Optimal Regular Expressions for Palindromes of Given Length

Hermann Gruber
Knowledgepark GmbH, München, Germany

Markus Holzer[1]
Institut für Informatik, University of Giessen, Germany

Abstract

The language P_n ($\widetilde{P}_n$, respectively) consists of all words that are palindromes of length $2n$ ($2n-1$, respectively) over a fixed binary alphabet. We construct a regular expression that specifies P_n ($\widetilde{P}_n$, respectively) of alphabetic width $4 \cdot 2^n - 4$ ($3 \cdot 2^n - 4$, respectively) and show that this is optimal, that is, the expression has minimum alphabetic width among all expressions that describe P_n ($\widetilde{P}_n$, respectively). To this end we give optimal expressions for the first k palindromes in lexicographic order of odd and even length, proving that the optimal bound is $2n + 4(k-1) - 2S_2(k-1)$ in case of odd length and $2n + 3(k-1) - 2S_2(k-1) - 1$ for even length, respectively. Here $S_2(n)$ refers to the Hamming weight function, which denotes the number of ones in the binary expansion of the number n.

2012 ACM Subject Classification Theory of computation → Regular languages; Mathematics of computing → Nonlinear equations

Keywords and phrases regular expression, descriptional complexity, lower bound, upper bound, recurrence, sum of digits

Digital Object Identifier 10.4230/LIPIcs.MFCS.2021.52

Acknowledgements We would like to thank the anonymous reviewers for their valuable suggestions.

1 Introduction

During the last two decades or so, literally hundreds of research papers have been investigating deterministic and nondeterministic state complexity of regular languages. Here, general purpose lower bound techniques are available, and in many cases, upper and lower bounds can be obtained that match exactly, not only asymptotically. For recent surveys, see, e.g., [8, 15].

The situation is less desirable if we investigate the minimum required size of regular expressions describing a regular language. While several different lower bound techniques are available, often the best known upper and lower bounds match only asymptotically. For illustration, the size blow-up when going from finite automata over a binary alphabet to regular expressions is at least c^n for some $c > 1$ for large enough n, cf. [12]. The current record holder for the upper bound is $O(1.682^n)$, see [5]. This gives a "tight" bound of $2^{\Theta(n)}$, which is on closer inspection a bit loose. To our knowledge, exactly matching upper and lower bounds for the minimum required expression size are known only for very few nontrivial language families: Namely, the Boolean n-bit parity function [7, 14], the less-than relation on an n-set [2], given as $\{ ij \mid 1 \leq i < j \leq n \}$, and the permutations of an n-set [23].

The set of all palindromes over the alphabet $\{a, b\}$ is context-free but not regular; virtually every computer science student in the world will learn this during their curriculum. Not surprisingly, this basic observation is as old as the Chomsky hierarchy itself [3]. Of course, if we consider only palindromes of a given length, the set thus obtained is finite, and therefore

[1] Corresponding author.

46th International Symposium on Mathematical Foundations of Computer Science (MFCS 2021).
Editors: Filippo Bonchi and Simon J. Puglisi; Article No. 52; pp. 52:1–52:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

regular. We exactly determine the optimum regular expressions for this set, for every given length. In the course of the proof, we also determine the optimum regular expressions for the lexicographically first k palindromes of a given length, for every k. The difficulty of course lies in establishing a matching lower bound. To this end, we use and expand a method from [23] to obtain a recurrent lower bound. The recurrence thus obtained involves a "minvolution" in the sense of [11] and the minimum operator of course yields a nonlinear recurrence. A long line of research concerns asymptotic and exact solutions of recurrences involving minimum and maximum functions, see, e.g., [18] and references therein. Our recurrence falls into neither of the known categories. So we develop a tailor-made strategy for solving the recurrence, and derive a novel identity involving sums of Hamming weights. We hope that this will serve as a helpful example for researchers in need of solving similar nonlinear recurrences.

Some of our results contribute to the knowledge about integer sequences: We give a characterization of the number of multiplications to compute the $(n+1)$th power by the ancient Indian Chandah-sutra method in terms of Hamming weights (Lemma 8). Also, we find a new recurrence for the numbers having a partition into distinct Mersenne numbers greater than zero (Lemma 10). The functions giving the optimal lengths of the regular expressions we consider can be enumerated in lexicographic order; accompanying submissions to the On-line Encyclopedia of Integer Sequences (OEIS) are in preparation, since these sequences are not yet covered by OEIS.

With some extra effort, all of our results can be generalized to larger alphabet sizes. These results will be presented in the full version of this paper.

2 Preliminaries

We assume that the reader is familiar with the basic notions of formal language theory as contained in [16]. In particular, let Σ be an *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 $|\epsilon| = 0$, and the total number of occurrences of the alphabet symbol a in w is denoted by $|w|_a$. In this paper, we mainly deal with finite languages. The *order* of a finite language L is the length of a longest word belonging to L. A finite language L is *homogeneous* if all words in the language have the same length. In order to fix the notation, we briefly recall the definition of regular expressions and the languages described by them.

The *regular expressions* over an alphabet Σ are defined inductively in the usual way:[2] $\emptyset$, ϵ, and every letter a with $a \in \Sigma$ is a regular expression; and when E and F are regular expressions, then $(E+F)$, $(E \cdot F)$, and $(E)^*$ are also regular expressions. The language defined by a regular expression E, denoted by $L(E)$, is defined as follows: $L(\emptyset) = \emptyset$, $L(\epsilon) = \{\epsilon\}$, $L(a) = \{a\}$, $L(E+F) = L(E) \cup L(F)$, $L(E \cdot F) = L(E) \cdot L(F)$, and $L(E^*) = L(E)^*$. The *alphabetic width* or *size* of a regular expression E over the alphabet Σ, denoted by $\mathsf{awidth}(E)$, is defined as the total number of occurrences of letters of Σ in E. For a regular language L, we define its alphabetic width, $\mathsf{awidth}(L)$, as the minimum alphabetic width among all regular expressions describing L.

[2] 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.

3 A Lower Bound for Palindromes of Even Length

For a nonnegative integer n, let $P_n = \{\, ww^R \mid w \in \{a,b\}^n \,\}$ denote the set of palindromes of length $2n$. In this section, we give a tight bound on the required regular expression size of P_n. For our toolbox, we need to investigate the concatenation of homogeneous languages.

▶ **Lemma 1.** *Let L_1 and L_2 be homogeneous languages. Then* $\mathsf{awidth}(L_1 \cdot L_2) = \mathsf{awidth}(L_1) + \mathsf{awidth}(L_2)$.

