IFIG Research Report

Institut für Informatik

Regular Expressions From Deterministic Finite Automata, Revisited

Hermann Gruber Markus Holzer

IFIG Research Report 1403
May 2014

Institut für Informatik
JLU Gießen
Arndtstraße 2
35392 Giessen, Germany
Tel: +49-641-99-32141
Fax: +49-641-99-32149
mail@informatik.uni-giessen.de
www.informatik.uni-giessen.de

IFIG Research Report
IFIG Research Report 1403, May 2014

Regular Expressions From Deterministic Finite Automata, Revisited

Hermann Gruber[1]

knowledgepark AG
Leonrodstraße 68, 80636 München, Germany

and

Markus Holzer[2]

Institut für Informatik, Universität Giessen
Arndtstraße 2, 35392 Giessen, Germany

Abstract. We continue our research on the conversion of deterministic finite automata to equivalent regular expressions from [H. Gruber, M. Holzer: Provably Shorter Regular Expressions from Deterministic Finite Automata. In: 12*th DLT*, Number 5257 of LNCS, 2008], where it was shown that every language over a binary alphabet accepted by an n-state *deterministic* finite automaton can be converted into an equivalent regular expression of alphabetic width at most $O(1.742^n)$. This result is based on a graph theoretic argument that a transition digraph with bounded average (undirected) degree has a large size induced subdigraph of tree-width 2. Recently this upper bound was improved to $O(1.682^n)$ by [K. Edwards, H. Farr: Improved Upper Bounds for Planarization and Series-Parallelization of Degree-Bounded Graphs. *Elec. J. Comb.*, 19(2), 2012] using a similar argument on induced subdigraphs of tree-width 3. Here we significantly improve this bound to $O(1.588^n)$ by taking care of the average *directed* degree, proving that every digraph with bounded average outdegree has a large induced subdigraph of small cycle rank, which can be advantageously transformed. To this end, we develop fragments of a theory on d-outdegenerate digraphs.

Categories and Subject Descriptors: F.1.1 [**Computation by Abstract Devices**]: Models of Computation—*Automata*; F.4.3 [**Mathematical Logic and Formal Languages**]: Formal Languages—*Classes defined by grammars or automata*; I.1.1 [**Symbolic and Algebraic Manipulation**]: Expressions and Their Representation—*Simplification of expressions*;

Additional Key Words and Phrases: regular expressions, finite automata, cycle rank, digraph measures, outdegenerated digraph

[1]E-mail: hermann.gruber@knowledgepark-ag.de
[2]E-mail: holzer@informatik.uni-giessen.de

1 Introduction

One of the most popular algorithms for the conversion of finite automata into equivalent regular expressions is the state elimination algorithm [27], whose performance crucially relies on the elimination ordering of the states. This algorithm is one of the few classical ones, see, e.g., [25], for converting finite automata into equivalent regular expressions, which all look different at first glance, but, as Sakarovitch [24] pointed out, are more or less a reformulation of the algorithmic idea of state elimination. The drawback of all these algorithms is that they return expressions of exponential size in the worst case, and in fact they are doomed to do so by a result of Ehrenfeucht and Zeiger [9], who exhibited a family of languages for which an exponential blow-up is inevitable. The rough upper bound of all of these algorithms is $O(4^n)$ on the size of the resulting regular expression, where n is the number of states of the given finite automaton. Nevertheless, the desire to obtain shorter regular expressions can be traced back to the work by McNaughton and Yamada [22], who already noticed the above mentioned influence of the ordering in which the states of a given automaton are processed.

Proving size bounds on the conversion of finite automata into equivalent regular expressions is challenging, and is the subject of active research, see, e.g., [8, 10, 14, 16, 17]. In particular, in [16] it is shown that deterministic finite automata with binary input alphabet can be converted into regular expressions of size at most $O(1.742^n)$. There, the main technical result concerns the problem of converting finite automata into regular expressions, parametrized by the *undirected* cycle rank. The size bound is then derived by bounding the undirected cycle rank of those. This has subsequently been improved to $O(1.682^n)$ using the same method, by refining the bound on undirected cycle rank in [8]. In this paper, we harness the theory of digraphs in order to get a further improved algorithm. The proof has two key ingredients, as with the original algorithm presented in [16]. First, we provide a nontrivial extension of the technique from [16] for converting finite automata into regular expressions, which relates cycle rank and alphabetic width. The theorem reads as follows:

Theorem 1. *Let $L \subseteq \Sigma^*$ be a regular language, and let r be a positive integer. If L is accepted by an n-state nondeterministic finite automaton having cycle rank at most r, then*

$$\operatorname{alph}(L) \leq |\Sigma| \cdot n^{r \cdot O(\log n)}.$$

Intuitively, the cycle rank of a finite automaton measures the nesting depth of its directed cycles and the alphabetic width $\operatorname{alph}(L)$ of a language L is the minimum size among all regular expressions describing the language L—if necessary, we provide formal definitions of these terms in the forthcoming sections. Throughout the paper we use standard graph theory as well as formal language and automata theory notation. See, e.g., [3] and [18].

For the second ingredient, we need to develop fragments of a theory on k-outdegenerate digraphs. This allows us to show that every digraph of small average outdegree has a large induced subdigraph of small cycle rank. DAG-width and Kelly-width are recently introduced digraph width measures [5, 19].

Theorem 2. *Let D be a digraph of order n with average outdegree $d \geq 2$. Then D has an induced subdigraph of order at least $\frac{2}{d+1}n$, which is*

1. *of DAG-width at most 1,*
2. *of Kelly-width at most 2, and*
3. *of cycle rank $O(\log n)$.*

Now following the basic strategy developed in [16], one can split up the state set of the automaton into an "easy" and a "hard" part. The regular expression size for converting the easy part is governed by the bound in Theorem 1. Converting the hard part by state elimination yields an exponential blow-up in the size of this part; but by Theorem 2, we can ensure that the hard part of the input for the conversion is sufficiently small. We will thus obtain our main result.

