponding to the given automaton.

Regular expressions considered in this paper are defined in the standard recursive way: a and ϵ are regular expressions, and if R and S are both regular expressions, so are R^*, RS and $R+S$. The size of such expression is simply the number of characters necessary to write it down.

We will also consider languages described by context-free grammars in Chomsky normal form, meaning that all productions are of the form $A \to a$ or $A \to BC$ where A, B, C are nonterminals. While it is known that context-free grammars over an unary alphabet describe exactly regular languages [11], they might allow a more succinct description of the language in question that nondeterministic automata.

3 The algorithm

While our goal is a $\mathcal{O}(n^3)$ time algorithm, we start with a simpler $\mathcal{O}(n^4)$ version.

We are interested in lengths of paths from q_0 to q_f in the corresponding directed graph G. We are going to compute a succinct description of all such paths. First consider the case when G is acyclic. Then any path consists of at most $n-1$ edges so in time $\mathcal{O}(nm)$ we can easily compute for all vertices v and possible lengths of path $0 \le \ell < n$ whether there exists a path from q_0 to v of length ℓ. This gives us a description of all paths from q_0 to q_f.

Now consider the case when G is not acyclic, i.e., contains vertices belonging to nontrivial strongly connected components. We can compute all lengths of paths from q_0 to q_f avoiding vertices belonging to nontrivial strongly connected components in the same way as in the acyclic case. Thus now we are only concerned with paths that go through at least one vertex v belonging to a nontrivial strongly connected component. We are going to consider all possible choices of v one by one.

Lemma 1. *Given a vertex v belonging to a nontrivial strongly connected component we can represent all accepting paths through v by a path of length $2n^2$ followed by a cycle of length at most n. The representation can be found in $\mathcal{O}(n^3)$ time.*

Proof. First we need (any) simple cycle $v = v_0 \to v_1 \to \cdots \to v_d = v$ containing v. In fact we can find the shortest such cycle in linear time by replacing v with two vertices v' and v'', all edges of the form $u \to v$ with $u \to v''$, and all edges of the form $v \to u$ with $v' \to u$, and computing the shortest path from v' to v''. It corresponds to the shortest cycle containing v. As the cycle is simple, $d \le n$.

Observe that whenever we have a path from q_0 to q_f through v of length ℓ, there is such a path of length $\ell + d$ as well. Thus among all paths with $\ell \equiv r \pmod{d}$ we need to find just the shortest one. Such shortest paths can be computed efficiently in the following way: for each vertex u in the original graph create its $2d$ copies $u(0), u(1), \ldots, u(d-1)$ and $u'(0), u'(1), \ldots, u'(d-1)$. Then add appropriate edges so that there is a path from $q_0(0)$ to $u(r)$ of length ℓ if

and only if $\ell \equiv r \pmod{d}$ and there is a path from q_0 to u of the same length in G. Similarly, there is a path from $q_0(0)$ to $u'(r)$ of length ℓ when $\ell \equiv r \pmod{d}$ and there is a path from q_0 to u going through v and of the same length in G. It is easy to check that the following construction ensures the above conditions: for each edge $x \to y$ and for all possible values of $r = 0, 1, \ldots, d-1$ create edges:

1. $x(r) \to y((r+1) \bmod d)$
2. $x'(r) \to y'((r+1) \bmod d)$
3. if $y = v$, $x(r) \to y'((r+1) \bmod d)$

Then use breadth first search to find shortest paths from $q_0(0)$ in the resulting graph. This requires time $\mathcal{O}(d(n+m)) = \mathcal{O}(nm)$ and gives us a succinct description of all paths from q_0 to q_f going through v. Indeed, there is such path of length ℓ if and only if the distance to $q'_f(\ell \bmod d)$ is finite and does not exceed ℓ. Observe that the new graph contains $2dn$ vertices so all finite distances do not exceed $2dn$. Hence we can represent all those paths by creating a path of length $2dn \le 2n^2$ followed by a cycle of length $d \le n$. □

The above lemma gives a description of all paths going through a fixed vertex v. Hence we must consider all possible n choices for v and take the union of the representations found for all of them. As for each of them we create a path of the same length $2n^2$, we can share it among all representations, and then follow with a single nondeterministic choice to a set of n disjoint cycles. This construction works in time $\mathcal{O}(n^2 m) = \mathcal{O}(n^4)$ but it is not enough to match the bounds of the original proof: we must show that the combined size of all the cycles is at most n. Although this can be ensured in the above version, it is more convenient to give an improved algorithm which is faster by an order of magnitude and explicitly guarantees this property.