Inspired by the method recently used to exactly determine the alphabetic width of the set of permutations [23], define $\ell(n,k)$ to be the minimum alphabetic width of a regular expression describing a subset of P_n, where the subset has cardinality at least k. Note that $\ell(n,k)$ is monotone in k by definition, that is, $\ell(n,k) \leq \ell(n,k')$, for $k \leq k'$.

▶ **Lemma 2.** *Let $n \geq 0$ and $1 \leq k \leq 2^n$. Then $\ell(n,k)$ obeys the following recurrence:*

$$\ell(n,k) \geq \min\{\, \ell(n-1,k)+2, \min_{1\leq i<k}\{\ell(n,i)+\ell(n,k-i)\}\,\}, \text{ for } n \geq 2 \text{ and } 2 \leq k \leq 2^{n-1},$$

$$\ell(n,k) \geq \min_{1\leq i<k}\{\ell(n,i)+\ell(n,k-i)\}, \text{ for } n \geq 1 \text{ and } k > 2^{n-1},$$

and

$$\ell(n,1) = 2n.$$

Proof. In the case $k = 1$, each regular expression describing a nonempty subset of $\{a,b\}^{2n}$ must have alphabetic width at least $2n$. For $n \geq 1$, the expression a^{2n} describes at least one word in P_n. For $n = 0$, ϵ is an optimal regular expression describing the only nonempty subset of $\{a,b\}^0 = \{\epsilon\}$. Thus we have $\ell(n,1) = 2n$ for all $n \geq 0$.

For $n \geq 1$ and $2 \leq k \leq 2^n$, let E be a regular expression denoting a subset of P_n which has cardinality at least k. The language P_n is homogeneous, so we may safely assume that neither ϵ nor $\emptyset$ occur in E, and the same holds for the Kleene star, see, e.g., [14]. Thus, E is of the form $F + G$ or of the form $F \cdot G$, and each of F and G have alphabetic width at least 1.

If $E = F+G$, then both F and G denote subsets of P_n, say of sizes k_1 and k_2, respectively. Then $k_1 + k_2 \geq k$, and, by minimality, $k_1, k_2 < k$. We thus obtain the following recurrence in the case of union:

$$\begin{aligned}\ell(n,k) &\geq \ell(n,k_1) + \ell(n,k_2)\\ &\geq \ell(n,k_1) + \ell(n,k-k_1)\\ &\geq \min_{1\leq i<k}\{\ell(n,i)+\ell(n,k-i)\},\end{aligned}$$

where we used the monotonicity of $\ell(n,k)$ with respect to k for the second estimation.

The other case is that $E = F\cdot G$. We may assume that the words in F have length at most n – otherwise, we apply the argument to $E^R = G^R \cdot F^R$, and note that $\mathsf{awidth}(E) = \mathsf{awidth}(E^R)$. Let n_1 denote the length of the words in F. Then we have $1 \leq n_1 \leq n$. We claim that $L(F)$ must be a singleton language, that is, $L(F) = \{w\}$ for some word w. For the sake of contradiction, assume $L(F)$ contains another word x with $x \neq w$. Since E describes P_n and $L(E) = L(F) \cdot L(G)$, the language $L(E)$ contains a word of the form wzw^R, for some infix z. Since there is only one way to write wzw^R as product of words in $L(F)$ and $L(G)$, the word zw^R must be in $L(G)$. But then the non-palindromic word xzw^R is a member of $L(E)$, which yields the desired contradiction to establish the claim.

Further, we can assume that $n_1 = 1$ without loss of generality. This can be seen as follows. By Lemma 1, $\mathsf{awidth}(L(E)) = \mathsf{awidth}(L(F)) + \mathsf{awidth}(L(G))$. For $1 \leq i \leq n_1$, let a_i denote the ith letter in w. Since $L(F) = \{w\}$, we have $\mathsf{awidth}(L(F)) = n_1$. Thus, if we replace the subexpression F of E with the expression $\widetilde{f} = a_1 \cdot (a_2 \cdots a_{n_1})$, the expression $\widetilde{E}$ thus obtained is again minimal. By applying the associative law for concatenation to $\widetilde{E}$, we obtain yet another minimal expression $\widetilde{E}' = \widetilde{f}' \cdot G''$ with $\widetilde{f}' = a_1$ and $G'' = (a_2 \cdots a_{n_1} \cdot G)$.

Since all words in P_n are palindromic, $L(G'') = S \cdot a_1$, for some subset S of P_{n-2}. Also, set S must be of the same cardinality as $L(E)$. We thus obtain the following recurrence in the case of concatenation:

$$\ell(n,k) \geq \ell(n-1,k) + 2.$$

Observe that k can be at most 2^{n-1} in this case, since there are no more than 2^{n-1} palindromes of length $2(n-1)$. Also, we must have $n \geq 2$ in the case of concatenation, since both $k \geq 2$ and $k \leq 2^{n-1}$ hold.

Either the case of union or of concatenation applies – because E has no Kleene star, and we obtain the recurrence relation

$$\ell(n,k) \geq \min\{\, \ell(n-1,k) + 2,\ \min_{1 \leq i < k} \{\ell(n,i) + \ell(n,k-i)\}\,\}, \text{ for } n \geq 2 \text{ and } 2 \leq k \leq 2^{n-1},$$

and

$$\ell(n,k) \geq \min_{1 \leq i < k} \{\ell(n,i) + \ell(n,k-i)\}, \text{ for } n \geq 1 \text{ and } 2^{n-1} < k \leq 2^n,$$

as desired. ◀

In the above proof, we derived a recursive lower bound on $\ell(n,k)$. Let f denote the integer-valued function which is defined by that recurrence, that is,

$$f(n,k) = \min\{\, f(n-1,k) + 2,\ \min_{1 \leq i < k} \{f(n,i) + f(n,k-i)\}\,\}, \text{ for } n \geq 2 \text{ and } 2 \leq k \leq 2^{n-1},$$

$$f(n,k) = \min_{1 \leq i < k} \{f(n,i) + f(n,k-i)\}, \text{ for } n \geq 1 \text{ and } 2^{n-1} < k \leq 2^n,$$

and

$$f(n,1) = 2n.$$

The recursive definition can be simplified with the aid of the following lemma.