Theorem 3. *Let L be a regular language over a k-ary alphabet. If L is accepted by an n-state deterministic finite automaton, then*

$$\operatorname{alph}(L) \leq k \cdot 4^{\frac{k-1}{k+1}n} \cdot n^{O(\log n)^2}.$$

In the case of binary alphabets, the above given theorem improves the best previously known bound of $O(1.682^n)$ from [8] to $O(1.587^n)$.

2 Cycle Rank and Regular Expression Size

In this section, we derive Theorem 1, which gives a relation between cycle rank of finite automata and size of equivalent regular expressions. The *cycle rank* $r(D)$ of a digraph D is defined inductively as follows: if D is acyclic, then $r(D) = 0$; if D is strongly connected, then $r(D) = 1 + \min_{v \in V} r(D - v)$; otherwise, the cycle rank of D equals the maximum cycle rank among the strongly connected components (SCCs) of D. The *size*, or *alphabetic width*, of a regular expression r over the alphabet Σ, denoted by $\operatorname{alph}(r)$, is defined as the total number of occurrences of letters of Σ in r. For a regular language L, we define its alphabetic width, $\operatorname{alph}(L)$, as the minimum alphabetic width among all regular expressions describing L.

We aim at extending the methods from [16], which were based on undirected cycle rank, such that they work on digraphs. But moving from graphs to digraphs entails a nontrivial complication in our case: the result from [16] yields regular expressions of polynomial size for automata of bounded undirected cycle rank. Bounding the directed cycle rank no longer gives us such a powerful tool: already for acyclic finite automata, which are of cycle rank 0, a superpolynomial lower bound of $2^{\Omega(\log n)^2}$ on required regular expression size is known [17]. In order to address this apparent difficulty, we collect a few preliminary results first.

The set of walks in a digraph D connecting a vertex s to a vertex t can naturally be interpreted as a language, where the arc set of D serves as the alphabet. Then an elementary result connecting the theory of digraphs with formal language theory is the fact that this set of walks is a regular language [12, Chapter V.5]. We introduce some notation about sets of walks. Let $D = (V, A)$ be a digraph, s, t be vertices in V, and U such that $\emptyset \subseteq U \subseteq V$. We define the

language $L_{st}(D[U])$ of all s-t-walks in $D[U]$ as follows: for every arc $(i,j) \in A$, we introduce an alphabet symbol a_{ij}. Then $L_{st}(D[U])$ denotes the set of all walks $a_{i_0i_1}a_{i_1i_2}\cdots a_{i_{k-2}i_{k-1}}a_{i_{k-1}i_k}$ in $D[U]$ that start in $i_0 = s$, end in $i_k = t$, and where all internal vertices $i_1, i_2, \ldots, i_{k-1}$ of the walk are from U—note that s and t are not necessarily members of U. For $U = V$ we simply write $L_{st}(D)$ instead of $L_{st}(D[V])$. Next, we collect some easy observations about walks in digraphs. The first lemma concerns walks that start and end at the same vertex.

Lemma 4. *Let D be a digraph. For $v \in V$, let $C(v)$ denote the strongly connected component of D that contains v. Then $L_{vv}(D) = L_{vv}(D[C(v)])$.* □

The second observation concerns arbitrary s-t-walks in a digraph.

Lemma 5. *Let D be a digraph, and let w be an s-t-walk in D. Then w can be decomposed as $w = x_1a_{v_1v_2}x_2a_{v_2v_3}\cdots x_{k-1}a_{v_{k-1}v_k}x_k$, such that*

- *$v_1, v_2, \ldots, v_k$ is some ordering of a subset of the vertices that occur in w, and*
- *x_i is in $L_{v_iv_i}(D)$ for $1 \leq i \leq k$.* □

Observe that in the above decomposition, some of the x_i may be identical to the empty word, thus denoting an empty walk from v_i to v_i. We also need an estimate for the size of a regular expression denoting all s-t-walks of bounded length in a digraph. The following two results are essentially rephrasings of [10, Theorem 20] and [10, Corollary 22].

Theorem 6. *Let $D = (V, A)$ be a digraph of order n, and let s, t be two vertices. For each integer $\ell \geq 1$, the set $L_{st}(D)^{\leq \ell}$ can be described by a regular expression of alphabetic width at most $(n+1) \cdot \ell^{\log n+1}$.*

The next corollary deals with acyclic digraphs and their induced language.

Corollary 7. *Let $D = (V, A)$ be an acyclic digraph of order $n \geq 2$, and let s, t be two vertices. Then the set $L_{st}(D)$ can be described by a regular expression of alphabetic width at most $(n+1) \cdot (n-1)^{\log n+1}$.*

Now we are ready to prove Theorem 1, which was mentioned in the introduction, and states that $\mathrm{alph}(L) \leq |\Sigma| \cdot n^{r \cdot O(\log n)}$, if L is accepted by an n-state nondeterministic finite automaton having cycle rank at most r. This upper bound immediately follows from the next lemma.

Lemma 8. *Let $D = (V, A)$ be a digraph of order n, let $r \leq n$ be a nonnegative integer, and let s, t be two vertices in D. If D has cycle rank at most r, then*

$$\mathrm{alph}(L_{st}(D)) \leq \begin{cases} 4^r g(n)^{\max(r-1,0)} h(n), & \text{if } D \text{ is strongly connected} \\ 4^r g(n)^r h(n), & \text{otherwise,} \end{cases}$$

with $g(n) = (n+1) \cdot (2n-1)^{\log n+1}$ and $h(n) = (n-1)^{\log n+1}$.