Theorem 1. *We can represent all accepting paths by a path of length $2n^2$ followed by a nondeterministic choice to a collection of disjoint cycles, with the combined size of all the cycles at most n. Such a representation can be found in $\mathcal{O}(n^3)$ time.*

Proof. To improve the running time of Lemma 1 we try to process vertices in groups instead of one-by-one. Take any simple cycle $v_0 \to v_1 \to \cdots \to v_d = v_0$. We will consider all paths going through at least one of the vertices on this cycle at once. Among all such paths of length ℓ with $\ell \equiv r \pmod{d}$ we need to find just the shortest one. This can be done by a similar construction as in Lemma 1, the only difference being that we create edge $x(r) \to y'((r+1) \bmod d)$ when $y = v_i$ for any $i = 0, 1, \ldots, d-1$. Then there is a path from q_0 to q_f of length ℓ going through the cycle if and only if the distance in the new graph to $q'_f(\ell \bmod d)$ is finite and does not exceed ℓ, so we can represent all such paths by a single path of length $2n^2$ followed by a cycle of length d. As this describes all possible paths going through the cycle, we can then delete all vertices $v_0, v_1, \ldots, v_{d-1}$ and repeat, as long as the graph is not acyclic. Let the lengths of the cycles found in successive iterations be $c_1, c_2, \ldots, c_t$. As they are all disjoint, $\sum c_i \le n$, so the whole complexity is $\sum_i \mathcal{O}(c_i m) = \mathcal{O}(nm) = \mathcal{O}(n^3)$. Also, the combined size of the cycles in the representation is at most n. □

Using the method from the above theorem we can also prove the following lemma. It will be an important tool in the subsequent sections.

Lemma 2. *We can represent all accepting paths going through at least one vertex contained in some cycle of length at most c by a path of length $2cn$ followed by a nondeterministic choice to a collection of disjoint cycles, with the combined size of all the cycles at most n. The representation can be found in $\mathcal{O}(n^3)$ time.*

Proof. We apply a slightly modified method from Theorem 1: as long as there exists a simple cycle of length at most c, we choose it and construct a representation of all accepting paths going through this cycle as a path of length at most $2cn$ followed by a cycle of length at most c. Then we remove the cycle and repeat. By combining the representations we get one path of length $2cn$ followed by a nondeterministic choice to a collection of disjoint cycles with combined size at most n.

To implement the above method efficiently, recall that given a vertex v we can find the shortest cycle containing v in linear time. Thus we can iterate through the vertices one-by-one, and if the shortest cycle containing the current vertex is of length c_i exceeding c, continue. Otherwise we find the representations of all paths going through at least one vertex from this cycle in $\mathcal{O}(c_i m)$ time and remove the cycle. As $\sum_i c_i \leq n$, the complexity is as claimed. □

As recently shown by Geffert [6], it is always possible to convert an automaton with into the Chrobak normal form so that the path consists of at most $n^2 - 2$ vertices, if $n > 1$. While Theorem 1 gives us a path of length $2n^2$, we can improve this bound to $n^2 - n$. Note that we assumed that there is just one accepting state and there are no edges incoming into q_0, and it might increase n by 2. This was just for the sake of simplicity: all above proofs can be modified to work even without such assumption.

Theorem 2. *We can represent all accepting paths by a path consisting of $n^2 - n$ vertices followed by a nondeterministic choice to a collection of disjoint cycles, with the combined size of all the cycles at most n, assuming $n > 1$. Such a representation can be found in $\mathcal{O}(n^3)$ time.*

Proof. We use the same method as in Theorem 1 but bound the shortest paths lengths more carefully. Assume that for some $0 \leq r < d$ the shortest path from $q_0(0)$ to $q'_f(r)$ contains more than $n^2 - n$ vertices. Then there must be a vertex v in the original graph which appears on this path at least n times. By cutting out parts between two occurrences of v, we get different shorter paths. There are two problems here: by cutting out parts we might remove all vertices from the chosen cycle of length d (and hence get a path to $q_f(r)$ instead of $q'_f(r)$), and we might get a different remainder modulo d of the resulting path length. The former can be removed by reserving one occurrence of v. Hence if $d \leq n - 2$, by the pigeonhole principle we can always find two occurrences such that the distance between them on the path is divisible by d. It remains to deal with the case of $d \geq n - 1$. d can be assumed to be the smallest cycle length possible.