▶ **Lemma 3.** *$f(n,k) = f(n-1,k) + 2$, for $n \geq 2$ and $2 \leq k \leq 2^{n-1}$.*

Proof. Recall the recursive definition of f in this parameter range is

$$f(n,k) = \min\{\, f(n-1,k) + 2,\ \min_{1 \leq i < k} \{f(n,i) + f(n,k-i)\}\,\},$$

for $n \geq 2$ and $2 \leq k \leq 2^{n-1}$, so the inequality $f(n,k) \leq f(n-1,k) + 2$ is immediate. For the converse inequality, we claim that

$$\min_{1 \leq i < k} \{f(n,i) + f(n,k-i)\} \geq f(n-1,k) + 2.$$

We prove this by lexicographic induction on (n,k). To show the statement for $n \geq 2$ and $k \geq 2$, we assume that the statement holds for all pairs (n',k') with $n' < n$, as well as for all pairs with $n' = n$ and $k' < k$. Observe, that by the induction hypothesis on (n,k) we also can safely assume that $f(n',k') \geq f(n'-1,k') + 2$, which follows from the recursive definition of f. The base case $(2,2)$ is easily verified with

$$\min_{1 \leq i < 2} \{f(2,i) + f(2,2-i)\} = f(2,1) + f(2,1) = 4 + 4 \geq 4 + 2 = f(1,2) + 2,$$

because $f(1,2) = \min_{1\leq i<2}\{f(1,i) + f(1,2-i)\} = f(1,1) + f(1,1) = 2 + 2 = 4$. For the induction step, we apply the induction hypothesis and $f(n,k) \geq f(n-1,k) + 2$ twice to obtain

$$\begin{aligned}\min_{1\leq i<k}\{f(n,i) + f(n,k-i)\} &\geq \min_{1\leq i<k}\{(f(n-1,i)+2) + (f(n-1,k-i)+2)\}\\ &\geq \min_{1\leq i<k}\{f(n-1,i) + f(n-1,k-i)\} + 4\\ &\geq f(n-1,k) + 6,\end{aligned}$$

which means that $\min_{1\leq i<k}\{f(n,i) + f(n,k-i)\} \geq f(n-1,k) + 2$ as desired.

Having established the claim, the equality of $f(n,k)$ with $f(n-1,k)+2$ now follows immediately. This completes the proof of the lemma. ◀

Still, the second recurrence equation entails the full history of the parameter k. One might hope that f is convex in the parameter k, and that the minimum in the formula $\min_{1\leq i<k}\{f(n,i) + f(n,k-i)\}$ is always attained in the middle, i.e., $\arg\min i = \lfloor \frac{k}{2} \rfloor$. Compare, e.g., [21, p. 366] on convex recurrences. But this is, unfortunately, not the case: for instance, we have $f(3,6) = \min_{1\leq i<6}\{f(3,i) + f(3,6-i)\} = f(3,2) + f(3,4) = 8 + 14 = 22$, while $2 \cdot f(3,3) = 2 \cdot (f(2,3) + 2) = 2 \cdot (10 + 2) = 24$. In fact, computations for small ranges of n and k suggest a nontrivial behavior of f – see Table 1.

Table 1 Some $f(n,k)$ values for small n and k.

	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$	$k=9$	$k=10$	$k=11$	$k=12$	$k=13$	$k=14$	$k=15$	$k=16$
$n=1$	2	4														
$n=2$	4	6	10	12												
$n=3$	6	8	12	14	20	22	26	28								
$n=4$	8	10	14	16	22	24	28	30	38	40	44	46	52	54	58	60

At least, we are interested only in the value of $f(n,2^n)$. Once we put forward a suitable induction hypothesis (which admittedly is somewhat flabbergasting), we can establish a simple closed form for $f(n,2^n)$ with a laborious lexicographic induction.

▶ **Lemma 4.** $f(n,2^n) = 2^{n+2} - 4$.

Proof. For the upper bound, observe that $f(n,2^n) \leq 2 \cdot f(n,2^{n-1})$ easily follows from the recurrence equations defining f and with the help of Lemma 3 we obtain

$$f(n,2^n) \leq 2 \cdot (f(n-1,2^{n-1}) + 2).$$

With $f(1,2) = 4$, this boils down to an inhomogeneous linear recurrence with variable n, which can be solved as $f(n,2^n) \leq 4(2^n - 1) = 2^{n+2} - 4$.

The lower bound will follow immediately once we have established the following claim.

▷ Claim 5. Let $n \geq 1$ and $1 \leq k \leq 2^n$. Then

$$f(n,k) \geq \begin{cases} 4k & \text{if } k < 2^{n-1},\\ 4k-2 & \text{if } k \text{ is not a power of two and } k > 2^{n-1}, \text{ and}\\ 4k-4+2n-2\log k & \text{if } k \text{ is a power of two.}\end{cases}$$

The remaining part of the proof is a lexicographic induction on (n,k), which tedious details are left to the reader. ◀

4 Some Digit Theory

Now that our appetite is whetted, we want to solve the recurrence also in the general case where k is not a power of two. In this section, we develop the necessary tools regarding "digit theory," that is, mathematical properties of digit sums, that we will need for the analysis. Let $S_2(n)$ denotes the "digit sum to base 2" function. This function is often referred to as the *Hamming weight function* and denotes the number of ones in the binary expansion of the number n. Throughout the rest of this paper, for a nonnegative integer n, we refer to the function $\lambda(n)$ defined as

$$\lambda(n) = \begin{cases} 0, & \text{if } n = 0 \\ \lfloor \log_2 n \rfloor, & \text{otherwise.} \end{cases}$$

Here $\log_2 n$ refers to the logarithm to base 2. For the digit sum to base 2 we find the following equations useful whenever powers of 2 are involved somehow.

▶ **Lemma 6.** *Let n be a nonnegative integer. Then*
1. $S_2(2^n - 1) = n$ *and*
2. $S_2(n - 2^{\lambda(n)}) = S_2(n) - 1$.

Next we recall an alternative characterization of the digit sum to base 2 that proves useful in the forthcoming calculations.

▶ **Lemma 7.** *Let n be a nonnegative integer. Then*

$$S_2(n) = n - \sum_{i=1}^{\infty} \left\lfloor \frac{n}{2^i} \right\rfloor .$$

Observe, that the sum contains only a finite number of non-zero summands.

More generally, for prime q the sum $\sum_{i=1}^{\infty} \left\lfloor \frac{n}{q^i} \right\rfloor$ is famously known to be equal to the largest power of q that divides $n!$ (Legendre's formula [22]). For non-prime q, the latter equality ceases to hold in general, because for $n = 8$ and $q = 4$ the largest integer power of 4 that divides 8! is 3, because $8! = (2 \cdot 4) \cdot 7 \cdot (2 \cdot 3) \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 7 \cdot 5 \cdot 4^3 \cdot 3^2 \cdot 2$, while the sum evaluates to 2.

