TIGHT BOUNDS ON THE DESCRIPTIONAL COMPLEXITY OF REGULAR EXPRESSIONS

Hermann Gruber   Markus Holzer

IFIG RESEARCH REPORT 0901
FEBRUARY 2009

Institut für Informatik
JLU Gießen
Arndtstraße 2
D-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 0901, February 2009

Tight Bounds on the Descriptional Complexity of Regular Expressions

Hermann Gruber[1] and Markus Holzer[2]

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

Abstract. We improve on some recent results on lower bounds for conversion problems for regular expressions. In particular we consider the conversion of planar deterministic finite automata to regular expressions, study the effect of the complementation operation on the descriptional complexity of regular expressions, and the conversion of regular expressions extended by adding intersection or interleaving to ordinary regular expressions. Almost all obtained lower bounds are optimal, and the presented examples are over a binary alphabet, which is best possible.

Categories and Subject Descriptors: F.1.1 [**Computation by Abstract Devices**]: Models of Computation—*Relations between models*; F.1.2 [**Computation by Abstract Devices**]: Modes of Computation—*Parallelism and concurrency*; F.3.1 [**Logics and Meanings of Programs**]: Specifying and Verifying and Reasoning about Programs—*Specification techniques*; F.4.3 [**Mathematical Logic and Formal Languages**]: Formal Languages—*Operations on languages*

Additional Key Words and Phrases: Interleaving, shuffle, semi-extended regular expressions, regular expressions

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

Conversion	known results	this paper with $\lvert\Sigma\rvert = 2$
planar DFA to RE	$2^{\Theta(\sqrt{n})}$ for $\lvert\Sigma\rvert = 4$ [10]	$2^{\Theta(\sqrt{n})}$ [Thm. 4]
$\neg$ RE to RE	$2^{2^{\Omega(\sqrt{n \log n})}}$ for $\lvert\Sigma\rvert = 2$ [10] $2^{2^{\Omega(n)}}$ for $\lvert\Sigma\rvert = 4$ [8]	$2^{2^{\Theta(n)}}$ [Thm. 8]
RE($\cap$) to RE	$2^{2^{\Omega(\sqrt{n})}}$ for $\lvert\Sigma\rvert = 2$ [7]	$2^{2^{\Theta(n)}}$ [Thm. 9]
RE($\sqcup\!\sqcup$) to RE	$2^{2^{\Omega(\sqrt{n})}}$ for $\lvert\Sigma\rvert$ const. [7]	$2^{2^{\Omega(n/\log n)}}$ [Thm. 16] $2^{2^{\Theta(n)}}$ for $\lvert\Sigma\rvert = O(n)$ [Thm. 10]

Table 1. Comparing the lower bound results for conversion problems of deterministic finite automata (DFA), regular expressions (RE), and regular expressions with additional operations (RE($\cdot$)), where $\cap$ denotes intersection, $\neg$ complementation, and $\sqcup\!\sqcup$ the interleaving or shuffle operation on formal languages. Entries with a bound in $\Theta(\cdot)$ indicate that the result is best possible, i.e., refers to a lower bound matching a known upper bound.

1 Introduction

It is well known that regular expressions are equally expressive as finite automata. In contrast to this equivalence, a by now classical result due to Ehrenfeucht and Zeiger states that finite automata, even deterministic ones, can sometimes allow exponentially more succinct representations than regular expressions [4]. Although they obtained a tight lower bound on expression size, their examples used a largely growing alphabet.

Reducing the alphabet size remained an open challenge [5] until the recent advent of new proof techniques, see [8, 10, 13]—most of our proofs in this paper rely on the recently established relation between regular expression size and star height of regular languages [10]. Although this resulted in quite a few new insights into the nature of regular expressions,see also [7, 11, 12], proving tight lower bounds for small alphabets remains a challenging task, and not all bounds in the mentioned references are both tight and cover all alphabet sizes. In this work, we close some of the remaining gaps: In the case of converting planar finite automata to regular expressions, we reduce the alphabet size to binary while retaining the tight lower bound. We prove this directly, by finding a witness language over a binary alphabet. For the other questions under consideration, namely the effect of complementation and of extending regular expression syntax by adding an intersection or interleaving operator, proceeding in this way appears more difficult. Yet, sometimes it proves easier to find witness languages over larger alphabets. For this case, we also devise a new set of encodings which are economic and, in some precise sense, robust with respect to both the Kleene star and the interleaving operation. This extends the outreach of known proof techniques, and allows us to give a definitive answer to some questions regarding the descriptional complexity of regular expressions that were not yet settled completely in previous works [5, 7, 8, 10]. Our main results are summarized and compared to known results in Table 1.

2 Basic Definitions

We introduce some basic notions in formal language and automata theory—for a thorough treatment, the reader might want to consult a textbook such as [16]. In particular, let Σ be a finite alphabet and Σ^* the set of all words over the alphabet Σ, including the empty word ϵ. The length of a word w is denoted by $|w|$, where $|\epsilon| = 0$. A *(formal) language* over the alphabet Σ is a subset of Σ^*.

The *regular expressions* over an alphabet Σ are defined recursively in the usual way:[1] $\emptyset$, ϵ, and every letter a with $a \in \Sigma$ is a regular expression; and when r_1 and r_2 are regular expressions, then $(r_1 + r_2)$, $(r_1 \cdot r_2)$, and $(r_1)^*$ are also regular expressions. The language defined by a regular expression r, denoted by $L(r)$, is defined as follows: $L(\emptyset) = \emptyset$, $L(\epsilon) = \{\epsilon\}$, $L(a) = \{a\}$, $L(r_1 + r_2) = L(r_1) \cup L(r_2)$, $L(r_1 \cdot r_2) = L(r_1) \cdot L(r_2)$, and $L(r_1^*) = L(r_1)^*$. The *size* or *alphabetic width* of a regular expression r over the alphabet Σ, denoted by $\mathrm{alph}(r)$, is defined as the total number of occurrences of letters of Σ in r. For a regular language L, we define its alphabetic width, $\mathrm{alph}(L)$, as the minimum alphabetic width among all regular expressions describing L.