Thus if $d = n$ the whole graph consists of just one cycle and the claim is obvious. The case of $d = n - 1$ is slightly more complicated. If the distance between two occurrences of some vertex is $n - 1$, we can shorten the path. Hence the path must of the form $v_1 \to v_2 \to \ldots \to v_n \to v_1 \to v_2 \to \ldots$. If its length exceeds $n^2 - n$, we can remove the first $n(n-1)$ vertices and get a shorter path with the same length modulo d. □

4 Application to regular expressions conversion

Given a NFA over a unary alphabet we would like to construct a small regular expression describing the same language. In the regular expression we are allowed to use concatenation, union and Kleene star. A straightforward construction gives an expression of size $\mathcal{O}(n^2)$. We will show that with some number theoretic insight this can be improved to $\mathcal{O}(\frac{n^2}{\log n})$. While the improvement is of only logarithmic magnitude, it requires combining a few ideas, and refutes the conjecture of Martinez who asked for a quadratic lower bound [9].

First we show that for some automata converting into the Chrobak normal form implies a quadratic blow-up. More precisely, we construct an infinite family of automata N_n on n states requiring such an increase in size after the conversion.

Lemma 3. *For any n there exists an automaton N_n on n states such that for any automaton M with $L(N_n) = L(M)$ consisting of a path followed by a nondeterministic choice into a collection of disjoint cycles, the path is of length $\Omega(n^2)$.*

Proof. Let N_n be an automaton consisting of two cycles of lengths $k = \lfloor \frac{n-1}{2} \rfloor$ and $k+1 = \lfloor \frac{n+1}{2} \rfloor$ sharing exactly one vertex, which is at the same time starting and final. Assume that we have M recognizing the same language $L(M) = \{a^{\alpha k+\beta(k+1)} : \alpha, \beta \geq 0\}$ which is in a normal form. We claim that its path is of length greater than $\ell = k(k+1) - k - (k+1)$. Assume otherwise. Note that ℓ is the greatest integer not belonging to $L(M)$. Let the lengths of all M's cycles be $c_1, c_2, \ldots, c_t$. Then $\ell + \mathrm{lcm}(c_1, c_2, \ldots, c_t) \in L(M)$ and so it must be accepted on a cycle of length c_i. But then ℓ corresponds to the same vertex on this cycle, and so belongs to $L(M)$, a contradiction. □

To overcome the quadratic increase we must use a stronger notion than the Chrobak normal form alone. For that to happen we split the set of all accepting paths into acyclic, *strongly cyclic*, and *weakly cyclic*. All paths of a given type will be represented separately as regular expressions of bounded size.

Definition 1. *A path is:*

1. strongly cyclic *if it contains a vertex v such that there is a cycle of length at most $\frac{n}{\alpha \log n}$ through v,*
2. weakly cyclic *if it is not strongly cyclic but contains a vertex belonging to some cycle,*
3. acyclic *otherwise.*

The constant α in the above definition is to be chosen later.

Lemma 4. *A regular expression of size $\mathcal{O}(n)$ describing all acyclic accepting paths can be constructed in $\mathcal{O}(nm)$ time.*

Proof. Computing the lengths of all acyclic accepting paths in the claimed complexity is trivial. To encode them in a regular expression, use the following simple trick: if $x_1 < x_2 < \ldots < x_k$ then $a^{x_1} + (\epsilon + a^{x_2 - x_1}(\ldots(\epsilon + a^{x_k - x_{k-1}})\ldots))$ generates exactly $\{x_1, x_2, \ldots, x_k\}$ and is of size $\mathcal{O}(x_k)$. □

Lemma 5. *A regular expression of size $\mathcal{O}(\frac{n^2}{\log n})$ describing all strongly cyclic accepting paths can be constructed in $\mathcal{O}(nm)$ time.*

Proof. To describe all strongly cyclic paths we can use the method from the previous section. More precisely, applying Lemma 2 with $c = \frac{n}{\alpha \log n}$ we get a path of length $2\frac{n^2}{\alpha \log n}$ followed by a set of cycles with the combined size at most n. This can be converted into a regular expression of size $\mathcal{O}(\frac{n^2}{\log n})$: apply the trick from Lemma 4 to encode the words accepted on the path, and then again for each of the cycles. □

We still have to construct an expression representing the weakly cyclic paths. Note that we have already described all strongly cyclic paths and so we can safely assume that the girth is at least $\frac{n}{\alpha \log n}$. Nonexistence of smaller cycles implies that there are at most $\alpha \log n$ nontrivial strongly connected components.

We split weakly cyclic paths into two groups. For that we define $C(v) = \{1 \leq \ell \leq n :$ there is a cycle through v of length $\ell\}$ and $C(S) = \bigcup_{v \in S} C(v)$.

