On the Descriptional and Algorithmic Complexity of Regular Languages

Hermann Gruber

HARLAND
media

Bibliografische Information Der Deutschen Bibliothek
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.ddb.de abrufbar.

Gießener Dissertation
Fachbereich Mathematik und Informatik, Physik, Geographie
Justus-Liebig-Universität Gießen (D26) 2009

1. Gutachter: Prof. Dr. rer. nat. Markus Holzer
2. Gutachter: Prof. Dr. rer. nat. Martin Kutrib

Termin der Disputation: 25.09.2009

HARLAND media, Lichtenberg (Odw.) 2010
www.harland-media.de

Gedruckt auf alterungsbeständigem Papier nach ISO 9706 (säure-, holz- und chlorfrei).

Printed in Germany

ISBN 978-3-938363-62-1

Digital ist besser
Tocotronic

Preface

This thesis is a research monograph. As such, it targets at an audience of experts, primarily in the fields of foundations of computer science and discrete mathematics. Nevertheless, already several persons showed interest in understanding at least the main theme of this thesis, but have only little background in mathematics and computer science. This preface can not serve a bluffer's guide to the main body of work, but we try at best to explain at least the central abstractions at an informal level.

The present work deals with mathematical models of computational processes. Several such models exist, each with its own advantages and characteristics. We will concentrate on the simplest of these models, namely on finite automata. An example drawn from everyday life naturally modeled as finite automaton is a vending machine. The observable behavior of a such a vending machine is described as sequence of atomic events. For simplicity, let us assume the possible events are: a coin worth one or two units of money is inserted, in symbols ❶ or ❷, a cup of coffee ∪ is requested and brewed, and alternatively the cancel button © can be hit. Thereafter some coins might be returned, actions for which we introduce the symbols ① or ②. Now the behavior of a correctly operating vending machine is described as a set of sequences made up from these symbols. We expect that most users of the machine will be content with the sequence

❶❷①∪❶❶©②❷∪,

except possibly for the price of two money units for a cup of coffee. In contrast, the sequence

❷∪❶❶©©©①©©©©©

is certainly *not* expected to be observed on a machine that operates correctly. We thus model the observable behavior of a vending machine as a set of sequences over some finite alphabet—in our example, this set will contain infinitely many correct sequences, but it will not include all possible sequences. The internal behavior can be modeled as a control unit, which is at each point in time in one of finitely many possible internal states. At this point, we mention that our vending machine always immediately returns overpaid amounts. In this way, we can retain a finite number of states. Otherwise it would be possible to insert arbitrarily large amounts of money before hitting the © button. In order to guarantee that it always returned the correct amount of money, such a machine would require a fairly large memory unit in that case.

A mathematically precise model of a finite state control unit, such as the one found inside the vending machine, is the concept of a *finite automaton*. The set of sequences that form the behavior of a finite automaton is then a *regular language*.[1] This model

[1] In everyday life, we conceive the term *language* in a much narrower sense. In computer science, the term is very generously defined: Any set of symbol sequences makes up a (formal) language. There is not too much of interest to say about such a general concept. Thus we study certain interesting families of formal languages, such as the family of regular languages.

draws a clear distinction between the internal realization of the vending machine—the finite automaton—and its externally observable behavior—a regular language.

An advantage of these notions is that once a prototype is developed, the designer may eventually want to replace the internal circuitry by an easer or cheaper one. The users will be content as long as the behavior of the new machine is the same as that of the old one. One part of this thesis is devoted to the question to what extent such simplifications can be automatically computed, while consuming only a reasonable amount of memory and computation time.

The above scenario requires that the desired behavior of the vending machine is specified in the form of a prototype. There is also a more convenient way to specify the desired behavior. The behavior can be described by so-called *regular expressions*. These are a kind of formula that bear some superficial similarity to arithmetical expressions, or the formulas from mathematical logic. For mathematically trained persons, such formulas are often much easier to understand than the complex wiring diagrams of finite automata. The larger part of this thesis deals with questions regarding regular expressions. One such question is the following: Assume we already have a vending machine, but the formula specifying its behavior got lost. The problem is now to reverse-engineer a formula, in the form of a regular expression, from the wiring diagram of the machine. In order to remain understandable, the description of the behavior should of course be as short as possible. From a bird-eye's perspective, this is similar to the process of translating (say) Latin to English. But instead of a description in Latin, we have a formal description in the form of a wiring diagram, and instead of translating it to English for easier understanding, we want to translate it into a regular expression. In both translation tasks, we will encounter sentences and constructs for which there is no direct analogue in the target language. Then we have to think about how to paraphrase these constructs, of course as succinctly as possible. This thesis aims to provide a deeper understanding about how smoothly this task can be accomplished.

