Communication Complexity and Regular Expression Size*

Hermann Gruber[1] Jan Johannsen[2]

[1] Institut für Informatik, Justus-Liebig-Universität Gießen,
Arndtstraße 2, D-35392 Gießen, Germany
email: gruber@informatik.uni-giessen.de

[2] Institut für Informatik, Ludwig-Maximilians-Universität München
Oettingenstr. 67, 80538 München, Germany
email: jan.johannsen@informatik.uni-muenchen.de

1 Introduction

We consider the problem of converting a deterministic finite automaton (DFA) into a short regular expression (RE). Examples given by Ehrenfeucht and Zeiger in the 1970s show that the required expression size in the worst case is $2^{\Theta(n)}$ for infinite languages, and for finite languages in $n^{\Omega(\log \log n)}$ and $n^{O(\log n)}$, if the alphabet size is allowed to grow with the number of states n of the given automaton [2]. We develop a new lower bound method for regular expression size, based on communication complexity, to show that in the second case, the required size is indeed $n^{\Theta(\log n)}$, thus solving an old open problem stated in that work. Overmore, our witness languages are over a binary alphabet. For the case of infinite languages, exponential lower bounds for small alphabets have been obtained only very recently in [4, 6].

This is an abstract of the conference version [9], in which more details can be found.

*Most of the work was done while the first author was at Institut für Informatik, Ludwig-Maximilians-Universität München.

2 Preliminaries

We assume the reader to be familiar with the basic notions in formal language and automata theory as contained in [10]. In addition, we need the following notions: The *alphabetic width* (or *size*) of a regular expression E is defined as the total number of occurrences of symbols in Σ in E, and is denoted by $alph(E)$. In a similar way, for a regular language L we define $alph(L)$ as the minimum alphabetic width among all regular expressions describing L. We call a finite language L *homogeneous* if all words in L have the same length.

We will also need some notions from communication complexity theory. For a thorough treatment of that topic, the reader might want to consult [11]. Let X, Y, Z be finite sets and $R \subseteq X \times Y \times Z$ a ternary relation on them. In the search problem R, we have Alice given some input $x \in X$, Bob is given some input $y \in Y$. Initially, no party knows the other's input, and Alice and Bob both want to output some z such that (x, y, z) by communicating as few bits as possible. A *communication protocol* is a binary tree with each internal node v labeled either by a function $a_v : X \to \{0, 1\}$ if Alice transmits at this node, or $b_v : Y \to \{0, 1\}$ if Bob transmits at this node. Each leaf is labeled by an output $z \in Z$. We say that a protocol solves the search problem for relation R if for every input pair $(x, y) \in X \times Y$, walking down the tree according to the functions a_v and b_v leads to a leaf labeled with some admissible z, which satisfies $(x, y, z) \in R$. The *protocol partition number* $C^P(R)$ denotes the minimum number of leaves among all protocols solving the search problem for R. For using this notion in formal language theory, let $\Sigma = a_1, \ldots, a_k$ be an ordered alphabet. The order on Σ is extended componentwise to a partial order on Σ^n. A homogeneous language $L \subseteq \Sigma^n$ is called monotonic, if

$$v \in L \text{ and } w \geq v \text{ implies } w \in L \; .$$

For a homogeneous language $\emptyset \subset L \subset \Sigma^n$, the search problem $R_L \subseteq L \times (\Sigma^n \setminus L) \times [n]$ is defined by $(v, w, i) \in R_L$ iff $v_i \neq w_i$. If L is monotonic, then additionally the monotonic search problem R_L^m is defined by $(v, w, i) \in R_L^m$ iff $v_i > w_i$.

3 A new Lower Bound Technique for Regular Expression Size

We outline next how techniques from communication complexity can be used for proving lower bounds on the size of regular expressions for homogeneous languages.

Lemma 3.1 *For every homogeneous language L with $\emptyset \subset L \subset \Sigma^n$, $alph(L) \geq C^P(R_L)$. Moreover, if L is monotonic, then $alph(L) \geq C^P(R_L^m)$.*