Definition 2. *A weakly cyclic path going through strongly connected components $S_1, S_2, \ldots, S_k$ is:*

1. thin *if* $\left|\bigcup_{i=1}^{k} C(S_i)\right| \leq \beta \log n$,
2. fat *otherwise.*

The constant β will be chosen later. Using the above notion and defining the set of *nonnegative combinations* of positive integers $a_1, a_2, \ldots, a_n$ as $N(a_1, \ldots, a_n) = \{\sum_{i=1}^{n} x_i a_i : x_i \geq 0 \text{ for all } i\}$ we can establish a certain normal form of all accepting paths.

Definition 3. *Given an accepting path $v_0 \to v_1 \to \ldots \to v_\ell$ we define its* skeleton *of length ℓ' to be an accepting path $v'_0 \to v'_1 \to \ldots \to v'_{\ell'}$ such that $\ell - \ell' \in N\left(\bigcup_{i=0}^{\ell'} C(v'_i)\right)$.*

Note that a skeleton of a given accepting path can possibly go through completely different vertices than the original path. The only required condition is on its length and the set of cycles it intersects (which, again, does not have to be anyhow similar to the set of cycles the original path intersects).

Lemma 6. *Any accepting path $v_0 \to v_1 \to \ldots \to v_\ell$ has a skeleton of length at most $n + n\left|\bigcup_{i=0}^{\ell} C(v_i)\right|$.*

Proof. Let $C = \bigcup_{i=0}^{\ell} C(v_i)$ be the set of the lengths of all cycles having nonempty intersection with the path. For each element of $c \in C$ we mark the first vertex v_i such that $c \in C(v_i)$. As long as there exist $i < j$ such that $v_i = v_j$ and no vertex v_k with $i < k < j$ is marked we can cut out $v_{i+1}, v_{i+2}, \ldots, v_j$ obtaining a shorter accepting path with the same set C. If such pair of indices does not exists, the distance between any pair of marked vertices must be strictly smaller than n. Thus the total length ℓ' of the final path cannot exceed $n + n\,|C|$. Because it has been constructed by cutting out cycles, $\ell - \ell'$ can be represented as a nonnegative combination of elements of C. Thus this final path is a skeleton of claimed length. □

Lemma 7. *A regular expression of size $\mathcal{O}(n^{1+\alpha+\beta}\log n)$ describing all thin accepting paths can be constructed in polynomial time.*

Proof. We construct a separate expression for each possible choice of the set of strongly connected components $S_1, S_2, \ldots, S_k$ such that $\left|\bigcup_{i=1}^{k} C(S_i)\right| \leq \beta \log n$. Assume such fixed choice and remove all other strongly connected components.

We would like to generate all pairs (ℓ', C) such that there exists a skeleton of length ℓ' and a specified set of cycles $C' \subseteq C = \bigcup_{i=1}^{k} C(S_i)$. There are just n^β subsets of C and by Lemma 6 we can restrict our attention to $\ell' \leq n\,(1 + \beta \log n)$ so the maximum number of possible pairs is fairly small. To generate the pairs efficiently we define a new graph G' with the following vertices and edges:

$$V' = \{(v, X) : v \in V, X \subseteq C\}$$
$$E' = \{((u, X), (v, X \cup C(v))) : (u, v) \in E\}$$

It is easy to see that there exists a path from $(q_0, C(q_0))$ to (q_f, C') of length ℓ' in G' if and only if there exists a skeleton of length ℓ' and the set of cycles C'. Thus we can generate all valid pairs (ℓ', C') in polynomial time by computing paths of lengths not exceeding $n\,(1 + \beta \log n)$ from $(q_0, C(q_0))$ in G'. Then we consider all pairs $(\ell'_1, C'), (\ell'_2, C'), \ldots, (\ell'_k, C')$ with the same set of cycles C'. We can construct a regular expression of size $\mathcal{O}(n \log n)$ describing all paths with the corresponding skeletons using the trick from Lemma 4 to encode all $\ell'_1, \ell'_2, \ldots, \ell'_k$ and appending $(a^{c_1} + a^{c_2} + \ldots + a^{c_s})^*$ where $C' = \{c_1, c_2, \ldots, c_s\}$.

For fixed choices of the set of strongly connected components and C' we get a description of all thin paths of size $\mathcal{O}(n^{1+\beta}\log n)$. There are n^α choices possible so the total size is $\mathcal{O}(n^{1+\alpha+\beta}\log n)$. □

To deal with fat accepting paths we need to dig deeper into the structure of nonnegative combinations.