Our arguments on lower bounds for the alphabetic width of regular languages is based on a recent result that utilizes the star height of regular languages [10]. Here the star height of a regular language is defined as follows: For a regular expression r over Σ, the star height, denoted by $h(r)$, is a structural complexity measure inductively defined by: $h(\emptyset) = h(\epsilon) = h(a) = 0$, $h(r_1 \cdot r_2) = h(r_1 + r_2) = \max(h(r_1), h(r_2))$, and $h(r_1^*) = 1 + h(r_1)$. The star height of a regular language L, denoted by $h(L)$, is then defined as the minimum star height among all regular expressions describing L. The next theorem establishes the aforementioned relation between alphabetic width and star height of regular languages [10]:

Theorem 1. *Let $L \subseteq \Sigma^*$ be a regular language. Then* $\mathrm{alph}(L) \geq 2^{\frac{1}{3}(h(L)-1)} - 1$.

The star height of a regular language appears to be more difficult to determine than its alphabetic width, see, e.g., [14]. Fortunately, the star height can be determined more easily for a certain subclass of regular languages, namely the family of bideterministic regular languages, which are defined as follows: A regular language L is *bideterministic* if there exists a deterministic finite automaton A with a single final state such that a deterministic finite automaton accepting the reversed language L^R is obtained from A by reverting the direction of each transition and exchanging the roles of the initial and final state. For these languages, the star height can be determined from the digraph structure of the minimal DFA: The *cycle rank* of a digraph $G = (V, E)$, denoted by $cr(G)$, is inductively defined as follows: (1) If G is acyclic, then $cr(G) = 0$. (2) If G is strongly connected, then $cr(G) = 1 + \min_{v \in V}\{cr(G - v)\}$, where $G - v$ denotes the graph with the vertex set $V \setminus \{v\}$ and appropriately defined edge set. (3) If G

[1] For convenience, parentheses in regular expressions are sometimes omitted and the concatenation is simply written as juxtaposition. The priority of operators is specified in the usual fashion: concatenation is performed before union, and star before both product and union.

is not strongly connected, then $cr(G)$ equals the maximum cycle rank among all strongly connected components of G.

For a given finite automaton A, let its cycle rank, denoted by $cr(A)$, be defined as the cycle rank of the underlying digraph. Eggan's Theorem states that the star height of a regular language equals the minimum cycle rank among all NFAs accepting it [3]. Later, the following was proved by McNaughton in [19], building on his earlier work [20]:

Theorem 2 (McNaughton's Theorem). *Let L be a bideterministic language, and let A be the minimal trim, i.e., without a dead state, deterministic finite automaton accepting L. Then $h(L) = cr(A)$.*

In fact, the minimality requirement in the above theorem is not needed, since every bideterministic finite automaton in which all states are useful is already a trim minimal deterministic finite automaton. Here, a state is useful if it is both reachable from the start state, and if some final state is reachable from it.

3 Lower Bounds on Regular Expression Size

This section is three folded. First we show an optimal bound converting planar deterministic finite automata to equivalent regular expressions and then we present our results on the alphabetic width on complementing regular expression and on regular expressions with intersection and interleaving. While the former result utilizes a characterization of cycle rank in terms of a cops and robbers game given in [10], the latter two results are mainly based on star height preserving homomorphisms.

3.1 Converting Planar DFAs into Regular Expressions

For the main result of this subsection we need a characterization of cycle rank in terms of a cops and robber game [20, 10]. This characterization provides a useful tool in proving lower bounds on the cycle rank of specific families of digraphs. The *cops and strong visible robber game*, defined in [17], is given as follows: Let $G = (V, E)$ 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 both a (possibly empty) directed path from v to v', and a (possibly empty) directed path back from v' to v in $G - (X \cap X')$, i.e., he has to avoid to run into a stationary cop, and to run along a path inside the current strongly connected component of the graph induced by the vertices free of stationary cops. 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.

The *immutable cops* variant of the above game restricts the movements of the cops in the following way: Once a cop has been placed on some vertex of the graph, he has to stay there forever. The *hot-plate* variant of the game restricts the movements of the robber in that he has to move along a nontrivial path in each move—even if the path consists only of a self-loop. The following theorem from [10] gives a characterization of the cycle rank in terms of such a game.

Theorem 3. *Let G be a digraph and $k \geq 0$. Then k cops have a winning strategy for the immutable cops and hot-plate strong visible robber game if and only if the cycle rank of G is at most k.*

Now we are ready for the main result of this subsection. Note that in [5] it was shown that for planar finite automata, one can construct equivalent regular expressions of size at most $2^{O(\sqrt{n})}$, for all alphabet sizes polynomial in n. This is a notable improvement over the general case, since conversion from n-state deterministic finite automata to equivalent regular expressions was shown to be of order $2^{\Theta(n)}$ in [10]. Also in [10] a tight lower bound on the conversion of planar deterministic finite automata to regular expressions of $2^{\Theta(\sqrt{n})}$ for alphabet size at least four was proven. Next we improve this result to alphabets of size two, using the above given characterization of cycle rank in terms of a cops and robber game.

Theorem 4. *There is an infinite family of languages L_n over a binary alphabet acceptable by n-state planar deterministic finite automata, such that* $\mathrm{alph}(L_n) = 2^{\Omega(\sqrt{n})}$.

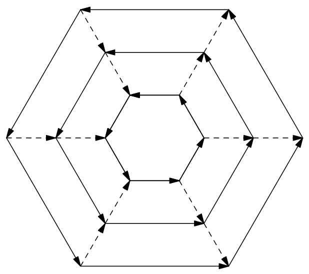

Fig. 1. A drawing of the graph G_3. When viewed as automaton A_3, the solid (dashed, respectively) arrows indicate a-transitions (b-transitions, respectively).