Proof. The bounds are proven by lexicographic induction on (r, n). For $r = 0$, we have $\mathrm{alph}(L_{st}(D)) \leq h(n)$ by Corollary 7. For $r \geq 1$, we carry out the the induction step. There we distinguish two cases:

1. The digraph D is strongly connected. Then D contains a vertex v such that $D - v$ has cycle rank at most $r - 1$. The McNaughton-Yamada equation [22] for eliminating state v yields

$$L_{st}(D) = L_{st}(D - v) \cup L_{sv}(D - v) \cdot L_{vv}(D - v)^* \cdot L_{vt}(D - v).$$

Since $D - v$ is of order $n - 1$, we can apply the induction hypothesis to each language on the right-hand-side. We obtain

$$\begin{aligned} \operatorname{alph}(L_{st}(D)) &\leq 4 \cdot \left(4^{r-1} g(n-1)^{r-1} h(n)\right) \\ &\leq 4^r g(n)^{r-1} h(n). \end{aligned}$$

This completes the induction step for this case.

2. The digraph D has multiple strongly connected components. Let D' be the digraph obtained from D by adding a self-loop to each vertex. Then Lemma 5 shows that each walk w in the set $L_{st}(D)$ can be decomposed as $w = x_1 a_{v_1 v_2} x_2 a_{v_2 v_3} \cdots x_{k-1} a_{v_{k-1} v_k} x_k$, with $k \leq n$, and each x_i in $L_{v_i v_i}(D)$, for $1 \leq i \leq k$. Thus, the set $L_{st}(D)$ can be obtained from $L_{st}(D')^{\leq 2n-1}$ by applying the substitution map $\sigma : a_{vv} \mapsto L_{vv}(D)$. Hence, we have $L_{st}(D) = \sigma\left(L_{st}(D')^{\leq 2n-1}\right)$. It remains to estimate the size of the regular expression obtained by this construction. By Lemma 4 holds $L_{vv}(D) = L_{vv}(D[C(v)])$, where $C(v)$ denotes the SCC containing v, so we have the equality $\sigma(a_{vv}) = L_{vv}(D[C(v)])$. Since D has multiple SCCs, each set $C(v)$ is of cardinality smaller than n. So we can apply the induction hypothesis to estimate the alphabetic width of $L_{vv}(D[C(v)])$ for each such vertex v. In this way, we obtain

$$\begin{aligned} L_{vv}(D[C(v_i)]) &\leq 4^r g(|C(v_i)|)^{r-1} h(|C(v_i)|) \\ &\leq 4^r g(n)^{r-1} h(n). \end{aligned}$$

By Lemma 6, the set $L_{st}(D')^{\leq 2n-1}$ admits a regular expression of size at most $g(n)$. Applying the substitution σ to such an expression will blow up its size by a factor of at most $4^r g(n)^{r-1} h(n)$. Thus we have

$$\begin{aligned} \operatorname{alph}(L_{st}(D)) &= \operatorname{alph}\left(\sigma\left(L_{st}(D')^{\leq 2n-1}\right)\right) \\ &\leq g(n) \cdot 4^r g(n)^{r-1} h(n), \end{aligned}$$

and the induction step is completed also for this case. □

A similar bound was previously known for automata parametrized by *undirected* cycle rank [16]. On the one hand, the previous result yields regular expressions of polynomial size for automata of undirected cycle rank in $O(1)$. On the other hand, there are many digraphs of directed cycle rank in $O(1)$ but undirected cycle rank in $\Omega(n)$. So Theorem 1 yields a quasipolynomial bound for a much larger class of finite automata.

3 Outdegeneracy and Digraph Width Measures

Since computing the treewidth of undirected graphs is NP-hard, several lower bound methods have been devised, which are computationally easy. These can serve as combinatorial bounds in proofs, or to speed up branching algorithms for computing the treewidth. The degeneracy of a graph counts among the easiest and most practical lower bounds on its treewidth [6]. In the following, we generalize this result to digraphs, by giving a corresponding lower bound on the DAG-width and Kelly-width of digraphs in terms of outdegeneracy.

In the theory of undirected graphs, the degeneracy of a graph is a measure for the sparseness of a graph [11, 21]. Several generalizations of this notion to the case of digraphs have been proposed in [4]; we deliberately choose one of these, namely outdegeneracy, and take a closer look at it.

Let $D = (V, A)$ be a digraph. In the following, let $d^+(v)$ and $d^-(v)$ denote the outdegree and indegree of vertex v, respectively. A digraph D is *k-degenerate*, if every induced subdigraph of D contains a vertex with at most k different neighbors (that is, in- or out-neighbors) in that subdigraph. The *degeneracy* of D, denoted by $\kappa(D)$, is defined as the smallest integer such that D is k-degenerate. In a similar vein, a digraph D is *k-outdegenerate*, if every induced subdigraph of D has a vertex of outdegree at most k. The *outdegeneracy* of D, denoted by $\kappa^+(D)$, is defined in an analogous manner to the degeneracy of the digraph D. Both measures can be computed in linear time [4]. It is obvious from the definitions that $\kappa^+(D) \leq \kappa(D)$. Furthermore, for symmetric digraphs, the two measures coincide. Also, a digraph D is 0-outdegenerate if and only if D is acyclic. Finally, a straightforward induction on the order n of the digraph shows that the number of arcs in a k-outdegenerate digraph is at most $kn(n-1)/2$.