The study of the following maximization problem

$$\max_{0 \le i \le n} \{S_2(i) + S_2(n - i)\}$$

is essential for our main result. Remarkably, the formula on the right-hand side of the identity in Lemma 8 below is famously known as the number of multiplications to compute the $(n + 1)$th power by the ancient Indian *Chandah-sutra method*. This appears as sequence A014701 in the On-line Encyclopedia of Integer Sequences, and is referred to as the *left-to-right binary method*[3] in [20, Chap. 4.6.3].

▶ **Lemma 8.** *Let n be a nonnegative integer. Then*

$$\max_{0 \le i \le n} \{S_2(i) + S_2(n - i)\} = \lambda(n + 1) + S_2(n + 1) - 1.$$

[3] We note that the formula given in [20, p. 463] refers to the *right-to-left binary method*. As explained there, the latter takes one more multiplication than the left-to-right binary method.

Proof. Observe, that $S_2(n)$ denotes the Hamming weight of n, that is, the number of ones in the binary expansion of the number n. We shall prove first the easier inequality, namely $\max_{0\leq i\leq n}\{S_2(i)+S_2(n-i)\}\geq \lambda(n+1)+S_2(n+1)-1$. It suffices to find a suitable decomposition $n=j+(n-j)$ for some j, which attains the bound. We choose $j=n+1-2^{\lambda(n+1)}$. Then j is equal to $n+1$ modulo $2^{\lambda(n+1)}$, and thus their binary expansions differ only in the highest order bit. In other words, $S_2(j)=S_2(n+1)-1$ by Lemma 6.2. Also, by the finite geometric series expansion,

$$n-j=n-\left(n+1-2^{\lambda(n+1)}\right)=2^{\lambda(n+1)}-1=\sum_{i=0}^{\lambda(n+1)-1}1\cdot 2^i,$$

and thus $S_2(n-j)=\lambda(n+1)$ – see Lemma 6.1.

The converse inequality requires more effort, namely to prove that

$$\max_{0\leq i\leq n}\{S_2(i)+S_2(n-i)\}\leq \lambda(n+1)+S_2(n+1)-1.$$

Our strategy is as follows. Given any decomposition $n=x+y$ with $x,y\geq 0$, we write x and y in binary positional notation $x_{\lambda(x)}\cdots x_1x_0$ and $y_{\lambda(y)}\cdots y_1y_0$. Then we shall apply a certain set of rules to x and y such that

- after each rule application, the sum of the two summands x' and y' thus obtained is n, that is, $x'+y'=n$,
- after each rule application, the sum of their Hamming weights is not decreased, that is, $S_2(x')+S_2(y')\geq S_2(x)+S_2(y)$, and
- after the last rule application, in the larger summand thus obtained, all bits are equal to 1.

This will of course suffice to show that the decomposition into j and $n-j$, as described at the beginning of the proof of this lemma, attains the maximum.

When looking at the bits of x and y, there are several constellations that need to be addressed. The first rule concerns the case $x_0=y_0=0$, that is, the lowest order bits are both zero. Assume x is greater than or equal to y, otherwise we exchange the roles of x and y. Let ℓ denote the lowest order nonzero bit position of x, that is $x_\ell=1$ and $x_k=0$, for all k with $0\leq k<\ell$. Then decreasing the number x by 1 amounts to setting $x_\ell=0$ and $x_k=1$, for all k with $0\leq k<\ell$. Also, increasing the number y by 1 amounts to setting $y_0=1$, while all other bits of y remain unchanged. Observe, that this maneuver increases the Hamming weight of both summands. Also, the two summands thus obtained add up to n, and both summands have their lowest order bit set to 1. For an illustration of this situation we refer the reader to the left drawing of Figure 1.

We now generalize this to the case $x_i=y_i=0$, for $0\leq i<\lambda(x)$. Here again, we assume that $x\geq y$; otherwise we exchange the roles of x and y. Here essentially the same mechanism applies, but, roughly speaking, we need to "multiply everything" by 2^i. In the same spirit as above, let ℓ denote the lowest order nonzero bit position of x above i, that is $x_\ell=1$ and $x_k=0$, for all k with $i\leq k<\ell$. Note that ℓ is guaranteed to exist, since $x\geq y$. Observe, that decreasing the number x by 2^i amounts to setting $x_\ell=0$ and $x_k=1$, for all k with $i\leq k<\ell$. Also, increasing the number y by 2^i amounts to setting $y_i=1$, while all other bits of y remain unchanged. So this maneuver increases the Hamming weight of both summands. Also, the obtained summands sum up to n, and both summands have their ith bit set to 1. This completes the description of the first rule. For an illustration of this situation we refer to the right drawing of Figure 1.

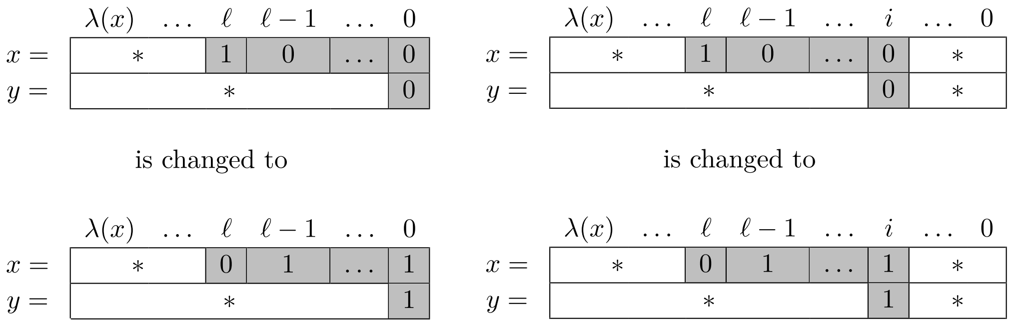

Figure 1 First bit manipulation rule for the decomposition of n into x and y for the first situation (left), i.e., $x_0 = y_0 = 0$, and the general situation (right), i.e., $x_i = y_i = 0$, for $0 \leq i < \lambda(x)$.

We iteratively apply this rule to the resulting pair of summands from the previous round, for each i in increasing order, requiring that $x \geq y$ at the beginning at every round; otherwise the *rôles* of x and y are exchanged. After the $(i+1)$th round, no constellations remain with $x_r = y_r = 0$, for $0 \leq r \leq i$. Finally, for every i with $0 \leq i \leq \lambda(x)$, no constellations remain with $x_i = y_i = 0$.

When y denotes the smaller summand obtained by the above procedure, the constellations where $y_i = 1$ do not need to be fixed. The remaining constellations are those where $y_i = 0$ and $x_i = 1$, for some $i \leq \lambda(x)$. The second rule is to exchange the bit values, that is, we set $y_i = 1$ and $x_i = 0$. It is clear that the two summands thus obtained add up to n. Also, the sum of the Hamming weights is unaffected. We apply the second rule as often as needed, and the number of these rule applications is of course bounded by $\lambda(x) + 1$. For an illustration of the second rule we refer to Figure 2.