Proof. By Theorems 1 and 2, it suffices to find an infinite family of bideterministic finite automata A_k of size $O(k^2)$ such that the digraph underlying A_k has cycle rank $\Omega(k)$.

The deterministic finite automata A_k witnessing the claimed lower bound are inspired by a family of digraphs G_k defined in [17]. These graphs each admit a planar drawing as the union of k concentric equally directed $2k$-cycles, which are connected to each other by $2k$ radial directed k-paths, the first k of which are directed inwards, while the remaining k are directed outwards; see Figure 1 for illustration. Formally, for $k \geq 1$, let $G_k = (V, E)$ be the graph with vertex set $V = \{ u_{i,j} \mid 1 \leq i, j \leq k \} \cup \{ v_{i,j} \mid 1 \leq i, j \leq k \}$, and whose edge set can be partitioned into a set of directed $2k$-cycles C_i, and two sets of directed k-paths P_i and Q_i with $1 \leq i \leq k$. Here each C_i admits a walk visiting the vertices $u_{i,1}, u_{i,2}, \ldots, u_{i,k}, v_{i,1}, v_{i,2}, \ldots, v_{i,k}$ in order, each P_i admits a walk visiting the vertices $u_{1,i}, u_{2,i}, \ldots, u_{k,i}$ in order, and Q_i admits a walk visiting the vertices $v_{k,i}, v_{k-1,i}, \ldots, v_{k,1}$ in order.

Fix $\{a, b\}$ as a binary input alphabet. If we interpret the edges in G_k belonging to the cycles C_i as a-transitions, the edges belonging to the paths P_i and Q_i as b-transitions, interpret the vertices as states and choose a single initial and a single final state (both arbitrarily), we obtain a finite automaton A_k with $O(k^2)$ states whose underlying digraph is G_k. It is easily observed that A_k is bideterministic; thus it only remains to show that for the underlying graph G_k holds $cr(G_k) = \Omega(k)$.

To this end, we use the game characterization of cycle rank given by Theorem 3, by showing that the robber can escape against k cops in the immutable cops and hot-plate strong visible robber game. It was shown in [17] that on the graph G_k, the robber has a winning strategy against k cops in the cops and strong visible robber game, even if the cops are not immutable and can freely jump between vertices on the graph. The result is, however, not established for the hot-plate variant of the game, which restricts the allowed movements of the robber. But note that at most one additional cop is needed if we drop the hot-plate restriction: The allowed movements of the robber in hot-plate variant coincide with the original ones as long as he resides in a strongly connected component of size at least two. The situation becomes different only once the robber is finally trapped in a strongly connected component consisting only of one vertex (and no loop). When playing the game in the hot-plate variant, the robber is caught in this situation. Otherwise, we place the additional cop at this vertex to catch the robber. Note that since the extra cop never moves, this argument equally applies in the immutable cops variant of the game. □

3.2 Operations on Regular Expressions: Alphabetic Width of Complementation

As noted in [5], the naive approach to complement regular expressions, of converting first the given expression into an nondeterministic finite automaton, determinizing, complementing the resulting deterministic finite automaton, and converting back to a regular expression gives a doubly exponential upper bound of $2^{2^{O(n)}}$. The authors of [5] also gave a lower bound of $2^{\Omega(n)}$, and stated as an open problem to find tight bounds. A doubly-exponential lower bound was found in [8], but only for alphabets of size at least four. Their witness language is a 4-symbol encoding of the set of walks in an n-vertex complete digraph. They gave a very short regular expression describing the complement of the encoded set, and provided a direct and technical proof showing that the encoded language requires large regular expressions, carefully adapting the approach originally taken by Ehrenfeucht and Zeiger [4]. Resulting from an independent approach pursued by the authors, in [10] a roughly doubly-exponential lower bound of $2^{2^{O(\sqrt{n \log n})}}$ was given for binary alphabet.

Now it appears tempting to encode the language from [8] using a star height preserving homomorphism to further reduce the alphabet size, as done in [10] for a similar problem. Unfortunately, the proof from [8] does not offer any clue about the star height of the witness language, and thus we cannot mix these proof techniques. At least, it is known [2] that the *preimage* of the encoded language has large star height:

Theorem 5 (Cohen). *Let J_n be the complete digraph on n vertices with self-loops, where each edge (i, j) carries a unique label a_{ij}. Let W_n denote the set of all walks $a_{i_0 i_1} a_{i_1 i_2} \cdots a_{i_{r-2} i_{r-1}} a_{i_{r-1} i_r}$ in J_n, including the empty walk ϵ. Then the star height of language W_n equals n.*

To obtain a tight lower bound for binary alphabets, here we use a similar encoding as in [8], but make sure that the encoding is a star height preserving homomorphism. Here a homomorphism ρ *preserves star height*, if the star height of each regular language L equals the star height of the homomorphic image $\rho(L)$. The existence of such encodings was conjectured in [3] and proved in [20]:

Theorem 6. *Let $\Sigma = \{a_1, a_2, \ldots, a_d\}$ be a finite alphabet with $d \geq 1$ and define $\sigma : \Sigma^* \to \{a, b\}^*$ be the homomorphism given by $\sigma(a_i) = a^i b^{d-i+1}$, for $1 \leq i \leq d$. Then for every regular language L over Σ, the star height of L equals the star height of $\sigma(L)$.*

A full characterization of star height preserving homomorphisms was established later in [15], which reads as follows:

Theorem 7 (Hashiguchi/Honda). *A homomorphism $\rho : \Gamma^* \to \Sigma^*$ preserves star height if and only if (1) ρ is injective, (2) ρ is both prefix-free and suffix-free, that is, no word in $\rho(\Gamma)$ is prefix or suffix of another word in $\rho(\Gamma)$, and (3) ρ has the non-crossing property, that is, for all $x_1 y_1$ and $x_2 y_2$ that form two distinct words in $\rho(\Gamma)$, at least one of the cross-wise concatenations $x_1 x_2$ and $x_2 y_1$ do not belong to $\rho(\Gamma)$.*

Observe that the given lower bound matches the aforementioned upper bound on the problem under consideration.