We will also seize the strength and limitations of regular expressions as a specification formalism. In the role of a requirements engineer, we often want to combine smaller fragments to specify more complex requirements. Astonishingly, we will find in this thesis that several very simple mechanisms of assembling more complex units cannot be easily described in the language of regular expressions—rather often, such combinations of requirements have to be circumscribed in an extremely cumbersome way. In these cases, practitioners may prefer to use more elaborate formalisms that are more succinct than regular expressions.

We hope that also non-expert readers could catch a glimpse of the material treated in this thesis, and the type of questions addressed. For the expert audience, we hope that this short distraction served as a little *canapé* that wetted their appetite for the technical developments to come in the present thesis.

Collaborations

Some parts of this thesis arose from collaborative work, and we want to acknowledge these contributions. Contributions resulting from joint work with Markus Holzer were presented, partly in preliminary form, at the 10th, 11th, 12th and 13th *International Conference on Developments in Language Theory* [74, 77, 80, 83], at the 8th and 10th installments of the *International Workshop on Descriptional Complexity of Formal Systems* [75, 81], at the 1st *International Conference on Language and Automata Theory and*

Applications [76], and at the 35th *International Colloquium on Automata, Languages and Programming* [79]. The part concerning the conversion of finite automata accepting finite languages into regular expressions is based on joint work with Jan Johannsen, and was presented at the 11th *International Conference on Foundations of Software Science and Computation Structures* [84]. Two of these contributions already appeared, in revised and expanded form, as journal articles [78, 82].

Acknowledgments

I would like to thank all persons who directly or indirectly contributed to this work. Any attempt of listing all contributions will be necessarily incomplete. I hope that at least the most immediate contributors are mentioned. First of all, I would like to thank my advisor, Markus Holzer, not only for his continuous support and advice, but also for his great care at reading a draft of this thesis. In this context, I would also like to thank Irmgard Kellerer, Martina Mensch and Renate Szweda for reading portions of a preliminary draft. I am indebted to Jan Johannsen for sharing his expertise in communication complexity with me, and for becoming enthusiastic about topics outside his main area of research. Thanks also goes to Jeffrey Shallit and Wouter Gelade for sending me preprints of their papers at times these did not appear yet in print. Finally, I thank all of my current and former colleagues for countless inspiring discussions.

Contents

List of Figures

List of Tables

Part I

Getting Started

1 Introduction

Possibly the simplest mathematical model capturing the nature of computation is that of a finite automaton. Admittedly, virtually every computer scientist has taken some courses in which she or he learned about this concept. We start with a short historical survey of the development of the theory of finite automata. Thereafter, we present the main questions addressed and highlight a few of the obtained results.

In the 1940s and 1950s there were many different lines of research that eventually ended up in the same concept, namely that of finite automata. We begin with a very short historical survey of these developments.

Perhaps surprisingly, finite state systems were historically first studied by biologists, rather than by computer scientists or electrical engineers. In the 1940s, McCulloch and Pitts suggested a precursor of the concept of finite automata as a mathematical model of neural activity in nerve nets [139]. A few years later, Kleene [125] introduced a textual specification formalism, the concept of regular expressions. He proved that the sets of sequences that can be described by regular expressions are exactly those that can be described by finite automata. Since that time, this family of languages is known as the regular languages. The definition of (deterministic) finite automata as we know it today was shaped only shortly before by Huffman [105, 106] and Moore [145], whose motivation for studying finite automata was to formalize the behavior of switching circuits. After Kleene's invention of regular expressions, Rabin and Scott [157] introduced another method for describing regular languages, the nondeterministic finite automata. This was such an outstanding conceptual contribution that they later were given the Turing award for this work, the most prestigious prize in computer science [7].

Other lines of research that flowed into the early development of the theory of finite automata came from mathematics, in particular logic and algebra. The dream of having a logical calculus for mechanically proving or disproving mathematical statements dates back to the ideas of Leibniz in the 17th century, and was later formulated precisely by Hilbert [41]. After Gödel had given a negative answer to Hilbert's famous *Entscheidungsproblem* in the 1930s, many mathematicians sought for fragments of arithmetic that were powerful enough to be useful yet weak enough to retain the desirable feature of a decidable theory. Here, Büchi [24], Elgot [57], and Trakhtenbrot [174] identified a decidable fragment of second order logic. As it turns out, this logical theory captures exactly the regular languages. This means that each formula of that theory can be converted into an equivalent finite automaton and *vice versa*. Beside this logical characterization, an algebraic viewpoint proved useful in charting the fine structure of regular languages. It known that Kleene's theorem also allows for an algebraic interpretation in the theory of semigroups, see e.g. [155]. An early success of this approach was Schützenberger's Theorem, which links a basic family of algebras with a basic logical theory: He proved that the aperiodic monoids are equally expressive as first-order logic with total order [166]. A detailed account on the algebraic approach to understanding aspects of regular languages is given in [56]. Last but not least, also in the 1950s, Chomsky was seeking for mathematical models capturing the relevant features of natural language. This led to the definition of the Chomsky hierarchy [31, 32], with the family of regular languages at its lowest level. Notably, although the Chomsky hierarchy was motivated by linguistics, it turned out to be more influential in the field of computer science, cf. [133]. An extensive historical survey about these early developments is given by Mahoney [133].