We turn to the definition of DAG-with and Kelly-width of a digraph. The *cops and visible robber game*, as defined in [5], is given as follows: let $D = (V, A)$ be a digraph. Initially, the cops occupy some set of $X \subseteq V$ vertices, with $|X| \leq k$, and the robber is placed on some vertex $v \in V \setminus X$. At any time, some of the cops can reside outside the graph, say, in a helicopter. In each round, the cop player chooses the next location $X' \subseteq V$ for the cops. The stationary cops in $X \cap X'$ remain in their positions, while the others go to the helicopter and fly to their new position. During this, the robber player, knowing the cops' next position X' from wire-tapping the police radio, can run at great speed to any new position v', provided there is a (possibly empty) directed path from v to v' in $D-(X\cap X')$, i.e., he has to avoid to run into a stationary cop, and to run along a path that is reachable from his current position. Afterwards, the helicopter lands the cops at their new positions, and the next round starts, with X' and v' taking over the roles of X and v, respectively. The cop player wins the game if the robber cannot move any more, and the robber player wins if the robber can escape indefinitely. In the *cop-monotone* variant of the game, the cops are not allowed to revisit a vertex once it has been vacated; in the *invisible robber* variant, the robber is invisible for the cops; and in the *inert robber* variant, the robber is only allowed to move if a cop is about to land on the robber's current position. The *DAG-width* of a digraph D is the minimum k such that k cops have a winning strategy in the cop-monotone variant if and only if the DAG-

width of D is at most k [5]. Similarly, the *Kelly-width* of D is the minimum k such that $k-1$ cops have a winning strategy in the cop-monotone, inert invisible robber variant [19].

The outdegeneracy of a digraph serves as a common lower bound for these two measures:

Theorem 9. *Let D be a digraph with $\kappa^+(D) \geq k$. Then the DAG-width of D is at least k, and the Kelly-width of D is at least $k+1$.*

Proof. We prove in fact a slightly more general statement, in terms of the cops and visible robber game: let D be a digraph with $\kappa^+(D) \geq k$. Then an inert robber has a winning strategy against k cops in the cops and visible robber game on D.

In analogy to the undirected case, we define the *k-outcore* of a graph as the maximal induced subgraph in which every vertex has outdegree at least k. This maximal induced subgraph is in fact unique [4]: it is obtained in a greedy manner, by iteratively removing a vertex of minimum outdegree from the current subdigraph, until all remaining vertices have outdegree at least k. The k-outcore of a digraph D is nonempty if and only if $\kappa^+(D) \geq k$. The strategy for the robber is to stay in the k-outcore of D. The robber only moves if a cop is about to land on his current position v. In this case at most $k-1$ out-neighbors of v can remain occupied by some cop. By definition, at least k out-neighbors of v belong to the k-outcore of D, so the robber can flee to an unoccupied out-neighbor without leaving the k-outcore. This strategy allows the robber to escape indefinitely.

The theorem now follows as a corollary to the claim we have just proved: clearly, if an inert visible robber has a winning strategy against k cops, the same holds *a forteriori* for variants of the game where the cop player is less powerful (e.g., cop-monotone, invisible robber), or where the robber player is more powerful, (e.g., non-inert). □

Unfortunately, there is no easy converse of this inequality: we recall from [20] a family of digraphs of DAG-width and Kelly-width $\Omega(\sqrt{n})$, which is are easily observed to be 2-outdegenerate. Here, n denotes the order of the respective digraph. More precisely, the digraphs J_k defined in [20] each admit a planar drawing as the union of k concentric, equally directed, $2k$-cycles, which are connected to each other by $2k$ radial k-paths, the first k of which are directed inwards, while the remaining k are directed outwards; see Figure 1 for illustration. The order of J_k is $n = 2k^2$. Nevertheless, we can obtain a tight characterization if the outdegeneracy is at most 1.

Lemma 10. *Let D be a digraph and let $k \in \{0, 1\}$. Then the following are equivalent:*

1. *D is k-outdegenerate,*
2. *D has DAG-width at most k, and*
3. *D has Kelly-width at most $k+1$.*

Proof. The digraph D is 0-outdegenerate iff D is acyclic iff the Kelly-width of D is 1 [19] iff the DAG-width of D is zero [5].

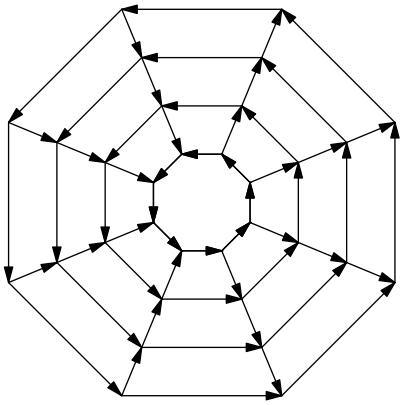

Fig. 1. A drawing of the digraph J_4. In general, J_k is 2-outdegenerate but has Kelly-width in $\Omega(k)$.

The case $k = 1$ is more interesting. We consider first the Kelly-width. By Theorem 9, a digraph of Kelly-width 2 has outdegeneracy 1, so it remains to show the reverse direction. To this end, we recall the characterization of digraphs of Kelly-width at most 2 in terms of arc contractions [23]: a digraph has Kelly-width at most 2 if and only if it can be converted to the empty digraph by repeatedly (i) removing a vertex of outdegree 0, or (ii) locating a vertex of outdegree 1 in the current digraph, followed by contracting the single outgoing arc of that vertex. Assume D is 1-outdegenerate. We will prove by induction on the order n of the digraph that D has Kelly-width at most 2. The base case $n = 0$ is trivial. For $n \geq 1$, let v be a vertex of outdegree at most 1 in D. Then contracting the arc leaving v does not affect the outdegree of any vertex in $V \setminus \{v\}$. Let D' denote the digraph thus obtained. Since $d^+_{D'}(w) = d^+_D(w)$ for all vertices w in $V \setminus \{v\}$, the digraph D' is again 1-outdegenerate. Applying the induction hypothesis, D' has Kelly-width at most 2, and D' can be reduced to the empty digraph in the desired fashion. This completes the proof of the reverse direction for Kelly-width.

Regarding DAG-width, it is known [19] that digraphs of Kelly-width at most 2 (and hence outdegeneracy at most 1) have DAG-width at most 1. For the reverse implication, Kelly-width greater than 2 implies outdegeneracy greater than 1, and thus, with the aid of Theorem 9, also DAG-width greater than 1. □

These characterizations show that the 1-outdegenerate digraphs form a robust family of digraphs. Let us remark that, as the width parameters grow, these concepts diverge very soon: there are digraphs of DAG-width 4 and Kelly-width 3, see [19].