Theorem 8. *There exists an infinite family of languages L_n over a binary alphabet Σ with* $\mathrm{alph}(L_n) = O(n)$*, such that* $\mathrm{alph}(\Sigma^* \setminus L_n) = 2^{2^{\Omega(n)}}$.

Proof. We will first prove the theorem for alphabet size 3, and then use a star-height preserving homomorphism to further reduce the alphabet size to binary. Let W_{2^n} be the set of walks in a complete 2^n-vertex digraph as defined in Theorem 5. Let $E = \{ a_{ij} \mid 0 \leq i, j \leq 2^n - 1 \}$ denote the edge set of this graph, and let $\Sigma = \{0, 1, \$\}$.

Now define the homomorphism $\rho : E^* \to \Sigma^*$ by $\rho(a_{ij}) = \mathrm{bin}(i) \cdot \mathrm{bin}(j) \cdot \mathrm{bin}(i) \cdot \mathrm{bin}(j)\$$, where $\mathrm{bin}(i)$ denotes the usual n-bit binary encoding of the number i. Observe that ρ is star height preserving. To his end one has to verify the properties of Theorem 7. It is easily seen that the encoding is injective, and since ρ maps every symbol to a word of length $4n+1$, it is both prefix-free and suffix-free. Finally, for the noncrossing property, observe that the set $\rho(E)$ coincides with the set of squares of length $4n$ over $\{0,1\}$ followed by a dollar sign, in symbols $\rho(E) = \{w^2\$ \mid w \in \{0,1\}^*, |w| = 2n\}$. Assume x_1y_1 and x_2y_2 are two distinct square words of length $4n$. If x_1 and y_1 have different lengths, then the length of $x_1y_2\$$ is not equal to $4n+1$. Otherwise, there is some position j where x_1y_1 and x_2y_2 have different letters. Then in the word x_1y_2, the letter at its j-th position differs from the letter at its $(4n-j)$-th position, and hence this word is not a square. Thus, ρ is a star height preserving homomorphism.

Our witness language for ternary alphabets is the complement of the set $L_n = \rho(W_{2^n})$. To establish the theorem for ternary alphabets, we give a regular expression of size $O(n)$ describing the complement of L_n; a lower bound of $2^{2^{\Omega(n)}}$ then immediately follows from Theorems 1 and 5 since the homomorphism ρ preserves star height. As for the witness language given in [8], our expression is a union of some local consistency tests: Every nonempty word in L_n falls apart into blocks of binary digits of each of length $4n$, separated by occurrences of the symbol \$, and takes the form

$$(\mathrm{bin}(i_0)\,\mathrm{bin}(i_1))^2\$\,(\mathrm{bin}(i_1)\,\mathrm{bin}(i_2))^2\,\$\cdots\$(\mathrm{bin}(i_{r-1})\,\mathrm{bin}(i_r))^2\$.$$

Thus, word w is not in L_n if and only if we have at least one of the following cases: (i) The word w has no prefix in $\{0,1\}^{4n}\$$, or w contains an occurrence of \$ not immediately followed by a word in $\{0,1\}^{4n}\$$; (ii) the region around the boundary of some pair of adjacent blocks in w is not of the form $\mathrm{bin}(i)\$\,\mathrm{bin}(i)$; or (iii) some block does not contain the pattern $(\mathrm{bin}(i)\,\mathrm{bin}(j))^2$, in the sense that inside the block some pair of bits at distance $2n$ does not match. To complete the proof for ternary alphabets, observe that at least one of the above three cases applies if and only if w matches the following regular expression of size $O(n)$:

$$\begin{aligned} r_n = {} & (0+1)^{\geq 1} + (\epsilon + \Sigma^*\$)\left((0+1)^{\leq 4n-1}\$ + (0+1)^{\geq 4n+1}\right)\Sigma^* \\ & + (\Sigma^*\$(0+1)^{3n}(0+1)^*0\Sigma^{n+1}1\Sigma^*) \\ & + (\Sigma^*\$(0+1)^{3n}(0+1)^*1\Sigma^{n+1}0\Sigma^*) \\ & + (\Sigma^*\$(0+1)^*0\Sigma^{2n}1\Sigma^*) \\ & + (\Sigma^*\$(0+1)^*1\Sigma^{2n}0\Sigma^*) \end{aligned}$$

To further decrease the alphabet size to binary, we use the star height preserving homomorphism σ given in Theorem 6, which already proved useful in [10]. Then $\sigma(L_n)$ has star height 2^n and thus again has alphabetic width at least $2^{2^{\Omega(n)}}$. For an upper bound on the alphabetic width of its complement, note first that every word w that is in $\sigma(\Sigma^*)$ but not in $\sigma(L_n)$ matches the morphic image under σ of the expression r_n given above; and $\sigma(r_n)$ still has alphabetic width $O(n)$. The words in the complement of $\sigma(L_n)$ not covered by the expression $\sigma(r_n)$ are precisely those not in $\sigma(\{0,1,\$\}^*)$, and the complement of the latter set can be described by a regular expression of constant size.