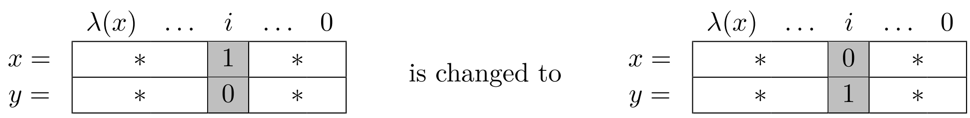

Figure 2 Second bit manipulation rule for the decomposition of n into x and y that is applied as often as needed.

After all applications of the second rule, we end up with the larger summand having all bits set to 1. Since the other two conditions are invariant under application of both rules, this completes the proof. ◀

5 Optimal Expressions for the first k Palindromes in Lexicographic Order

Now that we have collected the necessary tools, we aim to solve the recurrence $f(n, k)$ also for the case where k is not a power of two. Recall that Lemma 3 allows us to write up the recurrence in simplified form, as follows:

$$f(n,k) = f(n-1,k) + 2, \text{ for } n \geq 2 \text{ and } 2 \leq k \leq 2^{n-1},$$
$$f(n,k) = \min_{1 \leq i < k} \{f(n,i) + f(n,k-i)\}, \text{ for } n \geq 1 \text{ and } 2^{n-1} < k \leq 2^n,$$

and

$$f(n,1) = 2n.$$

We transform this recurrence into a recurrence on one unknown within two steps. In the first step, we define another function in two unknowns in terms of $f(n,k)$.

▶ **Lemma 9.** *Let $g(n,k) := \frac{1}{2}f(n,k) - n$. Then $g(n,k)$ satisfies the recurrence*

$$g(n,k) = g(n-1,k), \text{ for } n \geq 2 \text{ and } 2 \leq k \leq 2^{n-1},$$
$$g(n,k) = n + \min_{1\leq i<k}\{g(n,i) + g(n,k-i)\}, \text{ for } n \geq 1 \text{ and } 2^{n-1} < k \leq 2^n,$$

and

$$g(n,1) = 0.$$

We shall apply the second transformation only for the "interesting" parameter range of k, that is, when k is in the upper half of the admissible range. Namely, we observe, that whenever $2^{n-1} < k \leq 2^n$, then we can express n in terms of k as $n = 1 + \lambda(k-1)$. Recalling that $\lambda(0) = 0$, we set

$$h(k) := g(1+\lambda(k-1), k), \text{ for } k \geq 1.$$

Then we find the following situation:

▶ **Lemma 10.** *Let $h(k) := g(1+\lambda(k-1),k)$, for $k \geq 1$. Then $h(k)$ satisfies the recurrence*

$$h(1) = 0$$
$$h(k) = 1 + \lambda(k-1) + \min_{1\leq i<k}\{h(i) + h(k-i)\} \text{ for } k \geq 2,$$

and it has the solution $h(k) = 2(k-1) - S_2(k-1)$.

It is worth mentioning that the formula in Lemma 10 implies that the values of the recurrence h, starting from $h(1)$, coincide with the (zero-based) sequence A005187 in the On-Line Encyclopedia of Integer Sequences – the numbers having a partition into distinct Mersenne numbers greater than zero.

Now let's undo both transformations. We first determine $g(n,k)$ by using $h(k)$ and its explicit solution. Then, by elementary calculations we arrive at an alternative recurrence for $f(n,k)$.

▶ **Lemma 11.** *The function $f(n,k)$ satisfies the recurrence*

$$f(n,k) = f(n-1,k) + 2, \text{ for } n \geq 2 \text{ and } 2 \leq k \leq 2^{n-1},$$
$$f(n,k) = 2n + 4(k-1) - 2S_2(k-1), \text{ for } n \geq 1 \text{ and } 2^{n-1} < k \leq 2^n,$$

and

$$f(n,1) = 2n$$

and it has the solution $f(n,k) = 2n + 4(k-1) - 2S_2(k-1)$.

Proof. In order to undo both transformations, we first determine $g(n,k)$ by using $h(k)$ and its explicit solution from Lemma 10. We find

$$g(n,k) = g(n-1,k), \text{ for } n \geq 2 \text{ and } 2 \leq k \leq 2^{n-1},$$
$$g(n,k) = 2(k-1) - S_2(k-1), \text{ for } n \geq 1 \text{ and } 2^{n-1} < k \leq 2^n,$$

and

$$g(n,1) = 0,$$

because, for $k > 2^{n-1}$ we have

$$\begin{aligned} g(n,k) &= g(1+\lambda(k-1),k) \\ &= h(k) \\ &= 2(k-1) - S_2(k-1). \end{aligned}$$

Finally, recall that $f(n,k) = 2(g(n,k)+n)$, which results in

$$f(n,k) = f(n-1,k)+2, \text{ for } n \geq 2 \text{ and } 2 \leq k \leq 2^{n-1},$$

and for $k > 2^{n-1}$ we calculate

$$\begin{aligned} f(n,k) &= 2(g(n,k)+n) \\ &= 2\left(2(k-1) - S_2(k-1) + n\right) \\ &= 2n + 4(k-1) - 2S_2(k-1). \end{aligned}$$

For the terminating cases of the recurrence, we simply recall those from the original recurrence defining $f(n,k)$:

$$f(n,1) = 2n,$$

and this completes the proof.

It remains to solve the alternative recurrence, which is now done with ease. The statement is proved by lexicographic induction on (k,n). Let $k=1$, then $2n+4(k-1)-2S_2(k-1) = 2n$ is obviously an solution for any n. To show the statement for $k \geq 2$ and $n \geq 1$, we assume that the statement holds for all pairs (k',n') with $k' < k$, as well as for all pairs with $k' = k$ and $n' < n$. Then for the case $k > 2^{n-1}$ we have nothing to prove and in case $2 \leq k \leq 2^{n-1}$, we apply the induction hypothesis and get

$$\begin{aligned} f(n,k) &= f(n-1,k)+2 \\ &= 2(n-1) + 4(k-1) - 2S_2(k-1) + 2 \\ &= 2n + 4(k-1) - 2S_2(k-1) \end{aligned}$$

as desired. ◀

With the lower bound in place, it remains to give an optimal regular expression matching the lower bound. The expression $E_{n,k}$ describes the lexicographically first k palindromes of length $2n$, and is defined recursively as follows:

$$\begin{aligned} E_{n,k} &= a \cdot E_{n-1,k} \cdot a, \text{ for } n \geq 1 \text{ and } 1 \leq k \leq 2^{n-1}, \\ E_{n,k} &= a \cdot E_{n-1,2^{n-1}} \cdot a + b \cdot E_{n-1,k-2^{n-1}} \cdot b, \text{ for } n \geq 1 \text{ and } 2^{n-1} < k \leq 2^n, \end{aligned}$$

and

$$E_{0,1} = \epsilon.$$

We can prove by induction that this recursive upper bound on the alphabetic width meets the lower bound:

▶ **Lemma 12.** *For $n \geq 0$ and $k \geq 1$,* $\mathsf{awidth}(E_{n,k}) = f(n,k)$.