Fueled by these early successes, the theory of automata saw a golden age in the 1960s and 1970s, and was considered at the time as one of the main interests in theoretical computer science research, cf. [104, 179]. Inside this theory, finite automata became the best known models, cf. [22]. Along with the gained maturity, there had been a significant decrease of interest in research on these topics during the late 1970s. A possible reason is that after several decades of intensive research in the field, it seemed that almost everything interesting about regular languages was known, and only the most difficult problems remained. In a survey appearing in 1980, Brzozowski, one of the pioneers in automata theory, exemplified this view by listing six problems on finite automata [22]. All of these were presumably very difficult, since many excellent researchers had tried to solve them, with little success. Meanwhile, not only that several of these problems have been solved [45, 91, 169], in recent years there appears to be a renewed interest in the theory of finite automata, cf. [180]. This observation is corroborated by many interesting new results on classical questions, see e.g. [13, 86, 101, 118, 162]. The appearance of several recent listings describing old and new open challenges regarding regular languages [30, 59, 104, 181] gives further evidence for such a new momentum.

An attractive feature of regular languages is that the various different characterizations can be effectively obtained, in the sense that one can, in principle, convert automatically between various modes of description e.g. between the abovementioned logical formulas and finite automata. In the 1950s, when most of these *effectiveness* results were first proved, it was unusual to determine the time and memory requirements of such algorithms. Later, along with the growing availability of electronic computers, questions of *efficiency* gained importance: at once, such algorithms proved to be powerful tools in solving real-world problems. Apart from the traditional application domains which marked the origins of regular languages, the latter concept has found widespread use in many areas of computing. Notable examples include traditional applications such as lexical analysis in compiler design [1] and pattern matching in text processing [127]. More recent additions to the list of applications are UML statecharts in software engineering [52], specifications and query languages in XML data and document processing [130, 135], and network intrusion detection in internet packet routing [128].

Part of such applications deals with massive datasets, and there is a growing interest in memory-efficient representations of regular languages, cf. [180]. Such questions were addressed already since the beginnings of automata theory, see e.g. [132, 157]. A systematic study of questions of *descriptional complexity* started in the beginning 1970s, as witnessed by early papers of Meyer and Fischer [143], and of Maslov [136]. Meyer and Fischer compared different models of description, such as different types of automata and regular expressions, while Maslov investigated the effect of language manipulations on the required number of states in deterministic finite automata. Around the same time, the first *computational complexity* results regarding representations of regular languages appeared. Meyer and Stockmeyer showed that increased succinctness of description can render some problems computationally intractable [144]. For deterministic finite automata, efficient minimization is possible, and algorithms solving that problem were developed since the beginning of automata theory [72, 95, 105, 106, 145, 173]. Since it was known that there are cases where the smallest deterministic finite automaton is exponentially larger than the smallest equivalent nondeterministic finite automaton [132], the obvious question was whether we can have similar algorithms for minimizing nondeterministic finite automata. Here the result of Stockmeyer and Meyer implies that minimization of nondeterministic finite automata is computationally hard.

This is of course bad news, and a part of this thesis is devoted to look for ways out. To illustrate the kind of question we are interested in, a part of this thesis is devoted to the computational complexity of the minimization problem of nondeterministic finite automata. Since the problem is hard in the general case, we study special cases on the one hand and approximate solutions on the other hand. Here we continue a line of research that has been studied by different research groups in the past [70, 112, 114, 144]. We announce the solution of several research challenges that were left open by previous investigators. We also investigate various other aspects of minimal nondeterministic finite automata. This includes the comparison of techniques for proving lower bounds, and of different ways to measure the size of a nondeterministic finite automaton. Examples of our results include the following: We show that counting the number of states is essentially different from counting the number of transitions. The minimum number of states can numerically largely differ from the number of transitions, and sometimes it is impossible to minimize both measures simultaneously. We also show that minimizing nondeterministic finite automata is rendered computationally less complex if a finite language is specified explicitly, as a list of words. This setup appears, for example, in computational linguistics [167]. Still, even this severely restricted problem remains computationally intractable (**NP**-hard).