Lemma 8. *Let $a_1, a_2, \ldots, a_n$ be a set of different positive integers with $M = \max_i a_i$. Elements of $N(a_1, \ldots, a_n)$ greater than $2\frac{M^2}{n}$ are exactly the multiples of $\gcd(a_1, \ldots, a_n)$.*

Proof. The lemma follows from a result of Erdős and Graham [5]: if $0 < b_1 < b_2 < \ldots < b_n$ are integers with $\gcd(b_1, \ldots, b_n) = 1$ then the equation $N = \sum_{i=1}^{n} x_i b_i$ has a solution in nonnegative integers x_i provided that $N > 2b_{n-1}\left\lfloor \frac{b_n}{n} \right\rfloor - b_n$. Indeed, $N(a_1, \ldots, a_n)$ contains multiples of $d = \gcd(a_1, \ldots, a_n)$ and if $Nd > 2\frac{M^2}{n}$ we can define $b_i = \frac{a_i}{d}$ and by the aforementioned result $N = \sum_{i=1}^{n} x_i b_i$ has a solution in nonnegative integers, thus Nd belongs to $N(a_1, \ldots, a_n)$. □

Lemma 9. *If X is a set of positive integers, we can choose its subset $X' \subseteq X$ such that $\gcd(X) = \gcd(X')$ and $|X'| \leq \log \max_{x \in X} x$.*

Proof. Let $X = \{x_1, x_2, \ldots, x_s\}$. Start with $X' = \{x_1\}$, then for $i = 2, 3, \ldots, s$ check if $\gcd(X' \cup \{x_i\}) = \gcd(X')$ holds. If it does, continue. Otherwise add x_i to the current X'. Each time we add something to X', the value of $\gcd(X')$ decreases at least by a factor of 2, and the claim follows. □

Lemma 10. *A regular expression of size $\mathcal{O}(\frac{n^2}{\log n} + n^{1+\alpha})$ describing all fat accepting paths can be constructed in polynomial time.*

Proof. Choose a subset of strongly connected components $S_1, S_2, \ldots, S_k$ such that $\left|\bigcup_{i=1}^{k} C(S_i)\right| > \beta \log n$. If for some $i < j$ there is no path from S_i to S_j nor from S_j to S_i there exists no path hitting all those components at once and we take another subset. Otherwise we can sort all S_i topologically and compute $d = \gcd\left(\bigcup_{i=1}^{k} C(S_i)\right)$. Observe that $d \leq \frac{n}{\beta \log n}$ as a bigger value of d implies that there would be less than $\beta \log n$ different multiplies of d not exceeding n, and each element of $\bigcup_{i=1}^{k} C(S_i)$ must be a multiple of d. We compute for any $0 \leq r < d$ the smallest integer t_r such that there exists an accepting path of length $t_r d + r$ going through all $S_1, S_2, \ldots, S_k$. Note that we require that the path goes through each of those components. This can be done by constructing a new graph G' consisting of d copies of the original G:

$$V' = \{(v, r) : v \in V, 0 \leq r < d\}$$
$$E' = \{((u, r), (v, (r+1) \bmod d)) : (u, v) \in E\}$$

and repeating a multiple sources shortest path computation k times. First we compute the shortest paths from $(q_0, 0)$ ending in S_1 and avoiding all $S_2, S_3, \ldots, S_k$. Then, assuming we already have shortest paths visiting at least one vertex from each $S_1, S_2, \ldots, S_i$ and ending in S_i, we find the same information for $i + 1$ by a single multiple sources shortest paths computation in G'.

If $t_{\ell \bmod d}$ is not defined, there are no accepting paths of length ℓ, but the converse is not necessarily true, for at least two different reasons. First of all, by computing lengths modulo d we assumed that any multiple of d can be realized as a nonnegative combination of cycles. While it is true for sufficiently large multiples, we might need to combine cycles which are contained in some of the S_i but are completely disjoint with the path.