It remains to show that the definition of $E_{n,k}$ is semantically correct, in the sense that it describes exactly the set of the lexicographically first k palindromes.

▶ **Lemma 13.** *Let n, k be integers with $n \geq 0$ and $1 \leq k \leq 2^n$. Then the regular expression $E_{n,k}$ describes the lexicographically first k palindromes of length $2n$.*

Proof. We begin with a natural bijection between palindromes of length $2n$, for $n \geq 1$, and the nonnegative integers in the range $0, 1, \ldots, 2^n - 1$: for a nonnegative integer j with $0 \leq j < 2^n$, with binary expansion $\sum_{r=0}^{\infty} j_r 2^r = k$, let $\rho : \{0 \mapsto a, 1 \mapsto b\}$, and let $\rho_n(j) = \rho(j_{n-1})\rho(j_{n-2}) \cdots \rho(j_0)$ denote the usual n-bit binary representation of j in positional notation – with leading zeros if needed. Define the family of functions σ_n by letting $\sigma_n(j) = \rho_n(j)\rho_n(j)^R$. Whenever n is understood from the context, we shall drop the subscript and write $\rho(j)$ instead of $\rho_n(j)$, and similarly for σ. Observe, that, among the palindromes of length $2n$, the word $\sigma(j)$ is the $(j+1)$th palindrome in lexicographic order. Conversely, for a palindrome w of length $2n$, the preimage $\sigma^{-1}(w)$ equals the (zero-based) lexicographic index of w among the palindromes of length $2n$. For convenience, we extend the definition of σ_n to the case $n = 0$ by letting $\sigma_0(0) = \epsilon$.

We claim that, given $n \geq 1$ and k with $1 \leq k \leq 2^n$, as well as a nonnegative integer $j < k$, the word $\sigma(j)$ is in $L(E_{n,k})$. This claim will be proven by induction on n. The base case is $n = 0$. We thus have $k = 1$. Here, $\sigma_0(0) = \epsilon$, and $E_{0,1} = \epsilon$. For the induction step, we now assume $n \geq 1$. We consider two cases:

Case 1. Consider first the case $j < 2^{n-1}$. Then $\sigma_n(j) = a \cdot \sigma_{n-1}(j) \cdot a$. By the induction hypothesis, $\sigma_{n-1}(j) \in L(E_{n-1,k})$, and by the recursive definition of the regular expression $E_{n,k}$, we have $a \cdot L(E_{n-1,k}) \cdot a \subseteq L(E_{n,k})$. Hence, $\sigma(j) \in L(E_{n,k})$ in this case.

Case 2. The other case is $j \geq 2^{n-1}$. Then $\sigma_n(j) = b \cdot \sigma_{n-1}(j - 2^{n-1}) \cdot b$. Observe, that also $k > 2^{n-1}$ holds, since $j < k$. Let $k' = k - 2^{n-1}$ and $j' = j - 2^{n-1}$. Then $k' \geq 1$ and $j' \geq 0$, as well as $n - 1 \geq 0$. Using the induction hypothesis, we have $\sigma_{n-1}(j') \in L(E_{n-1,k'})$. In other words, $\sigma_{n-1}(j - 2^{n-1}) \in L(E_{n-1,k-2^{n-1}})$. Now, by the recursive definition of the regular expression $E_{n,k}$, we obtain $b \cdot L(E_{n-1,k-2^{n-1}}) \cdot b \subseteq L(E_{n,k})$. Hence, $\sigma(j) \in L(E_{n,k})$ also in this case.

This completes the induction, and the claim is established.

It remains to show that no other words are described by $E_{n,k}$. To this end, we note first that the recursive definition of $E_{n,k}$ ensures that it describes no non-palindromic words, and only words of length $2n$. Now let w be any word that is described by $E_{n,k}$. Recall that $\sigma^{-1}(w)$ is equal to the lexicographic index of w among all palindromes of length $2n$.

We shall prove by induction on n that the lexicographic index of every w described by $E_{n,k}$ is at most $k - 1$. In the base case $n = 0$, we must have $k = 1$ and $w = \epsilon$, and $\sigma^{-1}(w) = 0 = k - 1$ in this case. Now assume $n \geq 1$. We distinguish two cases:

Case 1. Consider first the case that the word w is the form axa. We need to consider two subcases. The first subcase is $k \leq 2^{n-1}$. Here, by definition of $E_{n,k}$, the word x is in $L(E_{n-1,k})$. Bearing in mind that $\sigma^{-1}(x) = \sigma_{n-1}^{-1}(x)$ and $\sigma^{-1}(w) = \sigma_n^{-1}(w)$ denote two different functions, we will again drop the subscripts for convenient reading. By the induction assumption, $\sigma^{-1}(x) \leq k - 1$. Recalling the bijection between natural numbers and palindromes, we have $\sigma^{-1}(axa) = \sigma^{-1}(x)$. With $axa = w$, we obtain $\sigma^{-1}(w) = \sigma^{-1}(x)$ in this subcase. The second subcase is $k > 2^{n-1}$. Here, by definition of $E_{n,k}$, the word x is in $L(E_{n-1,2^{n-1}})$. By the induction assumption, $\sigma^{-1}(x) \leq 2^{n-1} \leq k - 1$, and using again the bijection between natural numbers and palindromes, $\sigma^{-1}(w) = \sigma^{-1}(x)$.

Case 2. Now consider the case that the word w is of the form bxb. By the recursive definition of $E_{n,k}$, we can conclude that $x \in L(E_{n-1,k-2^{n-1}})$ – and that $k > 2^{n-1}$. By the induction assumption, $\sigma^{-1}(x) \leq k - 2^{n-1} - 1$. Recalling the bijection between natural numbers and palindromes, we have $\sigma^{-1}(bxb) = 2^{n-1} + \sigma^{-1}(x)$. Taking these two facts together, we obtain $\sigma^{-1}(w) = 2^{n-1} + \sigma^{-1}(x) \leq 2^{n-1} + k - 2^{n-1} - 1 \leq k - 1$, as desired.