Often we are already content with approximate solutions to such hard problems. In 1993, Jiang and Ravikumar [114] raised the question whether we can obtain an approximate solution in polynomial time, provided the input is specified in a not too succinct manner. In this direction, Gramlich and Schnitger [70] provided some evidence that even weak approximations are impossible to obtain efficiently. Yet there were some technical issues with the result, and these authors posed a few open questions. For instance, they relied on an unusual cryptographic assumption. Most of these issues are resolved by the study presented here. We strengthen their results in several directions: First, we give evidence for hardness of approximation based on the standard assumption, *i.e.*, $\mathbf{P} \neq \mathbf{NP}$. Second, we can show that the problem remains as hard to approximate in the abovementioned use case appearing in computational linguistics. And finally, in several cases we obtain quantitative improvements on the bounds of approximability.

While finite automata are ideally suited for manipulation by computers, regular expressions are the preferred choice to be understood and specified by humans, cf. [141]. This can be explained as follows: just like ordinary arithmetic expressions, regular expressions have a hierarchical structure, which is often more easily perceived than that of a finite automaton. Another advantage is that regular expressions are handy to write down as a formula, thus utilizing only one dimension. In contrast, for finite automata it is even a nontrivial task to find a layout in two dimensions [20]. Still, there are regular expressions that are more difficult to understand than others. A potential source of such perceived complexity is nested application of the Kleene star operator, cf. [141]. This observation led Eggan to the definition of the star height of regular languages in the beginning 1960s [54]. Indeed, already early investigations showed that limiting access to the star operator gives rise to a fine-grained hierarchy inside the regular languages [46, 54, 141]. More precisely, the languages of star height at most k form a strict subset of the languages of star height at most $k+1$, for every integer k. In the decade that followed Eggan's seminal paper, quite a few works investigating regular expressions appeared, see e.g. [35, 36, 46, 92, 140, 141]. All of these were devoted to the concept of star height. The large interest in this concept stemmed from the fact that it was at the time a famous open problem whether the star height of a regular language is computable. Later, in 1980, this question constituted one of Brzozowski's six hard problems mentioned above [22]. The problem was resolved only after 25 years by Hashiguchi [91], with a rather intricate algorithm, cf. [124]. As a structural complexity measure, Brzozowski [22] states that the star height is a rather direct measure of complexity for regular languages.

But the most obvious measure related to the regular expression model is minimum required regular expression size. For that reason, the present thesis is largely concerned with this measure. Properties relating to regular expression size were first subject to systematic study in a paper by Ehrenfeucht and Zeiger in the mid-1970s [55]. In contrast to Eggan's paper introducing star height, that paper did not trigger much follow-up work at the time. It can only be speculated about the reasons for that. One possible reason for this is that the paper appeared at a time of decreasing interest in automata theory. But despite the recently regained research interest in this topic, still very little was known about the descriptional complexity of regular expressions at the time the author started working on this thesis, cf. [59]. In hindsight, this might be due to the lack of suitable techniques for proving lower bounds on regular expression size. Part of this thesis is devoted to the development of tools for proving such lower bounds. At this point, we note that around the same time Gelade and Neven [65] came up with another, quite different, lower bound technique. They to provide answers that partly overlap with results presented in this thesis. One of our tools for establishing lower bounds is based on the insight that if a regular language has sufficiently complex internal structure, then it requires huge regular expressions. More technically, we prove an exponential relation between the star height (*structural* complexity) and minimum expression size (*descriptional* complexity) for the regular languages. In this way, we can harness the rich literature on star height of regular languages to prove descriptional complexity results. The proof techniques developed in this thesis have a wealth of consequences. First, they allow us to study the conversion of finite automata into regular expressions and variations thereof. Second, we can use them to highlight the dynamic aspects of regular expressions, such as the evolution of regular expression size under various language operations. Finally, we are able to answer quite a few open questions regarding regular expressions. This includes not only an open question raised in the 1970s by Ehrenfeucht and Zeiger but also several research challenges proposed more recently by Ellul et al. [59].

We mention in particular the following results: The original proof of Kleene's theorem readily implies that deterministic finite automata over binary alphabets can be converted into regular expressions of size $O(4^n)$. As observed by Ehrenfeucht and Zeiger [55] in the 1970s, we can do better if the given automaton accepts only finitely many words. There we get an upper bound of $n^{O(\log n)}$ on regular expression size. Regarding lower bounds, these authors could only show that size at least $n^{\Omega(\log \log n)}$ will be necessary in the worst case. Consequently, they posed the question of narrowing the gap between the upper and lower bound. We will present a definite answer to their question, as we are able to raise the lower bound to give a tight estimate of $n^{\Theta(\log n)}$ on required size in the worst case. Our new lower bound already applies for binary alphabets. For the general case of infinite languages, the classical bound of $O(4^n)$ remained the best known until present, cf. [59].