4 Large Induced Subdigraphs of Small Directed Width

We now turn to the investigation of width measures on sparse digraphs. The following known results illustrate the kind of bounds we are aiming at: each undirected graph of average degree $d \geq 4$ contains an induced subgraph of order

at least $\frac{d-17/18}{d+1}n$ and treewidth at most 3, see [8]. Furthermore, the pathwidth of 3-regular graphs is at most $\frac{1}{6} + o(n)$, and graphs with average degree d have treewidth at most $\frac{13dn}{150} + o(n)$, see [13, Chapter 5]. All of these results rely on the symmetry of the arc relation at some point in their proofs. We shall now see how to derive similar bounds for unsymmetric digraphs.

Recall that the well-known Caro-Wei inequality [7, 26] can be used to bound the order of a large independent set in a graph in terms of its average degree. Notice that an independent set induces a 0-degenerate subgraph, and *vice versa*. Consequently, that inequality was generalized to give a bound on the order of a large d-degenerate induced subgraph in a given graph in [1]. A different generalization of the Caro-Wei inequality, to the case of digraphs, was given in [15], yielding a lower bound on the order of an acyclic set in a given digraph. Observe that an acyclic set induces a 0-outdegenerate subdigraph, and *vice versa*. Therefore, it is quite natural to ask for a joint generalization of the results from both [1] and [15]. This is given by our next theorem. The proof is similar to the proof of Turán's Theorem using the probabilistic method given in [2, Chapter 7].

Theorem 11. *Let $D = (V, A)$ be a digraph. Then D contains a k-outdegenerate induced subdigraph of order at least $\sum_{v \in V} \min(1, \frac{k+1}{d^+(v)+1})$.* □

Proof. We utilize a randomized greedy algorithm, similar to the proof of Turán's Theorem using the probabilistic method given in [2, Chapter 7]. The algorithm is as follows: we choose an ordering $<$ on the vertex set V uniformly at random, and we use a working set $U \subseteq V$, which is initially empty. Then we visit each vertex v in the order given by $<$. For each v, we add v to U if and only if at most k out-neighbors of v are already in U. This completes the description of the algorithm.

We claim that the subdigraph induced by U is k-outdegenerate. To prove this, we need to show that for each subset $W \subseteq U$, the digraph induced by W contains a vertex of outdegree at most k. For $W \subseteq U$, let w be the last vertex in W w.r.t. the ordering $<$. Then for each out-neighbor x of w in the induced subdigraph $D[W]$, we have both $x < w$ and $x \in U$. Thus, by construction of the set U, vertex w can have at most k out-neighbors in $D[W]$. Thus, the induced subdigraph $D[U]$ is k-outdegenerate, as desired.

It remains to derive the claimed bound on the cardinality of U. We start with a general observation about uniformly random orderings of a finite set V. For a subset $S \subseteq V$ and an element $v \in S$, let $S_{<v}$ denote the set of elements in S that are smaller than v w.r.t. the ordering $<$. For $0 \leq i < |S|$, we have $Pr[\,|S_{<v}| = i\,] = \frac{1}{|S|}$, since every position for v is equally likely. Similarly, we have

$$Pr[\,|S_{<v}| \leq i\,] = \min\left(1, \frac{i+1}{|S|}\right),$$

for $i \geq 0$.

Now we return to the randomized greedy algorithm. For a vertex v, let $N^+(v)$ denote the set of out-neighbors of v. Observe that $|N^+(v) \cup \{v\}| \leq d^+(v) + 1$

(and equality holds in case D is a loop-free digraph). The vertex v is added to the working set U if and only if at most k out-neighbors are smaller than v w.r.t. the ordering $<$. Using the notation introduced above, vertex v is added to U if and only if $|(N^+(v) \cup \{v\})_{<v}| \leq k$. Thus,

$$\begin{aligned} Pr[v \in U] &= Pr[\,|(N^+(v) \cup \{v\})_{<v}| \leq k\,] \\ &= \min\left(1, \frac{k+1}{|(N^+(v) \cup \{v\}|}\right) \\ &\geq \min\left(1, \frac{k+1}{d^+(v)+1}\right). \end{aligned}$$

For a vertex v, let I_v denote the indicator variable for the event "$v \in U$." Then $E[\,I_v\,] = Pr[\,v \in U\,]$, and $|U| = \sum_{v \in V} I_v$. By linearity of expectation, we have

$$E[\,|U|\,] = \sum_{v \in V} Pr[\,v \in U\,] \geq \sum_{v \in V} \min\left(1, \frac{k+1}{d^+(v)+1}\right).$$

This bound on the expected size of U clearly implies the existence of an induced k-outdegenerate subdigraph of that order, and the proof is completed. □

We have the following corollary in terms of the number of arcs in D, or, equivalently, in terms of the average outdegree.

Corollary 12. *Let $D = (V, A)$ be a digraph of order n with average outdegree $d \geq 2k$. Then D has an induced k-outdegenerate subdigraph of order at least $\frac{k+1}{d+1} \cdot n$.* □

Proof. For the special case of symmetric digraphs, a proof is sketched in [1, pp. 208f]. The argument carries over with obvious modifications, but entails some careful calculations. For completeness, we include a streamlined proof.

We start with a bit of integer mathematics. Let k be a nonnegative integer, let n and a be positive integers, and let $d_1, d_2, \ldots, d_n$ be real numbers. Let w denote the minimum possible value of the expression

$$\sum_{i=1}^{n} \min\left(1, \frac{k+1}{d_i+1}\right), \tag{4.1}$$

where the minimum is taken subject to the constraint

$$\sum_{i=1}^{n} d_i = a, \text{ and all } d_i \text{ are nonnegative integers.} \tag{4.2}$$

For the following two claims, let $b_1, b_2, \ldots, b_n$ be numbers such that the assignment $d_i = b_i$ for $1 \leq i \leq n$ satisfies Condition (4.2).