Assume $\ell > 2(1+\beta)n \log n + 3\frac{n^2}{\beta \log n}$ and $t_{\ell \bmod d}$ is defined. Then we can find an accepting path of length $\ell' \leq \ell$ visiting all components S_i such that $\ell' \leq \frac{n^2}{\beta \log n}$

and d divides $\ell - \ell'$ (because ℓ' is created by subtracting multiples of d from ℓ). By Lemma 9 we can choose a set of at most $\log n$ vertices $v_1, v_2, \ldots, v_s$ from the strongly connected components S_i such that $d = \gcd\left(\bigcup_{i=1}^{s} C(v_i)\right)$. Because the total number of different cycle lengths in all components $S_1, S_2, \ldots, S_k$ exceeds $\beta \log n$, we can also choose a set of at most $\beta \log n$ vertices $v'_1, v'_2, \ldots, v'_{s'}$ such that $\left|\bigcup_{i=1}^{k} C(v'_i)\right| \geq \beta \log n$ and $s' \leq \beta \log n$. By extending the path to hit all v_i and v'_i we can create another accepting path ℓ'' such that $\ell'' \leq \frac{n^2}{\beta \log n} + 2(1+\beta)n \log n$, d divides $\ell'' - \ell$ and the path visits all vertices v_i and v'_i. Hence there exists a collection of cycles $c_1, c_2, \ldots, c_t$ having nonempty intersections with this new path of length ℓ'' such that $\gcd(c_1, c_2, \ldots, c_t)$ divides $\ell - \ell'$ and $t \geq \beta \log n$. Then by Lemma 8 the new path is a skeleton of the original path so there exists an accepting path of length $\ell > 2(1+\beta)n \log n + 3\frac{n^2}{\beta \log n}$ if and only if $t_{\ell \bmod d}$ is defined.

By repeating the above reasoning for all choices of strongly connected components we get a succinct description of all fat accepting paths: we can compute a collection of sets $R_1, R_2, \ldots, R_{n^\alpha}$ such that there is such path of length ℓ exceeding $2(1+\beta)n \log n + 3\frac{n^2}{\beta \log n}$ if and only if $\ell \bmod d_i \in R_i$ for some $1 \leq i \leq n^\alpha$. Thus we can construct a regular expression of size $\mathcal{O}(\frac{n^2}{\beta \log n} + n^{1+\alpha})$ describing all such paths by considering lengths smaller or equal and greater than $2(1+\beta)n \log n + 3\frac{n^2}{\beta \log n}$ separately. We write down the former explicitly and to deal with the latter we take a union of the expressions describing all R_i concatenated with $a^{\left\lceil 2(1+\beta)n \log n + 3\frac{n^2}{\beta \log n} \right\rceil}$ which is shared among all i. □

By choosing $\alpha + \beta < 1$ and combining Lemma 4, 5, 7 and 10 we get:

Theorem 3. *A regular expression of size $\mathcal{O}(\frac{n^2}{\log n})$ describing all accepting paths can be constructed in polynomial time.*

5 Application to context-free grammar conversion

Given a NFA M over a unary alphabet we would like to construct a small context-free grammar describing the same language. The grammar should be in Chomsky normal form (and thus we relax the problem a little bit by assuming that the empty word is not accepted by M), and we would like to minimize the number of nonterminals. An application of Chrobak normal form results in $\mathcal{O}(n^{2/3})$ bound [3]. In this section we develop a substantially more efficient conversion procedure requiring just $\mathcal{O}(\sqrt{n \log n})$ nonterminals. We start with a simple combinatorial lemma.

Lemma 11. *Given a collection of t sets $A_1, A_2, \ldots, A_t \subseteq U$ we can efficiently find $B \subseteq U$ of cardinality at most $\frac{|U|}{s} \lg t$ such that $A_i \cap B \neq \emptyset$ for all i, where $s = \min_i |A_i|$.*

Proof. We use a simple greedy method: start with $B = \emptyset$ and as long as there exists A_i disjoint with B, select $x \notin B$ maximizing $|\{i : A_i \cap B = \emptyset \text{ and } x \in A_i\}|$.

Let $t = t_0, t_1, t_2, \ldots, t_k \geq 1$ be the cardinalities of $\{i : A_i \cap B = \emptyset\}$ in successive steps. We claim that $t_{i+1} \leq t_i - t_i \frac{s}{|U|}$: there are t_i sets left, each of them contains at least s elements, thus there exists x belonging to at least $t_i \frac{s}{|U|}$ sets. Now observe that the claim implies $t_k \leq t \left(1 - \frac{s}{|U|}\right)^k$. Setting $k = \frac{|U|}{s} \lg t$ yields:

$$1 \leq t_k \leq t \left(1 - \frac{1}{\frac{|U|}{s}}\right)^{\frac{|U|}{s} \lg t} < t \left(\frac{1}{e}\right)^{\lg t} = 1$$

so the method terminates after the k-th step, which gives the lemma. □

We give two different conversion methods, one appropriate for large girth graphs and one which efficiently describes all paths going through at least one vertex contained in a short cycle. Consider the representation found using Lemma 2 (with some c to be chosen later). We call the part of $L(M)$ accepted on the path of length $2cn$ *finite* while and the part accepted on the collection of cycles *infinite*. Dealing with the infinite part is relatively simple, no matter what c is.