The first part of the lemma was proved for the special case of the parity function in [1]—in terms of Boolean formula size instead of $C^P(R_L)$, but this is equivalent to the setup used here, see [11, Chapter 5]. We note that only the second part allows us to prove a superpolynomial lower bound on the conversion problem. To establish this bound, for a given pair of integers (ℓ, n), we define a family of graphs $\mathcal{G}_{\ell,n}$ as the set of directed graphs whose vertex set V is organized in $\ell + 2$ layers, with n vertices in each each layer. Hence we assume $V = \{ \langle i, j \rangle \mid 1 \leq i \leq n, 0 \leq j \leq \ell + 1\}$. For all graphs in $\mathcal{G}_{\ell,n}$, we require in addition that each edge connects a vertex in some layer i to a vertex in the adjacent layer $i + 1$. The following definition serves to represent subsets of $\mathcal{G}_{\ell,n}$ as homogeneous languages over the alphabet $\{0, 1\}$: Fix a graph $G \in \mathcal{G}_{\ell,n}$ for the moment. Let $e(i, j, k) = 1$ if G has an edge from vertex i in layer j to vertex k in layer $j + 1$, and let $e(i, j, k) = 0$ otherwise. Next, for vertex i in layer j, the word $f(i, j) = e(i, j, 1)e(i, j, 2) \cdots e(i, j, n)$ encodes the set of outgoing edges for this vertex. Then for layer j, the word $g(j) = f(1, j)f(2, j) \cdots f(n, j)$ encodes the set of edges connecting vertices in layer j to vertices in layer $j + 1$, for $0 \leq j \leq \ell$. Finally, the graph G is encoded by the word $w(G) = g(0)g(1) \cdots g(\ell)$. It is easy to see that each word in the set $\{0, 1\}^{n^2(\ell+1)}$ can be uniquely decoded as a graph in the set $\mathcal{G}_{\ell,n}$. Without risk of confusion, we will henceforth not distinguish between graphs and their encodings, and between sets of graphs and the corresponding languages. A graph $G \in \mathcal{G}_{\ell,n}$ belongs to the subfamily $\mathsf{fork}_{\ell,n}$, if there exists a simple path starting in $\langle 1, 1\rangle$ ending eventually in a fork, that is, a node with outdegree at least two. We prove that the language $\mathsf{fork}_{\ell,n}$ admits a small DFA but needs a large RE. More precisely, we show that $L_k = \mathsf{fork}_{\ell,n}$, for some $\ell \in \Theta(n^3)$, can be accepted by a DFA with at most $k = n^6$ states, but a large lower bound on the minimum expression size is obtained from using Lemma 3.1 by a reduction—in the communication complexity sense— from the FORK relation defined in [5].

Theorem 3.2 *There exist an infinite family of finite languages L_k over a*

binary alphabet such that L_k is acceptable by a DFA with at most k states, but every equivalent regular expression has alphabetic width at least

$$k^{(1/144-o(1))\log k}.$$

4 Conclusions and Further Research

In this work, we established an asymptotically tight lower bound for the problem of converting a finite automaton accepting a finite language over binary alphabet into a regular expression. As mentioned in the introduction, the case of infinite languages was also settled recently in [4, 6]. These and follow-up works [3, 7, 8] also study the effect of common language operations, such as intersection or complement, on regular expression size. A topic for further research would be a corresponding study for the case of finite languages, thus paralleling the developments in state complexity (see e.g. [12]).

References

[1] K. Ellul, B. Krawetz, J. Shallit and M. Wang. Regular Expressions: New Results and Open Problems. *Journal of Automata, Languages and Combinatorics*, 10(4):407–437, 2005.

[2] A. Ehrenfeucht and H. P. Zeiger. Complexity Measures for Regular Expressions. *Journal of Computer and System Sciences*, 12(2):134–146, 1976.

[3] W. Gelade. Succinctness of Regular Expressions with Interleaving, Intersection and Counting. In: *Proceedings of MFCS*: 363–374, LNCS 5162, Springer, 2008.

[4] W. Gelade and F. Neven. Succinctness of the Complement and Intersection of Regular Expressions. In: *Proceedings of STACS*: 325-336, Dagstuhl Seminar Proceedings 08001, IBFI Schloss Dagstuhl, 2008.

[5] M. Grigni and M. Sipser. Monotone Separation of Logarithmic Space from Logarithmic Depth. *Journal of Computer and System Sciences*, 50(3):433–437, 1995.

[6] H. Gruber and M. Holzer. Finite Automata, Digraph Connectivity, and Regular Expression Size. In: *Proceedings of ICALP*: 39–50, LNCS 5126, Springer, 2008.

[7] H. Gruber and M. Holzer. Provably Shorter Regular Expressions from Deterministic Finite Automata (Extended Abstract). In: *Proceedings of DLT*: 383–395, LNCS 5257, Springer, 2008.

[8] H. Gruber and M. Holzer. Language Operations with Regular Expressions of Polynomial Size. In: *Proceedings of DCFS*: 182–193, CSIT University of Prince Edward Island, 2008.

[9] H. Gruber and J. Johannsen. Optimal Lower bounds on Regular Expression Size using Communication Complexity. In: *Proceedings of FoSSaCS*: 273–286, LNCS 4962, Springer, 2008. Full version in preparation.

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

[11] E. Kushilevitz and N. Nisan. *Communication Complexity.* Cambridge University Press, 1997.

[12] S. Yu. State Complexity of Finite and Infinite Regular Languages. *Bulletin of the EATCS*, 76: 142–152, 2002.