Claim. If $a \geq 2kn$ and the assignment $d_i = b_i$ for $1 \leq i \leq n$ attains the minimum of Equation (4.1) subject to Condition (4.2), then each of the b_is is either zero or greater than k.

Proof. The claim trivially holds for $n = 1$, so we assume $n \geq 2$. For the sake of contradiction, assume $d_i = b_i$, for $1 \leq i \leq n$, is an assignment that attains the minimum, and $0 < b_j \leq k$, for some j. Pick an index j' such that the value $b_{j'}$ is maximal among the b_is. Then $b_{j'} > k + 1$, since $a \geq 2kn$. Then define $c_j = 0$, $c_{j'} = b_{j'} + b_j$, and $c_i = b_i$, for $i \neq j, j'$. The assignment $d_i = c_i$ again fulfills Condition (4.2). But then $\min\left(1, \frac{k+1}{b_i+1}\right) = \min\left(1, \frac{k+1}{c_i+1}\right)$, for all $i \neq j'$, and $\min\left(1, \frac{k+1}{b_{j'}+1}\right) > \min\left(1, \frac{k+1}{c_{j'}+1}\right)$, contradicting the assumed minimality of the original assignment. This establishes the first claim. □

If a is large enough, we can also get rid of the zeros in the assignment:

Claim. Assume that for $1 \leq i \leq n$, each b_i is either zero or greater than k. If $a \geq m \cdot 2k$ for some $m \leq n$, and the number of positive b_is is ℓ, where $\ell < m$, then the sum in Equation (4.1) is *not increased* by changing one of the zeros to $2k$ and by decreasing the positive b_is by a total of $2k$ in such a way that each of them is still at least $2k$.

Proof. On the one hand, decreasing one of the positive b_is by 1 leads to increasing the sum by $\frac{k+1}{b_i(b_i+1)}$, and this is at most $\frac{k+1}{(2k+1)(2k+2)}$, since we have assumed that after decreasing b_i, the value $b_i - 1$ is still at least $2k$. By performing this step for $2k$ times, the sum is increased by an amount of at most $\frac{2k(k+1)}{(2k+1)(2k+2)}$. Observe, that we can find a sufficiently large b_i in each step, since the arithmetic mean of the positive b_is will be larger than $2k$. On the other hand, if we assign the value $2k$ instead of 0 to one of the d_is, the term 1 is replaced with $\frac{k+1}{2k+1}$. This decreases the sum by exactly $-1 + \frac{k+1}{2k+1} = -\frac{k}{2k+1}$. All the steps taken together do not increase the value of the sum, since we have $\frac{2k(k+1)}{(2k+1)(2k+2)} - \frac{k}{2k+1} = 0$. This establishes the second claim. □

The above two claims together imply that for $a \geq 2kn$, minimizing the sum in Equation (4.1) subject to Condition (4.2) is equivalent to minimizing the sum

$$\sum_{i=1}^{n} \frac{k+1}{d_i+1} \tag{4.3}$$

subject to Condition (4.2).

Now Theorem 11 implies that $D = (V, A)$ has an induced k-outdegenerate subdigraph of order at least w—recall, that w refers to the minimum possible value of the expression shown in Equation (4.1) subject to Condition (4.2). Since for the number of arcs holds $|A| = a \geq 2kn$, the corollary follows if we apply to Equation (4.3) the inequality relating arithmetic and harmonic mean. □

Now we are ready to prove Theorem 2, which was stated in the introduction. Namely, we can use the special case $k = 1$ to find in a digraph D with average outdegree $d \geq 2$ an induced subdigraph of order at least $\frac{2}{d+1} \cdot n$, which has Kelly-width 2, and consequently, cycle rank in $O(\log n)$.

Proof (of Theorem 2). By Corollary 12, the digraph D has a vertex subset U of cardinality $\frac{2}{d+1} \cdot n$, such that $D[U]$ is 1-outdegenerate. According to Lemma 10,

the digraph $D[U]$ has DAG-width at most 1 and Kelly-width at most 2. We claim that the digraph $D[U]$ has cycle rank at most $O(\log n)$.

Namely, digraphs of small Kelly-width always have some small vertex subset whose removal leaves a digraph with much smaller strongly connected subsets: since $D[U]$ is a digraph of Kelly-width at most 2, there is a vertex subset X of size at most 11, such that each strongly connected component C_i of $D[U] - X$ is of order at most 2/3 times the order of $D[U]$. This follows from a similar property of digraphs of small directed treewidth [20], together with the known relation between directed treewidth and Kelly-width [19]. By the definition of cycle rank, we have

$$r(D[U]) \leq r(D[U] - X) + |X| = \max_i r(D[C_i]) + 11.$$

Observe that the subdigraphs C_i again have Kelly-width at most 2. So we can apply the above reasoning recursively, until after $O(\log n)$ times the maximum order among the strongly connected components thus obtained equals 1. □

Also, observe that, since the Kelly-width of the subdigraph in the above proof is 2, its D-width, and directed treewidth is in $O(1)$, compare [5, 19]. Bounds for several directed width measures on sparse digraphs now follow immediately:

Theorem 13. *Let D be a digraph of order n with average outdegree $d \geq 2$. Then the cycle rank (directed pathwidth, respectively) of D is at most*

$$\frac{d-1}{d+1} \cdot n + O(\log n),$$

and the Kelly-width (D-width, DAG-width, directed treewidth, respectively) is at most

$$\frac{d-1}{d+1} \cdot n + O(1).$$

Proof. Let U be a vertex subset as implied by Theorem 2. Using the definition of cycle rank, we obtain

$$r(D) \leq |U| + r(D - U) \leq \frac{d-1}{d+1} \cdot n + O(\log n).$$

The same bound holds for directed pathwidth, since directed pathwidth is bounded above by cycle rank [15]. The bound for Kelly-width, D-width, DAG-width and directed treewidth follows along the same lines. □

5 Upper Bound on Converting Finite Automata to Regular Expressions

Now we are ready to finish the proof of Theorem 3, the main result of this paper. The conversion strategy follows along similar lines as developed in [16]. There it was shown how one can split up the state set of the automaton into an "easy" and a "hard" part. In our case the expression size for converting the easy part is governed by the bound in Theorem 1. Converting the hard part by

state elimination yields an exponential blow-up in the size of this part; but by Theorem 2, we can ensure that the hard part is sufficiently small. Recall, that we want derive an upper bound of $k \cdot 4^{\frac{k-1}{k+1}n} \cdot n^{O(\log n)^2}$ on the classic problem of converting deterministic finite automata over k-ary alphabet into equivalent regular expressions.