This completes the proof of the second claim, and the proof of the lemma is completed. ◀

We thus can summarize our findings about palindromes of even length in the last three lemmata in the following statement.

▶ **Theorem 14.** *Let k and n be integers, with $n \geq 0$ and $1 \leq k \leq 2^n$. Then the set of the lexicographically first k palindromes of length $2n$ over a binary alphabet requires regular expressions of alphabetic width exactly $2n + 4(k-1) - 2S_2(k-1)$.*

6 Alphabetic Width of Palindromes of Odd Length

We turn to palindromes of odd length. The recurrences essentially differ only in the terminating cases. But changing the starting conditions of a nonlinear system may, or may not, change everything. We thus provide a careful writeup.

To this end, for positive integer n, let $\widetilde{P}_n$ denote the set of palindromes of length $2n-1$ over a binary alphabet. Now define $\widetilde{\ell}(n,k)$ to be the minimum alphabetic width of a regular expression describing a subset of $\widetilde{P}_n$, where the subset has cardinality at least k. Again by definition, $\widetilde{\ell}(n,k)$ is monotone with respect to the parameter k.

▶ **Lemma 15.** *Let $n \geq 1$ and $1 \leq k \leq 2^n$. Then $\widetilde{\ell}(n,k)$ obeys the following recurrence:*

$$\widetilde{\ell}(n,k) \geq \min\{\, \widetilde{\ell}(n-1,k) + 2, \min_{1 \leq i < k} \{\widetilde{\ell}(n,i) + \widetilde{\ell}(n,k-i)\}\,\}, \text{ for } n \geq 2 \text{ and } 2 \leq k \leq 2^{n-1},$$

$$\widetilde{\ell}(n,k) \geq \min_{1 \leq i < k} \{\widetilde{\ell}(n,i) + \widetilde{\ell}(n,k-i)\}, \text{ for } n \geq 1 \text{ and } k > 2^{n-1},$$

and

$$\widetilde{\ell}(n,1) = 2n-1.$$

In analogy to the definition of the function f, let $\widetilde{f}$ denote the integer-valued function which is defined by that recurrence, that is,

$$\widetilde{f}(n,k) = \min\{\, \widetilde{f}(n-1,k) + 2, \min_{1 \leq i < k} \{\widetilde{f}(n,i) + \widetilde{f}(n,k-i)\}\,\}, \text{ for } n \geq 2 \text{ and } 2 \leq k \leq 2^{n-1},$$

$$\widetilde{f}(n,k) = \min_{1 \leq i < k} \{\widetilde{f}(n,i) + \widetilde{f}(n,k-i)\}, \text{ for } n \geq 1 \text{ and } 2^{n-1} < k \leq 2^n,$$

and

$$\widetilde{f}(n,1) = 2n-1.$$

We estimate the values of the function $\widetilde{f}(n,k)$ as follows:

▶ **Lemma 16.** *Let $n \geq 1$ and $1 \leq k \leq 2^n$. Then $\widetilde{f}(n,k) = f(n,k) - k$.*

Thus, we immediately obtain:

▶ **Lemma 17.** $\widetilde{f}(n,2^n) = 3 \cdot 2^n - 4$.

With the lower bound in place, it remains to give an optimal regular expression matching the lower bound. The expression $\widetilde{E}_{n,k}$ is defined recursively as follows.

$$\widetilde{E}_{n,k} = a \cdot \widetilde{E}_{n-1,k} \cdot a, \text{ for } n \geq 2 \text{ and } 1 \leq k \leq 2^{n-1},$$

$$\widetilde{E}_{n,k} = a \cdot \widetilde{E}_{n-1,2^{n-1}} \cdot a + b \cdot \widetilde{E}_{n-1,k-2^{n-1}} \cdot b, \text{ for } n \geq 2 \text{ and } 2^{n-1} < k \leq 2^n,$$

and

$$\widetilde{E}_{1,1} = a \text{ as well as } \widetilde{E}_{1,2} = a + b.$$

The semantic correctness proof runs along the lines of the proof of Lemma 13.

▶ **Lemma 18.** *Let n, k be integers with $n \geq 1$ and $1 \leq k \leq 2^n$. Then the regular expression $\widetilde{E}_{n,k}$ describes the lexicographically first k palindromes of length $2n-1$.*

It remains to show that the alphabetic width of $\widetilde{E}_{n,k}$ meets the lower bound. An easy induction reduces this to the case of even length palindromes, in a similar vein as we did it in Lemma 16.

▶ **Lemma 19.** *Let n, k be integers with $n \geq 1$ and $1 \leq k \leq 2^n$. Then $\mathsf{awidth}(\widetilde{E}_{n,k}) = \mathsf{awidth}(E_{n,k}) - k$.*

We thus can summarize our findings about palindromes of odd length in the following theorem – compare with Theorem 14.

▶ **Theorem 20.** *Let k and n be integers, with $n \geq 1$ and $1 \leq k \leq 2^n$. Then the set of the lexicographically first k palindromes of length $2n-1$ over a binary alphabet requires regular expressions of alphabetic width exactly $2n + 3(k-1) - 2S_2(k-1) - 1$.*

We conclude this section with a curious observation, which was contributed by an anonymous reviewer. Recall that

$$\widetilde{\ell}(n,k) = \min_{\substack{|L|\geq k \\ L\subseteq \widetilde{P}_n}} \{\mathsf{awidth}(L)\},$$

that is, $\widetilde{\ell}(n,k)$ denotes the minimum alphabetic width of a regular expression describing a subset of $\widetilde{P}_n$, where the subset has cardinality at least k. Then the analysis in the present work establishes that the minimum is attained by the set of the lexicographically first k palindromes, and a corresponding statement holds in the even length case. This observation is summarized in the following theorem (which no longer needs to distinguish between even and odd length):

▶ **Theorem 21.** *For $n \geq 0$ and $1 \leq k \leq 2^{\lceil n/2 \rceil}$, let Pal_n denote the set of palindromes of length n, and let $\mathrm{Lex}_{n,k}$ denote the set of the lexicographically first k palindromes of length n. Then*

$$\mathrm{Lex}_{n,k} \in \operatorname*{argmin}_{\substack{|L|\geq k \\ L\subseteq \mathrm{Pal}_n}} \mathsf{awidth}(L).$$

As the reviewer pointed out, this is reminiscent of the Kruskal-Katona Theorem from extremal combinatorics, see, e.g., [19]. Among several equivalent formulations of that theorem, one of them deals with minimization of the size of shadows in layers of the Boolean hypercube. The Kruskal-Katona Theorem then states that initial segments with respect to a version of the lexicographic ordering form sets with the smallest shadow possible.