The union of these two expressions gives a regular expression of size $O(n)$ as desired. □

3.3 Regular Expressions with Intersection and Interleaving

It is known that extending the syntax with an intersection operator can provide an exponential gain in succinctness over nondeterministic finite automata. For instance, in [6] it is shown that the set of palindromes of length n can be described by regular expressions with intersection of size $O(n)$. On the other hand, it is well known that the number of states of a nondeterministic finite automaton accepting P_n has $\Omega(2^n)$ states [21]. Of course, it appears more natural to compare the gain in succinctness of such extended regular expressions to ordinary regular expressions rather than to finite automata. There a $2^{2^{O(n)}}$ doubly exponential upper bound readily follows by combining standard constructions [7]. Yet a roughly doubly-exponential lower bound of $2^{2^{\Omega(\sqrt{n})}}$, for alphabets of growing size, was found only recently in [8], and a follow-up paper [7] shows that this can be reached already for binary alphabets. Here we finally establish a tight doubly-exponential lower bound, which even holds for binary alphabets.

Theorem 9. *There is an infinite family of languages L_n over a binary alphabet admitting regular expressions with intersection of size $O(n)$, such that* $\mathrm{alph}(L_n) = 2^{2^{\Omega(n)}}$.

Proof. First, we show that the set of walks $W_{2^n} \subseteq E^*$ defined in Theorem 5 allows a compact representation using regular expressions with intersection. First we define $M = \{ a_{i,j} \cdot a_{j,k} \mid 0 \leq i,j,k \leq 2^n - 1 \}$ and the observe, that the set *Even* of all nonempty walks of even length, i.e., total number of seen edges, in the graph J_n can be written as $\mathit{Even} = M^* \cap (E \cdot M^* \cdot E)$, while the the set *Odd* of all nonempty walks of odd length is $\mathit{Odd} = (E \cdot M^*) \cap (M^* \cdot E)$. Thus, we have $W_{2^n} = \mathit{Even} \cup \mathit{Odd} \cup \{\epsilon\}$. This way of describing W_{2^n} appears to be a long shot from our goal; it uses a large alphabet and does not even reach a linear-exponential gain in succinctness over ordinary regular expressions—a similar statement appears, already over thirty years ago, in [4].

In order to get the desired result, we present a binary encoding τ that preserves star height and allows a representation of the encoded sets $\tau(M)$ and $\tau(E)$ by regular expressions with intersection each of size $O(n)$. Let $\tau : E^* \to \{0,1\}^*$ be the homomorphism defined by $\tau(a_{i,j}) = \mathrm{bin}(i) \cdot \mathrm{bin}(j) \cdot \mathrm{bin}(j)^R \cdot \mathrm{bin}(i)^R$, for $0 \leq i,j \leq 2^{n-1}$. To see that τ preserves star height, we have to check the properties given in Theorem 7. It can be readily seen that τ is injective, and it is both prefix-free and suffix-free, since all words in $\tau(E)$ are of the same length. The set $\tau(E)$ is just the set of binary palindromes of length $4n$, and, by chance, a proof of the non-crossing property of this set is given already in [9], albeit in a different context. Thus, by Theorems 1 and 5, the set $\tau(W_{2^n})$ has alphabetic width at least $2^{2^{\Omega(n)}}$.

It remains to give expressions with intersection of size $O(n)$ for the set $\tau(W_{2^n})$. Since $\tau(W_{2^n}) = \tau(\mathit{Even}) \cup \tau(\mathit{Odd}) \cup \{\epsilon\}$, the homomorphism commutes with concatenation, union, and Kleene star, and, being injective, also with intersection, it suffices to give regular expressions with intersection for $\tau(E)$ and $\tau(M)$

of size $O(n)$. To this end, we we make use of an observation from [6], namely that the sets of palindromes of length $2m$ admit regular expressions with intersection of size $O(m)$. A straightforward extension of that idea gives a short regular expression with intersection for the set

$$S_{m,n} = \{\, vwv^R \in \{0,1\}^* \mid |v| = m, |w| = n \,\},$$

where m and n are fixed nonnegative integers: Namely, an expression $r_{m,n}$ describing this set is defined inductively by letting

$$r_{0,n} = (0+1)^n$$

and

$$r_{m,n} = ((0+1) \cdot r_{m-1,n} \cdot (0+1)) \cap (0(0+1)^*0 + 1(0+1)^*1)\,,$$

for $m > 0$. Clearly, expression $r_{m,n}$ has size $O(m+n)$ and describes the language $S_{m,n}$. Next, observe that the set $\tau(E) = \{\, ww^R \in \{0,1\}^* \mid |w| = 2n \,\}$ is described by expression $r_{n,0}$, which is of size $O(n)$. Finally, note that the set $\tau(M)$ being equal to

$$\{\, \mathrm{bin}(i)\,\mathrm{bin}(j)\,\mathrm{bin}(j)^R\,\mathrm{bin}(i)^R\,\mathrm{bin}(j)\,\mathrm{bin}(k)\,\mathrm{bin}(k)^R\,\mathrm{bin}(j)^R \mid 0 \leq i,j,k \leq 2^n - 1 \,\},$$

can be written as $\tau(E)^2 \cap \{0,1\}^{2n} \cdot S_{n,n} \cdot \{0,1\}^{3n}$. The latter set can be described by a regular expression with intersection of size $O(n)$ again, and the proof is completed. □

The interleaving of languages is another natural language operation known to preserve regularity. Regular expressions extended with interleaving were first studied in [18], with focus on the computational complexity of word problems. They also showed that regular expressions extended with an interleaving operator can be exponentially more succinct than nondeterministic finite automata [18]. Very recently, it was shown in [7] that regular expressions with interleaving can be roughly doubly-exponentially more succinct than regular expressions: Converting such expressions into ordinary regular expressions can cause a blow-up in required expression size of $2^{2^{\Omega(\sqrt{n})}}$, for constant alphabet size. This bound is close to an easy upper bound of $2^{2^{O(n)}}$ that follows from standard constructions, see, e.g., [7] for details. If we take alphabets of growing size into account, the lower bound can be increased to match this trivial upper bound. The language witnessing that bound is in fact of very simple structure.

Theorem 10. *There is an infinite family of languages L_n over an alphabet of size $O(n)$ having regular expressions with interleaving of size $O(n)$, such that* $\mathrm{alph}(L_n) = 2^{2^{\Omega(n)}}$.