Lemma 12. *A context-free grammar with $\mathcal{O}(\sqrt{n})$ nonterminals describing the infinite part of $L(M)$ can be constructed in polynomial time.*

Proof. Let $c_1 < c_2 < \ldots < c_\ell$ be lengths of the cycles in the Chrobak normal form. As $\sum_i c_i \leq n$, ℓ is at most $\sqrt{n}$. First we introduce $\mathcal{O}(\sqrt{n})$ nonterminals $X_0, X_1, \ldots, X_{\lfloor\sqrt{n}\rfloor}$ and $Y_0, Y_1, Y_2, \ldots, Y_{\lfloor\sqrt{n}\rfloor}$ such that X_i derives a^i and Y_i derives $a^{i\lfloor\sqrt{n}\rfloor}$. Using those nonterminals we can express any a^k as $X_i Y_j$ as long as k is at most n. Then we introduce ℓ nonterminals $C_1, C_2, \ldots, C_\ell$ such that C_i describes all words accepted on the i-th cycle. For that we first define D_i which derives exactly a^{c_i} and add production $C_i \to C_i D_i$. Then for any a_k which is accepted on this cycle we add production $C_i \to X_{k \bmod \lfloor\sqrt{n}\rfloor} Y_{\lfloor k/\lfloor\sqrt{n}\rfloor\rfloor}$. We combine all C_i by introducing a special nonterminal P which derives a^{2cn} (this can be done by introducing a logarithmic number of new nonterminals) and productions $S \to PC_i$ for all $i = 1, 2, \ldots, \ell$. □

First we show how to represent all accepting paths going through at least one vertex contained in a short cycle.

Lemma 13. *A context-free grammar with $\mathcal{O}\left(\sqrt{n} + (cn)^{1/3}\right)$ nonterminals describing accepting paths going through at least one vertex contained in a cycle of length at most c can be constructed in polynomial time.*

Proof. First apply Lemma 2 with a given c. By Lemma 12 we can represent the infinite part using $\mathcal{O}(\sqrt{n})$ nonterminals. By Lemma 2.1 of [3] any finite unary language L can be represented as a context-free grammar with $\mathcal{O}(m^{1/3})$ nonterminals, where $m = \max_{w \in L} |w|$, and the construction can be trivially implemented in $\mathcal{O}(m)$ time. We apply this lemma to the finite part, which contains words of length at most $m = 2cn$, so the claimed bound follows. □

Note that after applying the above lemma we can assume that there are no short cycles in the graph.

Lemma 14. *If $b \le g$, a context-free grammar with $\mathcal{O}\left(\sqrt{n} + \frac{n}{b}\log n + b\right)$ nonterminals describing all accepting paths can be constructed in polynomial time, where g is the girth of the underlying graph.*

Proof. Apply Theorem 1 and use Lemma 12 to describe the infinite part of the language found while introducing $\mathcal{O}(\sqrt{n})$ nonterminals. Let its finite part be $\{a^{x_1}, a^{x_2}, \ldots, a^{x_s}\}$ ($s \le 2n^2$). Choose any accepting computation for each a^{x_i}. Each computation corresponds to a path from q_0 to q_f of length $\ell \le 2n^2$. Take its prefix of length $\ell - \ell \bmod b$ and split it into blocks of consecutive b states. For each such block we create one set containing the corresponding states. Observe that because $b \le g$ the states in a single block do not repeat. Then by Lemma 11 we can choose $Q' \subseteq Q$ such that $|Q'| \le \frac{n}{b}\lg\frac{(2n^2)^2}{b} = \mathcal{O}(\frac{n}{b}\log n)$ and any block contains at least one element of Q'. We add q_0, q_f to Q' and create one nonterminal A_q for any $q \in Q'$. Then we create $2b$ nonterminals $B_1, B_2, \ldots, B_{2b}$ such that B_k derives a^k and for any $q, q' \in Q'$ and $k \le 2b$ we add production $A_q \to B_k A_{q'}$ whenever there is a path from q to q' of length k. The total number of introduced nonterminals is $\mathcal{O}(\frac{n}{b}\log n + b)$. Now observe that if we make A_{q_0} the starting state and add production $A_{q_f} \to \epsilon$, any a^{x_i} can be derived in the resulting grammar. Indeed, consider Figure 2: at least one state from each block belongs to Q', and we can jump between two adjacent blocks using nonterminals B_k. Furthermore, any word derived in the grammar corresponds to an accepting computation of M. The epsilon production can be removed without creating any new nonterminals. □