Almost 50 years after Kleene's initial discovery, we now devise an improved algorithm, which even attains a bound of $O(1.742^n)$. This is close to optimal, since we also provide a lower bound of c^n, for some constant $c > 1$. We also undertake first attempts at charting the borders of tractability in regular expression manipulation.

The present thesis is organized in three main parts; each of these ends with a summary of the respective technical results. These parts, which constitute the main body of the thesis, are surrounded by an introductory part and a final part. The introductory part will now continue with a short recapitulation of the basic definitions, and, at the very end, the final part discusses possible lines of further research.

Part V

Outro

16 Conclusion and Further Research

In this thesis, we considered questions regarding how we can deal efficiently with descriptions of regular languages, and where the inherent limitations of the respective mechanism lie. Although by now some parts of the landscape are charted much more completely than at the time this research work started, some questions had to remain unresolved. We shortly recapitulate our main findings and discuss some questions that had to be left open. Here we highlight only a few outstanding questions, in the hope that these are both of interest and also specific enough to stimulate further research in these directions in the near future. A few more questions are found in the respective summaries of each of the three main parts of the thesis.

The first main theme of the thesis was about minimum nondeterministic finite automata. We seized the strengths and limitations of known proof techniques for nondeterministic state complexity and gave a reformulation of these techniques as graph theoretic concepts. Then we looked at different notions of size of NFAs, namely the number of states *versus* the number of transitions. We found that these concepts are essentially different and the NFA minimization problem has to be studied separately for these two flavors. Unfortunately, nondeterministic transition complexity is far less understood. Consequently, classical computational complexity results were found mainly for state minimization. There we saw that the compactness of the input representation can largely influence the computational complexity of the NFA minimization problem if we consider finite languages. Such a phenomenon is known to occur in various computational settings, see, e.g. [63]. But earlier research on the NFA minimization problem [114] showed that the problem remains as hard if the input is specified less succinctly for infinite languages, so we could not expect such a result. Unfortunately, even when restricting to finite languages and specifying the input in the least succinct way that one may find, NFA minimization remains **NP**-hard. There already was evidence that no good approximation algorithms might exist either [70]. We put this evidence on more solid grounds and proved rather high limits on approximability of the variants of the problem under consideration. In this way, we provide a definite (negative) answer to the research question raised in 1993 by Jiang and Ravikumar, and also to most research problems posed more recently by Gramlich and Schnitger regarding other variants of this problem. What remains open yet is the quantitative question of determining more precise bounds: In the case where a unary NFA is given, we already determined the optimal bound under the weakest possible assumption. But in most of the other cases, still some gaps yawn between the upper and lower bounds. Perhaps most astonishing is that the question whether unary NFA minimization is **NP**-hard for given DFA, posed by Jiang et al. [112] almost twenty years ago, is still open.

Concerning the descriptional complexity of regular expressions, we obtained many results, partly with strongly negative algorithmic implications. Undoubtedly, the main result here is that the star height of a regular language can be at most logarithmic in the required regular expression size. While this is already a curious result on its own right, the counterpositive gives a powerful tool in proving lower bounds on regular expression size. This power is illustrated by the many tight bounds obtained not only for the conversion problem but also later, in the third part, on the effect of various language operations. For the problem of converting finite automata into regular expressions, we could prove a tight lower bound of $2^{\Theta(n)}$, where n is the number of states. Such a result had been established by Ehrenfeucht and Zeiger already in the 1970s. Yet the obvious catch in their lower bound was that they used an alphabet of size n^2. We managed to get around this not only by using a different proof technique, but also by crucially relying on a more recent concept from graph theory: Note that the existence of expander graphs was proved by Pinsker only at the time when Ehrenfeucht and Zeiger obtained their results.

Despite of the long list of results obtained by application of the technique based on star height, it tells us absolutely nothing about finite languages. There we devised a different technique, based on communication complexity. We related the sizes of regular expressions and monotone Boolean formulas. This allowed us to transfer results from computational complexity theory and thereby to resolve an old problem posed by Ehrenfeucht and Zeiger [55] concerning the size of regular expressions equivalent to acyclic finite automata. The ingredients in lower bound proofs often also hint at potential algorithmic hooks. This was the case here: We found an improved algorithm for converting DFAs over alphabets of constant size into regular expressions. This algorithm can be implemented as a strategy for state elimination. Notably, this algorithm gives the first improvement over the trivial bound that is known since the early 1960s. While the jump in performance from the previous $O(4^n)$ to $O(1.742^n)$ is already impressive, we feel that both our analysis and our techniques are far from optimal. A few things that come to mind are the following. We derived our bound based on the underlying undirected graph at the beginning of the elimination process. It might be possible to take edge directions and the dynamics of the elimination process better into account. Devising improved algorithms may be a challenging but surely rewarding research goal.