Proof. We consider the language L_n described by the shuffle regular expression

$$r_n = (a_1b_1)^* \mathbin{ш} (a_2b_2)^* \mathbin{ш} \cdots \mathbin{ш} (a_nb_n)^*$$

of size $O(n)$ over the alphabet $\Gamma = \{a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_n\}$.

To give a lower bound on the alphabetic width of L_n, we estimate first the star height of L_n. The language L_n can be accepted by a 2^n-state partial bideterministic finite automaton $A = (Q, \Sigma, \delta, q_0, F)$, whose underlying digraph forms a symmetric n-dimensional hypercube: The set of states is $Q = \{0,1\}^n$, the state $q_0 = 0^n$ is the initial state, and is also the only final state, i.e., $F = \{0^n\}$. For $1 \leq i \leq n$, the partial transition function δ is specified by $\delta(p, b_i) = q$ and $\delta(q, a_i) = p$, for all pairs of states (p, q) of the form $(x0y, x1y)$ with $x \in \{0,1\}^{i-1}$ and $y \in \{0,1\}^{n-i}$. It can be readily verified that this partial deterministic finite automaton is reduced and bideterministic. Therefore, the star height of L_n coincides with the cycle rank of the n-dimensional symmetric Cartesian hypercube. For a symmetric graph G, the cycle rank of G coincides with its (undirected) elimination tree height, which is in turn bounded below by the (undirected) pathwidth of G. Many structural properties of the n-dimensional hypercube are known, and among these is the recently established fact [1] that its pathwidth equals $\sum_{i=0}^{n-1} \binom{i}{\lceil i/2 \rceil} = \Theta(2^{n-1/2 \log n})$, where the latter estimate uses Stirling's approximation. Using Theorem 1, we obtain $\mathrm{alph}(L_n) = 2^{\Omega(2^{n-1/2 \log n})} = 2^{2^{\Omega(n)}}$, as desired. $\square$

For a similar result using binary alphabets, we will encode the above witness language in binary using a star height preserving homomorphism. Some extra care has to be taken, however. The ideal situation one might hope for is to find for each $\Gamma = \{a_1, a_2, \dots a_n\}$ a suitable star height preserving homomorphism $\rho : \Gamma^* \to \{0,1\}^*$ such that $\rho(x \sqcup\!\sqcup y) = \rho(x) \sqcup\!\sqcup \rho(y)$, for all $x, y \in \Gamma^*$. This aim however appears to be a bit too ambitious. In all cases we have tried, the right-hand side of the above equation can contain words which are not even valid codewords. In [7] this difficulty is avoided altogether by simulating regular expressions with intersection by those with interleaving, using a trick from [18]. The drawback here is that the simulation takes place at the expense of introducing an extra symbol and polynomially increased size of the resulting expression with interleaving. To overcome this difficulty, Warmuth and Haussler devised a particular encoding [22], which they called *shuffle resistant*, that has the above property once we restrict our attention to codewords. Inspired by a property of this encoding proved later by Mayer and Stockmeyer [18, Prop. 3.1], we are led to define in general a shuffle resistant encoding as follows:

Definition 11. *An injective homomorphism $\rho : \Gamma^* \to \Sigma^*$, for some alphabets Γ and Σ, is* shuffle resistant *if $\rho(L(r)) = L(\rho(r)) \cap \rho(\Gamma)^*$, for each regular expression r with interleaving over Γ.*

The following is proved in [18, Prop. 3.1] for the encoding proposed by Warmuth and Haussler in [22]:

Theorem 12. *Let $\Gamma = \{a_1, a_2, \dots, a_n\}$ and $\Sigma = \{a, b\}$. The homomorphism $\rho : \Gamma^* \to \Sigma^*$, which maps a_i to $a^{i+1}b^i$ is shuffle resistant.*

Incidentally, this encoding also preserves star height. The drawback is, however, that $\mathrm{alph}(h(r)) = \Theta(|\Sigma| \, \mathrm{alph}(r))$, for r a regular expression with interleaving. We now present a general family of more economic encodings, into alphabets of size at least 3, that enjoy similar properties.

Theorem 13. *Let Γ and Σ be two alphabets, and \$ be a symbol not in Σ. If $\rho : \Gamma^* \to (\Sigma \cup \{\$\})^*$ is an injective homomorphism with $\rho(\Gamma) \subseteq \Sigma^k\$$, for some integer k, then ρ is shuffle resistant.*

Proof. We need to show that for each such homomorphism ρ, the equality $\rho(L(r)) = L(\rho(r)) \cap \rho(\Gamma)^*$ holds for all regular expressions r with interleaving over Γ. The outline of the proof is roughly the same as the proof for Theorem 12 as sketched in [18]. The proof is by induction on the operator structure of r, using the stronger inductive hypothesis that

$$L(\rho(r)) \subseteq \rho(L(r)) \cup E, \quad \text{with} \quad E = (\rho(\Gamma))^* \Sigma^{\geq k+1} (\Sigma \cup \$)^* \qquad (1)$$

Roughly speaking, the "error language" E specifies that the first error occurring in a word in $L(\rho(r))$ but not in $(\rho(\Gamma))^*$ must consist in a sequence of too many consecutive symbols from Σ.

The base cases are easily established, and also the induction step is easy for the regular operators concatenation, union, and Kleene star. The more difficult part is to show that if two expressions r_1 and r_2 satisfy Equation (1), then this also holds for $r = r_1 \sqcup\!\sqcup\, r_2$. To prove this implication, it suffices to show the following claim:

Claim 14. For all words u, v in $\rho(\Gamma)^* \cup E$ and for each word z in $u \sqcup\!\sqcup\, v$ the following holds: If both $z \in (\Sigma^k\$)^*$ and $u, v \in \rho(\Gamma)^*$, then $z \in \rho\left(\rho^{-1}(u) \sqcup\!\sqcup\, \rho^{-1}(v)\right)$. Otherwise, $z \in E$.