Proof (of Theorem 3). Let $A = (Q, \Sigma, \delta, q_0, F)$ be an n-state deterministic finite automaton accepting the language L, and let D be its underlying digraph. For each pair of states (s, t), we derive a regular expression describing the language $L_{st}(D)$. Since A is deterministic, all vertices in D have outdegree at most k. Therefore, by Theorem 2, the digraph D has an induced subdigraph $D[U]$ of cycle rank $O(\log n)$, and of order at least $\frac{2}{d+1} \cdot n$. Now we apply Lemma 8 to $D[U]$: for each pair of states (s, t) in $Q \times Q$, there is a regular expression describing $L_{st}(D[U])$ of size at most $n^{O(\log n)^2}$. From these intermediate expressions, we obtain regular expressions $L_{st}(D)$ by repeated application of the McNaughton-Yamada-recurrence [22]. More precisely, for $W \subseteq V$ and $v \in Q - W$, we have

$$L_{st}(D[W \cup \{v\}]) = L_{st}(D[W]) \cup L_{sv}(D[W]) \cdot L_{vv}(D[W])^* \cdot L_{vt}(D[W]).$$

Starting with $W = U$, we apply the recurrence $|Q - U| = \frac{k-1}{k+1} \cdot n$ times, and each such application blows up the size of the intermediate expressions by a factor of at most 4. Altogether, this yields regular expressions describing the sets $L_{st}(D)$, each of size at most $4^{\frac{k-1}{k+1}n} \cdot n^{O(\log n)^2}$. The proof is completed by observing that a regular expression for $L(A)$ is obtained by a morphism that maps the arcs of the digraph to suitable elements in Σ. Hence we obtain

$$\begin{aligned} \operatorname{alph}(L(A)) &\leq |\Sigma| \cdot \sum_{f \in F} \operatorname{alph}(L_{q_0 f}(D)) \\ &\leq |\Sigma| \cdot 4^{\frac{k-1}{k+1}n} \cdot n^{O(\log n)^2}, \end{aligned}$$

and thus, the stated claim follows. □

In the case of binary alphabets, this gives a substantial improvement over the previously known bound of $O(1.682^n)$ of [8].

Theorem 14. *Let L be a regular language over a binary alphabet. If L is accepted by an n-state deterministic finite automaton, then*

$$\operatorname{alph}(L) \leq 4^{\frac{1}{3}n} \cdot n^{O(\log n)^2} \leq O\left(1.588^n\right). \qquad \square$$

For alphabet sizes three, four, and five, the results from [16] give regular expressions of size $O(2.209^n)$, $O(2.520^n)$, and $O(2.741^n)$, respectively. These bounds are correspondingly improved to $2^n \cdot n^{O(\log n)^2}$, $O(2.298^n)$, and $O(2.520^n)$, respectively.

References

1. N. Alon, J. Kahn, and P. D. Seymour. Large induced degenerate subgraphs. *Graphs and Combinatorics*, 3(1):203–211, 1987.
2. N. Alon and J. H. Spencer. *The Probabilistic Method*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons, 2008.
3. J. Bang-Jensen and G. Gutin. *Digraphs: Theory, Algorithms and Applications*. Springer Monographs in Mathematics. Springer, 2000.
4. V. Batagelj and M. Zaversnik. An $O(m)$ algorithm for cores decomposition of networks. arXiv:cs/0310049v1 [cs.DS], 2013.
5. D. Berwanger, A Dawar, P. W. Hunter, St. Kreutzer, and J. Obdrzálek. DAG-width and parity games. *Journal of Combinatorial Theory, Series B*, 102(4):900–923, 2012.
6. H. L Bodlaender, Th. Wolle, and A. M. C. A. Koster. Contraction and treewidth lower bounds. *Journal of Graph Algorithms and Applications*, 10(1):5–49, 2006.
7. Y. Caro. New results on the independence number. Technical report, Tel Aviv University, 1979.
8. K. Edwards and G. Farr. Improved upper bounds for planarization and series-parallelization of degree-bounded graphs. *The Electronic Journal of Combinatorics*, 19(2):#P25, 2012.
9. A. Ehrenfeucht and H. P. Zeiger. Complexity measures for regular expressions. *J. Comput. System Sci.*, 12(2):134–146, 1976.
10. K. Ellul, B. Krawetz, J. Shallit, and M.-W. Wang. Regular expressions: New results and open problems. *J. Autom., Lang. Comb.*, 9(2/3):233–256, 2004.
11. P. Erdős and A. Hajnal. On chromatic number of graphs and set-systems. *Acta Mathematica Academiae Scientiarum Hungaricae*, 17(1–2):61–99, 1966.
12. Ph. Flajolet and R. Sedgewick. *Analytic Combinatorics*. Cambridge University Press, 2009.
13. F. V. Fomin and D. Kratsch. *Exact Exponential Algorithms*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2010.
14. W. Gelade and F. Neven. Succinctness of complement and intersection of regular expressions. In S. Albers and P. Weil, editors, *Proceedings of the* 25*th International Symposium on Theoretical Aspects of Compter Science*, volume 1 of *LIPIcs*, pages 325–336, Bordeaux, France, 2008. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany.
15. H. Gruber. Bounding the feedback vertex number of digraphs in terms of vertex degrees. *Discrete Appl. Math.*, 159(8):872–875, 2011.
16. H. Gruber and M. Holzer. Provably shorter regular expressions from finite automata. *Internat. J. Found. Comput. Sci.*, 24(8):1255–1279, 2013.
17. H. Gruber and J. Johannsen. Optimal lower bounds on regular expression size using communication complexity. In R. Amadio, editor, *Proceedings of the* 11*th Conference Foundations of Software Science and Computation Structures*, number 4962 in LNCS, pages 273–286, Budapest, Hungary, 2008. Springer.
18. J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, 1979.
19. P. Hunter and S. Kreutzer. Digraph measures: Kelly decompositions, games, and orderings. *Theoret. Comput. Sci.*, 399(3):206–219, 2008.
20. T. Johnson, N. Robertson, P. D. Seymour, and R. Thomas. Directed tree-width. *Journal of Combinatorial Theory, Series B*, 82(1):138–154, 2001.
21. D. R. Lick and A. T. White. k-degenerate graphs. *Canadian Journal of Mathematics*, 22:1082–1096, 1970.
22. Robert McNaughton and Hisao Yamada. Regular expressions and state graphs for automata. *IRE Transactions on Electronic Computers*, EC-9(1):39–47, 1960.
23. D. Meister, J. A. Telle, and M. Vatshelle. Recongnizing digraphs of Kelly-width 2. *Discrete Appl. Math.*, 158(7):741–746, 2010.
24. J. Sakarovitch. The language, the expression, and the (small) automaton. In J. Ferré, I. Litovsky, and S. Schmitz, editors, *Proceedings of the* 10*th Conference on Implementation and Application of Automata*, number 3845 in LNCS, pages 15–30, Sophia Antipolis, France, 2005. Springer.