The last part of the thesis was devoted to the evolution of required regular expression size under various language operations. Regular expressions are often used as specification formalism, notably in the context of data exchange over the internet. Thus it was about time to answer the most basic questions regarding their dynamic succinctness properties. Some of these questions were explicitly posed by Ellul et al. [59]. We could provide an answer to several of these questions, including a tight doubly-exponential bound for the complement operation. Interestingly, Gelade and Neven worked at the same time along similar lines but using different techniques [65]. This witnesses that the time was indeed ripe to settle these questions. In several cases we showed that the simplest method of converting to a finite automaton, implementing the language operation under consideration in finite automata, and converting the result back to a regular expression is already optimal. In that last step, a sophisticated choice of the elimination ordering can significantly shorten the resulting regular expression, as we saw for the operations shuffle and intersection. In contrast, we also identified a few other language operations that can be implemented directly on regular expressions. In this way, we can avoid an exponential blow-up in size. There are of course many other interesting language operations waiting to be studied. Another direction is to study these problems for subclasses of regular languages. Often the instances that really arise in practical applications are much more constrained. For example, we determined the cost of the circular shift for finite languages. Some operations on other subclasses of regular languages were studied by Gelade and Neven [65].

Finally, we mention the following research direction, which might be a less obvious offspring of this thesis than the questions outlined above. Inspired by the tradeoff observed when comparing state minimization versus transition minimization for nondeterministic finite automata, we can ask similar questions for regular expressions. We think it is unlikely that every regular language admits a regular expression that is simultaneously of minimum alphabetic width and of minimum star height. This would indicate that there is a tradeoff between structural and descriptional complexity. A promising family of examples might be the languages presented in Example 2.5.

This last problem appears to be related to some difficult questions: For instance, a polynomial upper bound on the size of an equivalent regular expression of minimum star height would locate the famous star height problem in **PSPACE**. While the star height problem is known to be decidable for some time now, analyzing the computational complexity of the problem is still subject to active research, see e.g. [124].

We can also ask some more moderate questions with a similar flavor. For example, in this work, we have proved that the star height is always logarithmic in the alphabetic width of a regular language. It can be shown, using the methods presented in Chapter 10, that we can always obtain an expression of star height this low, while the size of the resulting expression is still polynomial in the minimum required size. Thus it is natural to ask whether we can trade expression size for star height: For instance, can we achieve sublinear star height while allowing only linear increase in expression size?

Thus it appears as if every obtained answer will entail a whole collection of new and intriguing research challenges. This stems from the fact that an improved understanding of a phenomenon often allows us to formulate more sophisticated, and also more detailed, questions about it.

Bibliography

[1] Alfred V. Aho, Ravi Sethi, and Jeffrey D. Ullman. *Compilers: Principles, Techniques, and Tools*. Addison-Wesley, 1986.

[2] Noga Alon, Jeff Kahn, and Paul D. Seymour. Large induced degenerate subgraphs. *Graphs and Combinatorics*, 3(1):203–211, 1987.

[3] Jérôme Amilhastre, Philippe Janssen, and Marie-Catherine Vilarem. FA minimisation heuristics for a class of finite languages. In Oliver Boldt and Helmut Jürgensen, editors, *4th International Workshop on Implementation of Automata*, volume 2214 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2001.

[4] Dana Angluin. Inference of reversible languages. *Journal of the ACM*, 29(3):741–765, 1982.

[5] Kenneth I. Appel and Wolfgang Haken. *Every planar map is four colorable*, volume 98 of *Contemporary Mathematics*. American Mathematical Society, 1989.

[6] Stefan Arnborg and Andrzej Proskurowski. Linear time algorithms for NP-hard problems restricted to partial k-trees. *Discrete Applied Mathematics*, 23(1):11–24, 1989.

[7] Robert L. Ashenhurst and Susan Graham, editors. *ACM Turing award lectures: the first twenty years: 1966-1985.* ACM Press/Addison-Wesley, 1987.

[8] Giorgio Ausiello, Marco Protasi, Alberto Marchetti-Spaccamela, Giorgio Gambosi, Pierluigi Crescenzi, and Viggo Kann. *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties*. Springer, 1999.

[9] Gérard Berry and Ravi Sethi. From regular expressions to deterministic automata. *Theoretical Computer Science*, 48(3):117–126, 1986.