Proof. We prove the claim by induction on the length of z. The base case with $|z| = 0$ is clear. For the induction step, assume $|z| > 0$ and consider the prefix y consisting of the first $k + 1$ letters of z. Such a prefix always exists if z is obtained from shuffling two nonempty words from $\rho(\Gamma)^* \cup E$. The cases where u or v is empty are trivial.

Observe first that it is impossible to obtain a prefix in $\Sigma^{<k}\$$ by shuffling two prefixes u' and v' of the words u and v. Also, a prefix in $\Sigma^{>k}$ always completes to a word $z \in E$.

It remains to consider the case z has a prefix y in $\Sigma^k\$$. To obtain such a prefix, two prefixes u' and v' have to be shuffled, with $(u', v') \in (\Sigma^j) \times (\Sigma^{k-j}\$)$ or $(u', v') \in (\Sigma^j\$) \times (\Sigma^{k-j})$. But since these are prefixes of words in $\rho(\Gamma)^* \cup E$, the index j can take on only the values $j = 0$ and $j = k$. Thus, if $y \in \Sigma^k\$$, then y is indeed in $\rho(\Gamma)$, and y is obtained by observing exclusively the first $k+1$ letters of u, or exclusively the first $k+1$ letters of v. Hence at least one of the subcases $y^{-1}z \in (y^{-1}u) \sqcup\!\sqcup\, v$ and $y^{-1}z \in u \sqcup\!\sqcup\, (y^{-1}v)$ holds. We only consider the first subcase, for the second one a symmetric argument applies.

It is not hard to see that we can apply the induction hypothesis to this subcase: Because $y \in \rho(\Gamma)$ and $u \in \rho(\Gamma)^* \cup E$, the word $y^{-1}u$ is again in the set $\rho(\Gamma)^* \cup E$. Having furthermore $|y^{-1}z| < |z|$, the induction hypothesis readily implies that claimed statement also holds for the word $z = y(y^{-1}z)$. This completes the proof of the claim. $\square$

Having established the claim, completing the proof of the statement $L(\rho(r)) \subseteq \rho(L(r)) \cup E$ is a rather easy exercise. $\square$

The existence of economic shuffle resistant binary encodings that furthermore preserve star height is shown next.

Theorem 15. *Let Γ be an alphabet. There exists a homomorphism $\rho : \Gamma^* \to \{0,1\}^*$ such that (1) $|\rho(a)| = O(\log |\Gamma|)$, for every symbol $a \in \Gamma$, and (2) the homomorphism ρ is shuffle resistant and preserves star height.*

Proof. Without loss of generality assume $\Gamma = \{a_1, a_2, \ldots a_{2^k}\}$ for some $k \geq 0$. In a first step, we encode into an alphabet of size three. Let $\sigma : \Gamma^* \to (\{0,1\} \cup \{\$\})^*$ be the homomorphism given by $\sigma(a_i) = \mathrm{bin}(i) \cdot \mathrm{bin}(i)\$$, with $\mathrm{bin}(i)$ being the usual k-bit binary encoding of the number i. Obviously, σ maps all alphabet symbols to strings of length $O(\log |\Gamma|)$. That the encoding is shuffle resistant follows from Theorem 13, and that it preserves star height is shown along the same lines as in the proof of Theorem 8, where we studied a rather similar encoding. In a second step, we use the homomorphism τ from Theorem 12 to further decrease the alphabet size from ternary to binary. This encoding is both shuffle resistant and preserves star height. The composed encoding $\rho = \tau \circ \sigma$ does the job: We have $\tau(\sigma(a)) = O(\log |\Gamma|)$, and it is readily proved that $\tau \circ \sigma$ is both shuffle resistant and preserves star height by expanding the definitions of these two notions. □

For regular expressions with interleaving we show that the conversion to ordinary regular expressions induces a $2^{2^{\Omega(n/\log n)}}$ lower bound for binary input alphabet.

Theorem 16. *There is an infinite family of languages L_n over a binary alphabet admitting regular expressions with interleaving of size $O(n)$, such that* $\mathrm{alph}(L_n) = 2^{2^{\Omega(n/\log n)}}$.

Proof. Our witness language will be described by the expression

$$\rho(r_n) = (\rho(a_1)\rho(b_1))^* \sqcup\!\sqcup (\rho(a_2)\rho(b_2))^* \sqcup\!\sqcup \cdots \sqcup\!\sqcup (\rho(a_n)\rho(b_n))^*,$$

obtained by applying the homomorphism ρ from Theorem 15 to the expression r_n used in the proof of Theorem 10. This expression has size $O(n \log n)$, and to prove the theorem, it will suffice to establish that $L(\rho(r_n))$ has alphabetic width at least $2^{2^{\Omega(n)}}$.

Recall from the proof of Theorem 10 that the star height of $L(r_n)$ is bounded below by $2^{\Omega(n)}$. Since ρ preserves star height, the same bound applies to the language $\rho(L(r_n))$. By Theorem 1, we thus have

$$\mathrm{alph}(\rho(L(r_n))) = 2^{2^{\Omega(n)}}. \tag{2}$$

Unfortunately, this bound applies to the language $\rho(L(r_n))$ rather than to $L(\rho(r_n))$. At least, as we know from Theorem 15 that ρ is a shuffle resistant encoding, these two sets are related by

$$L(\rho(r_n)) \cap \rho(\Gamma)^* = \rho(L(r_n)), \tag{3}$$

with $\Gamma = \{a_1, b_1, \ldots, a_n, b_n\}$.