25. B. W. Watson. *Taxonomies and Toolkits of Regular Language Algorithms.* PhD thesis, Eindhoven University of Technology, Department of Mathematics and Computer Science, Den Dolech 2, 5612 AZ Eindhoven, The Netherlands, 1995.
26. V. K. Wei. A lower bound on the stability number of a simple graph. Technical memorandum no. 81-11217-9, Bell Laboratories, 1981.
27. D. Wood. *Theory of Computation.* John Wiley & Sons, 1987.

Institut für Informatik

Justus-Liebig-Universität Giessen
Arndtstr. 2, 35392 Giessen, Germany

Recent Reports

(Further reports are available at `www.informatik.uni-giessen.de`.)

M. Kutrib, A. Malcher, M. Wendlandt, *Deterministic Set Automata*, Report 1402, April 2014.

M. Holzer, S. Jakobi, *Minimal and Hyper-Minimal Biautomata*, Report 1401, March 2014.

J. Kari, M. Kutrib, A. Malcher (Eds.), *19th International Workshop on Cellular Automata and Discrete Complex Systems AUTOMATA 2013 Exploratory Papers*, Report 1302, September 2013.

M. Holzer, S. Jakobi, *Minimization, Characterizations, and Nondeterminism for Biautomata*, Report 1301, April 2013.

A. Malcher, K. Meckel, C. Mereghetti, B. Palano, *Descriptional Complexity of Pushdown Store Languages*, Report 1203, May 2012.

M. Holzer, S. Jakobi, *On the Complexity of Rolling Block and Alice Mazes*, Report 1202, March 2012.

M. Holzer, S. Jakobi, *Grid Graphs with Diagonal Edges and the Complexity of Xmas Mazes*, Report 1201, January 2012.

H. Gruber, S. Gulan, *Simplifying Regular Expressions: A Quantitative Perspective*, Report 0904, August 2009.

M. Kutrib, A. Malcher, *Cellular Automata with Sparse Communication*, Report 0903, May 2009.

M. Holzer, A. Maletti, *An $n \log n$ Algorithm for Hyper-Minimizing States in a (Minimized) Deterministic Automaton*, Report 0902, April 2009.

H. Gruber, M. Holzer, *Tight Bounds on the Descriptional Complexity of Regular Expressions*, Report 0901, February 2009.

M. Holzer, M. Kutrib, and A. Malcher (Eds.), *18. Theorietag Automaten und Formale Sprachen*, Report 0801, September 2008.

M. Holzer, M. Kutrib, *Flip-Pushdown Automata: Nondeterminism is Better than Determinism*, Report 0301, February 2003

M. Holzer, M. Kutrib, *Flip-Pushdown Automata: $k+1$ Pushdown Reversals are Better Than k*, Report 0206, November 2002

M. Holzer, M. Kutrib, *Nondeterministic Descriptional Complexity of Regular Languages*, Report 0205, September 2002

H. Bordihn, M. Holzer, M. Kutrib, *Economy of Description for Basic Constructions on Rational Transductions*, Report 0204, July 2002

M. Kutrib, J.-T. Löwe, *String Transformation for n-dimensional Image Compression*, Report 0203, May 2002

A. Klein, M. Kutrib, *Grammars with Scattered Nonterminals*, Report 0202, February 2002

A. Klein, M. Kutrib, *Self-Assembling Finite Automata*, Report 0201, January 2002

M. Holzer, M. Kutrib, *Unary Language Operations and its Nondeterministic State Complexity*, Report 0107, November 2001

A. Klein, M. Kutrib, *Fast One-Way Cellular Automata*, Report 0106, September 2001

M. Holzer, M. Kutrib, *Improving Raster Image Run-Length Encoding Using Data Order*, Report 0105, July 2001

M. Kutrib, *Refining Nondeterminism Below Linear-Time*, Report 0104, June 2001

M. Holzer, M. Kutrib, *State Complexity of Basic Operations on Nondeterministic Finite Automata*, Report 0103, April 2001

M. Kutrib, J.-T. Löwe, *Massively Parallel Fault Tolerant Computations on Syntactical Patterns*, Report 0102, March 2001