[10] Jean Berstel and Jean-Eric Pin. Local languages and the Berry-Sethi algorithm. *Theoretical Computer Science*, 155(2):439–446, 1996.

[11] Dietmar Berwanger, Anuj Dawar, Paul Hunter, and Stephan Kreutzer. DAG-width and parity games. In Bruno Durand and Wolfgang Thomas, editors, *23rd Annual Symposium on Theoretical Aspects of Computer Science*, volume 3884 of *Lecture Notes in Computer Science*, pages 524–536. Springer, 2006.

[12] Dietmar Berwanger and Erich Grädel. Entanglement—A measure for the complexity of directed graphs with applications to logic and games. In Franz Baader and Andrei Voronkov, editors, *11th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning*, volume 3452 of *Lecture Notes in Computer Science*, pages 209–223. Springer, 2005.

[13] Philip Bille and Mikkel Thorup. Faster regular expression matching. In Susanne Albers, Alberto Marchetti-Spaccamela, Yossi Matias, Sotiris Nikoletseas, and Wolfgang Thomas, editors, *36th International Colloquium on Automata, Languages and Programming (Part I)*, volume 5555 of *LNCS*. Springer, 2009. To appear.

[14] Jean-Camille Birget. Intersection and union of regular languages and state complexity. *Information Processing Letters*, 43(4):185–190, 1992.

[15] Andreas Björklund, Thore Husfeldt, and Sanjeev Khanna. Approximating longest directed paths and cycles. In Josep Díaz, Juhani Karhumäki, Arto Lepistö, and Donald Sannella, editors, *31st International Colloquium on Automata, Languages and Programming*, volume 3142 of *Lecture Notes in Computer Science*, pages 222–233. Springer, 2004.

[16] Henrik Björklund and Wim Martens. The tractability frontier for NFA minimization. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walkuwiewicz, editors, *35th International Colloquium on Automata, Languages and Programming (Part II)*, volume 5126 of *Lecture Notes in Computer Science*, pages 27–38. Springer, 2008.

[17] Hans L. Bodlaender, Jitender S. Deogun, Klaus Jansen, Ton Kloks, Dieter Kratsch, Haiko Müller, and Zsolt Tuza. Rankings of graphs. *SIAM Journal on Discrete Mathematics*, 11(1):168–181, 1998.

[18] Hans L. Bodlaender, John R. Gilbert, Hjálmtyr Hafsteinsson, and Ton Kloks. Approximating treewidth, pathwidth, frontsize, and shortest elimination tree. *Journal of Algorithms*, 18(2):238–255, 1995.

[19] Hans L. Bodlaender and Babette van Antwerpen-de Fluiter. Parallel algorithms for series parallel graphs and graphs with treewidth two. *Algorithmica*, 29(4):534–559, 2001.

[20] Ronald V. Book and Ashok K. Chandra. Inherently nonplanar automata. *Acta Informatica*, 6(1):89–94, 1976.

[21] Janusz A. Brzozowski. Derivatives of regular expressions. *Journal of the ACM*, 11(4):481–494, 1964.

[22] Janusz A. Brzozowski. Open problems about regular languages. In Ronald V. Book, editor, *Formal language theory—Perspectives and open problems*, pages 23–47. Academic Press, 1980.

[23] Janusz A. Brzozowski and Edward J. McCluskey. Signal flow graph techniques for sequential circuit state diagrams. *IEEE Transactions on Electronic Computers*, EC-12(2):67–76, 1963.

[24] J. Richard Büchi. Weak second-order arithmetic and finite automata. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 6(1–6):66–92, 1960.

[25] Jean-Marc Champarnaud and Fabien Coulon. NFA reduction algorithms by means of regular inequalities. *Theoretical Computer Science*, 327(3):241–253, 2004. Erratum in 347(1-2):437–440, 2005.

[26] Jean-Marc Champarnaud and Jean-Eric Pin. A maxmin problem on finite automata. *Discrete Applied Mathematics*, 23(1):91–96, 1989.

[27] Jean-Marc Champarnaud and Djelloul Ziadi. Canonical derivatives, partial derivatives and finite automaton constructions. *Theoretical Computer Science*, 289(1):137–163, 2002.

[28] L. Sunil Chandran and Telikepalli Kavitha. The treewidth and pathwidth of hypercubes. *Discrete Mathematics*, 306(3):359–365, 2006.

[29] Richard Chang. Bounded queries, approximations, and the Boolean hierarchy. *Information and Computation*, 169(2):129–159, 2001.