To derive a similar lower bound on the language $L(\rho(r_n))$, we use the $2^{O(n(1+\log m))}$ upper bound from [12] on the alphabetic width of the intersection for regular languages of alphabet width m and n, respectively, for $m \geq n$. To this end, let $\alpha = \alpha(n)$ denote the alphabetic width of $L(\rho(r_n))$. We show first that $\alpha(n) \geq \mathrm{alph}(\rho(\mathit{\Gamma})^*)$. Assume the contrary. By Theorem 15, the set $\rho(\mathit{\Gamma})^*$ admits a regular expression of size $O(n \log n)$. Assuming $\alpha(n) \leq \mathrm{alph}(\rho(\mathit{\Gamma})^*)$, the upper bound on the alphabetic width of intersection implies that $\rho(L(r_n)) = L(\rho(r_n)) \cap \rho(\mathit{\Gamma}^*)$ admits a regular expression of size $2^{O(n \log^2 n)}$. But this clearly contradicts Inequality (2). Thus, $\alpha \geq \mathrm{alph}(\rho(\mathit{\Gamma})^*)$. Applying the upper bound for intersection to the left-hand side of Equation (3), we obtain

$$\mathrm{alph}(\rho(L(r_n))) = \mathrm{alph}(L(\rho(r_n)) \cap \rho(\mathit{\Gamma}^*)) = 2^{O(n \log n \log \alpha)}. \tag{4}$$

Inequalities (2) and (4) now together imply that there exist positive constants c_1 and c_2 such that, for n large enough, holds $2^{2^{c_1 n}} \leq 2^{c_2 n \log n \log \alpha}$. Taking double logarithms on both sides and rearranging terms, we obtain $c_1 n - O(\log n) \leq \log \log \alpha$. Since the the left-hand side is in $\mathit{\Omega}(n)$, we thus have $\mathrm{alph}(L(\rho(r_n))) = \alpha = 2^{2^{\mathit{\Omega}(n)}}$, and the proof is completed. $\square$

References

1. L. S. Chandran and T. Kavitha. The treewidth and pathwidth of hypercubes. *Discrete Mathematics*, 306(3):359–365, 2006.
2. R. S. Cohen. Star height of certain families of regular events. *Journal of Computer and System Sciences*, 4(3):281–297, 1970.
3. L. C. Eggan. Transition graphs and the star height of regular events. *Michigan Mathematical Journal*, 10:385–397, 1963.
4. A. Ehrenfeucht and H. P. Zeiger. Complexity measures for regular expressions. *Journal of Computer and System Sciences*, 12(2):134–146, 1976.
5. K. Ellul, B. Krawetz, J. Shallit, and M.-W. Wang. Regular expressions: New results and open problems. *Journal of Automata, Languages and Combinatorics*, 10(4):407–437, 2005.
6. M. Fürer. The complexity of the inequivalence problem for regular expressions with intersection. In J. W. de Bakker and J. van Leeuwen, editors, *International Colloquium on Automata, Languages and Programming*, number 85 of *LNCS*, pages 234–245. Springer, 1980.
7. W. Gelade. Succinctness of regular expressions with interleaving, intersection and counting. In E. Ochmanski and J. Tyszkiewicz, editors, *Mathematical Foundations of Computer Science*, number 5162 of *LNCS*, pages 363–374. Springer, 2008.
8. W. Gelade and F. Neven. Succinctness of the complement and intersection of regular expressions. In S. Albers and P. Weil, editors, *Symposium on Theoretical Aspects of Computer Science*, volume 08001 of *Dagstuhl Seminar Proceedings*, pages 325–336. IBFI Schloss Dagstuhl, Germany, 2008.
9. I. Glaister and J. Shallit. A lower bound technique for the size of nondeterministic finite automata. *Information Processing Letters*, 59(2):75–77, 1996.
10. H. Gruber and M. Holzer. Finite automata, digraph connectivity, and regular expression size. In L. Aceto, I. Damgård, L. A. Goldberg, M. M. Halldórsson, A. Ingólfsdóttir, and I. Walkuwiewicz, editors, *International Colloquium on Automata, Languages and Programming*, number 5126 of *LNCS*, pages 39–50. Springer, 2008.
11. H. Gruber and M. Holzer. Language operations with regular expressions of polynomial size. In C. Câmpeanu and G. Pighizzini, editors, *Descriptional Complexity of Formal Systems*, pages 182–193, Charlottetown, PEI, Canada, 2008. CSIT University of Prince Edward Island.

12. H. Gruber and M. Holzer. Provably shorter regular expressions from deterministic finite automata (Extended abstract). In M. Ito and M. Toyama, editors, *Developments in Language Theory*, number 5257 of *LNCS*. Springer, 2008.
13. H. Gruber and J. Johannsen. Optimal lower bounds on regular expression size using communication complexity. In R. Amadio, editor, *Foundations of Software Science and Computation Structures*, number 4962 of *LNCS*, pages 273–286. Springer, 2008.
14. K. Hashiguchi. Algorithms for determining relative star height and star height. *Information and Computation*, 78(2):124–169, 1988.
15. K. Hashiguchi and N. Honda. Homomorphisms that preserve star height. *Information and Control*, 30(3):247–266, 1976.
16. J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages and Computation.* Addison-Wesley, 1979.
17. T. Johnson, N. Robertson, P. D. Seymour, and R. Thomas. Directed tree-width. *Journal of Combinatorial Theory, Series B*, 82(1):138–154, 2001.
18. A. J. Mayer and L. J. Stockmeyer. Word problems - This time with interleaving. *Information and Computation*, 115(2):293–311, 1994.
19. R. McNaughton. The loop complexity of pure-group events. *Information and Control*, 11(1/2):167–176, 1967.
20. R. McNaughton. The loop complexity of regular events. *Information Sciences*, 1:305–328, 1969.
21. A. R. Meyer and M. J. Fischer. Economy of description by automata, grammars, and formal systems. In *IEEE Symposium on Switching and Automata Theory*, pages 188–191. IEEE Computer Society, 1971.
22. M. K. Warmuth and D. Haussler. On the complexity of iterated shuffle. *Journal of Computer and System Sciences*, 28(3):345–358, 1984.