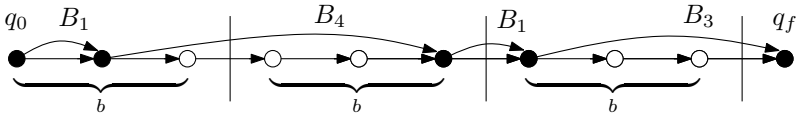

Fig. 2: Jumping between blocks using states from Q'.

Combining the two above lemmas gives the claimed bound.

Theorem 4. *A context-free grammar with $\mathcal{O}\left(\sqrt{n\log n}\right)$ nonterminals describing $L(M)$ can be constructed in polynomial time.*

Proof. We apply Lemma 13 to describe all accepting paths going through at least one vertex contained in a cycle of length at most $g = \sqrt{n\log n}$. This requires $\mathcal{O}((gn)^{1/3}) = \mathcal{O}(\sqrt{n\log^{1/3} n}) = \mathcal{O}(\sqrt{n\log n})$ nonterminals. Then we remove all such vertices, which leaves us with a graph of girth at least g, so by Lemma 14 with $b = \sqrt{n\log n}$ we can construct a context free grammar with $\mathcal{O}(\frac{n}{b}\log n + b) = \mathcal{O}(\sqrt{n\log n})$ nonterminals describing all remaining accepting paths. □

6 Acknowledgments

An intriguing open problem is to investigate if the $\mathcal{O}(\frac{n^2}{\log n})$ upper bound on the equivalent regular expression can be substantially improved, or to find a matching lower bound. a natural line of attack for the lower bound would be to count the number of distinct languages accepted by a n-state NFA. This turns out to be $\mathcal{O}(n \log n)$, though, which is not really helpful. As for the equivalent context-free grammar, the best known lower bound is $\Omega(n^{1/3})$. We believe that the true complexity of such question is $\Theta(\sqrt{n})$.

I am undoubtedly grateful to Tomasz Jurdziński and Artur Jeż for a thorough reading of an initial version of this paper and many helpful remarks.

References

1. Chrobak, M.: Finite automata and unary languages. Theor. Comput. Sci. 47, 149–158 (November 1986)
2. Chrobak, M.: Errata to: "finite automata and unary languages". Theor. Comput. Sci. 302, 497–498 (June 2003)
3. Domaratzki, M., Pighizzini, G., Shallit, J.: Simulating finite automata with context-free grammars. Information Processing Letters 84(6), 339 – 344 (2002)
4. Ehrenfeucht, A., Zeiger, P.: Complexity measures for regular expressions. In: Proceedings of the sixth annual ACM symposium on Theory of computing. pp. 75–79. STOC '74, ACM, New York, NY, USA (1974)
5. Erdős, P., Graham, R.: On a linear diophantine problem of Frobenius. Acta Arith 21, 399–408 (1972)
6. Geffert, V.: Magic numbers in the state hierarchy of finite automata. Inf. Comput. 205, 1652–1670 (November 2007)
7. Gruber, H., Holzer, M.: Finite automata, digraph connectivity, and regular expression size. In: Proceedings of the 35th international colloquium on Automata, Languages and Programming, Part II. pp. 39–50. ICALP '08, Springer-Verlag, Berlin, Heidelberg (2008)
8. Martinez, A.: Efficient computation of regular expressions from unary NFAs. pp. 174–187. DFCS '02 (2002)
9. Martinez, A.: Topics in formal languages: String enumeration, unary NFAs and state complexity. M. Math Thesis, University of Waterloo (2002)
10. Meyer, A.R., Fischer, M.J.: Economy of description by automata, grammars, and formal systems. In: Proceedings of the 12th Annual Symposium on Switching and Automata Theory (swat 1971). pp. 188–191. IEEE Computer Society, Washington, DC, USA (1971)
11. Parikh, R.J.: On context-free languages. J. ACM 13, 570–581 (October 1966)
12. Sawa, Z.: Efficient construction of semilinear representations of languages accepted by unary NFA. In: Proceedings of the 4th international conference on Reachability problems. pp. 176–182. RP'10, Springer-Verlag, Berlin, Heidelberg (2010)
13. To, A.W.: Unary finite automata vs. arithmetic progressions. Information Processing Letters 109(17), 1010 – 1014 (2009)