7 Conclusion

Most lower bound proofs for regular expression size can be put into the following three categories: proofs based on (arithmetic) circuit complexity, e.g., [4, 7, 14], proofs based on the star height lemma, e.g., [9, 12, 13], and specialized proofs that are tailor-made for a specific language family, e.g., [2, 6, 10, 23]. While the present work falls into the third category, the lower bound method is quite similar to the one for permutations [23]. We expect that the method can be expanded to further families of finite languages, where the best known regular expressions have a divide-and-conquer flavor. A few examples from the literature come to mind:

- First, the binomial language $B_{n,k} = \{ w \in \{0,1\}^n : |w|_1 = k \}$. A regular expression of divide-and-conquer flavor having size $n^{O(\log k)}$ for this language was proposed in [7], and the question of optimality was posed as an open problem. In [4], methods from arithmetic circuit complexity are utilized to derive a lower bound of $nk^{\Omega(\log k)}$.
- Regarding larger alphabets, the less-than relation on an n-set is given as $\{ ij \mid 1 \leq i < j \leq n \}$. For this language, the minimum required regular expression size was determined exactly in [2], which implies a lower bound on the complexity of rectifier networks. This language naturally generalizes to the set of increasing sequences of length k over an n-set. For this an arithmetic formula lower bound was derived in [17]. As pointed out in [4], that result transfers to lower bounds on regular expression size.
- For the set of permutations of an n-set, the exact bound was determined in [23], and an asymptotic lower bound is given in [4] using a different method. Its natural generalization is the set of k-permutations of an n-set. The nondeterministic state complexity of this language is studied in [1]. Their motivation is that a lower bound on nondeterministic state complexity gives lower bounds on the running time for parameterized algorithms following the divide-and-conquer paradigm. We claim that, by the well-nested nature of divide-and-conquer, a (potentially higher) lower bound on regular expression size would serve this goal equally well.

The cited examples witness a lot of cross-fertilization between lower bound methods on various models of computation, including arithmetic circuits, rectifier networks, families of parameterized algorithms, and, of course, regular expressions.

References

1 R. Ben-Basat, A. Gabizon, and M. Zehavi. The k-distinct language: Parameterized automata constructions. *Theoretical Computer Science*, 622:1–15, 2016.

2 D. Chistikov, Sz. Iván, A. Lubiw, and J. Shallit. Fractional coverings, greedy coverings, and rectifier networks. In Heribert Vollmer and Brigitte Vallée, editors, *Proceedings of the 34th Symposium on Theoretical Aspects of Computer Science*, volume 66 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 23:1–23:14. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017.

3 N. Chomsky. Three models for the description of language. *IRE Transactions on Information Theory*, 2(3):113–124, 1956.

4 E. Cseresnyes and H. Seiwert. Regular expression length via arithmetic formula complexity. In G. Jirásková and G. Pighizzini, editors, *22nd International Conference on Descriptional Complexity of Formal Systems*, volume 12442 of *LNCS*, pages 26–38. Springer, 2020.

5 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.

6 A. Ehrenfeucht and H. P. Zeiger. Complexity measures for regular expressions. *Journal of Computer and System Sciences*, 12(2):134–146, April 1976. `doi:10.1016/S0022-0000(76)80034-7`.

7 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.

8 Y. Gao, N. Moreira, R. Reis, and S. Yu. A survey on operational state complexity. *Journal of Automata, Languages and Combinatorics*, 21(4):251–310, 2016.

9 W. Gelade. Succintness of regular expressions with interleaving, intersection, and counting. *Theoretical Computer Science*, 411(31–33):2987–2998, 2010. `doi:10.1016/j.tcs.2010.04.036`.

10 W. Gelade and F. Neven. Succinctness of the complement and intersection of regular expressions. *ACM Transactions on Computational Logic*, 13(1):No. 4, January 2012. `doi:10.1145/2071368.2071372`.

11 D. H. Greene and D. E. Knuth. *Mathematics for the Analysis of Algorithms*. Progress in Computer Science. Birkhäuser, 2nd edition, 1982.

12 H. Gruber and M. Holzer. Finite automata, digraph connectivity, and regular expression size. In L. Aceto, I. Damgaard, L. A. Goldberg, M. M. Halldórsson, A. Ingólfsdóttir, and I. Walkuwiewicz, editors, *Proceedings of the 35th International Colloquium on Automata, Languages and Propgramming*, number 5126 in LNCS, pages 39–50, Reykjavik, Iceland, July 2008. Springer. `doi:10.1007/978-3-540-70583-3_4`.

13 H. Gruber and M. Holzer. Tight bounds on the descriptional complexity of regular expressions. In V. Diekert and D. Nowotka, editors, *Proceedings of the 13th International Conference Developments in Language Theory*, number 5583 in LNCS, pages 276–287, Stuttgart, Germany, June–July 2009. Springer. `doi:10.1007/978-3-642-02737-6_22`.

14 H. Gruber and J. Johannsen. Tight bounds on the descriptional complexity of regular expressions. In R. Amadio, editor, *Proceedings of the 11th Conference Foundations of Software Science and Computational Structures*, number 4962 in LNCS, pages 273–286, Budapest, Hungary, March–April 2008. Springer.

15 M. Holzer and M. Kutrib. Descriptional and computational complexity of finite automata—a survey. *Information and Computation*, 209(3):456–470, March 2011. `doi:10.1016/j.ic.2010.11.013`.

16 J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, 1979.

17 P. Hrubes and A. Yehudayoff. Homogeneous formulas and symmetric polynomials. *Computational Complexity*, 20(3):559–578, 2011.

18 H.-K. Hwang and T.-H. Tsai. An asymptotic theory for recurrence relations based on minimization and maximization. *Theoretical Computer Science*, 290(3):1475–1501, 2003.

19 S. Jukna. *Extremal Combinatorics: With Applications in Computer Science*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2nd edition, 2011.

20 D. E. Knuth. *Seminumerical Algorithms*, volume 2 of *The Art of Computer Programming*. Addison-Wesley, 3rd edition, 1998.

21 D. E. Knuth. *Sorting and Searching*, volume 3 of *The Art of Computer Programming*. Addison-Wesley, 3rd edition, 1998.

22 A.-M. Legendre. *Essai sur la théorie des nombres*. Courcier, 2ème edition, 1808.

23 A. M. Lovett and J. O. Shallit. Optimal regular expressions for permutations. In Ch. Baier, I. Chatzigiannakis, P. Flocchini, and S. Leonardi, editors, *Proceedings of the 46th International Colloquium on Automata, Languages, and Programming*, volume 132 of *LIPIcs*, pages 121:1–121:12, Patras, Greece, July 2019. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.