[30] Laura Chaubard and Jean-Eric Pin. Open problems on regular languages: an historical perspective. In Jorge M. André, Vitor H. Fernandes, Mário J.J.Branco, Gracinda M. S. Gomes, John Fountain, and John Meakin, editors, *International Conference on Semigroups and Formal Languages—in Honour of the 65th Birthday of Donald B. McAlister*, pages 39–56. World Scientific, 2007.

[31] Noam Chomsky. Three models for the description of language. *IRE Transactions on Information Theory*, IT-2(3):113–124, 1956.

[32] Noam Chomsky. On certain formal properties of grammars. *Information and Control*, 2(2):137–167, 1959.

[33] Marek Chrobak. Finite automata and unary languages. *Theoretical Computer Science*, 47(2):149–158, 1986. Errata in 302(1–3):497–498, 2003, and in [60].

[34] Fan R. K. Chung. *Spectral Graph Theory*, volume 92 of *CBMS Regional Conference Series in Mathematics*. American Mathematical Society, 1997.

[35] Rina S. Cohen. Star height of certain families of regular events. *Journal of Computer and System Sciences*, 4(3):281–297, 1970.

[36] Rina S. Cohen and Janusz A. Brzozowski. General properties of star height of regular events. *Journal of Computer and System Sciences*, 4(3):260–280, 1970.

[37] John H. Conway. *Regular Algebra and Finite Machines*. Chapman and Hall, 1971.

[38] Bruno Courcelle, Damian Niwinski, and Andreas Podelski. A geometrical view of the determinization and minimization of finite-state automata. *Mathematical Systems Theory*, 24(2):117–146, 1991.

[39] Pierluigi Crescenzi. A short guide to approximation preserving reductions. In *12th Annual IEEE Conference on Computational Complexity*, pages 262–273. IEEE Computer Society, 1997.

[40] Berteun Damman, Tingting Han, and Joost-Pieter Katoen. Regular expressions for PCTL counterexamples. In *5th International Conference on the Quantitative Evaluation of Systems*, pages 179–188. IEEE Computer Society, 2008.

[41] Martin Davis. *The universal computer—The road from Leibniz to Turing*. W. W. Norton & Company, Inc., 2000.

[42] Conrado Daws. Symbolic and parametric model checking of discrete-time Markov chains. In Zhiming Liu and Keijiro Araki, editors, *1st International Colloquium on Theoretical Aspects of Computing*, volume 3407 of *Lecture Notes in Computer Science*, pages 280–294. Springer, 2004.

[43] Nicolaas Govert de Bruijn. Acknowledgement of priority to C. Flye Sainte-Marie on the counting of circular arrangements of 2^n zeros and ones that show each n-letter word exactly once. Technical Report 75-WSK-06, Technological University Eindhoven, 1975.

[44] Dominique de Caen, David A. Gregory, and Norman J. Pullman. The Boolean rank of zero-one matrices. In Charles C. Cadogan, editor, *3rd Caribbean Conference on Combinatorics and Computing*, pages 169–173. Department of Mathematics, University of the West Indies, 1981.

[45] Aldo de Luca and Stefano Varricchio. On noncounting regular classes. *Theoretical Computer Science*, 100(1):67–104, 1992.

[46] Françoise Dejean and Marcel Paul Schützenberger. On a question of eggan. *Information and Control*, 9(1):23–25, 1966.

[47] Manuel Delgado and José Morais. Approximation to the smallest regular expression for a given regular language. In Michael Domaratzki, Alexander Okhotin, Kai Salomaa, and Sheng Yu, editors, *9th Conference on Implementation and Application of Automata*, volume 3317 of *Lecture Notes in Computer Science*, pages 312–314. Springer, 2004.

[48] Reinhard Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer, 3rd edition, 2006.

[49] Benjamin Doerr. Communication complexity and the protocol parition number. Technical Report 99-28, Berichtsreihe des Mathematischen Seminars der Universität Kiel, 1999.

[50] Michael Domaratzki, Derek Kisman, and Jeffrey Shallit. On the number of distinct languages accepted by finite automata with n states. *Journal of Automata, Languages and Combinatorics*, 7(4):469–486, 2002.

[51] Michael Domaratzki and Kai Salomaa. Lower bounds for the transition complexity of NFAs. *Journal of Computer and System Sciences*, 74(7):1116–1130, 2008.

[52] Doron Drusinsky. *Modeling and Verification Using UML Statecharts: A Working Guide to Reactive System Design, Runtime Monitoring and Execution-based Model Checking*. Newnes, 2006.

[53] Keith Edwards and Graham E. Farr. Planarization and fragmentability of some classes of graphs. *Discrete Mathematics*, 308(12):2396–2406, 2008.

[54] Lawrence C. Eggan. Transition graphs and the star height of regular events. *Michigan Mathematical Journal*, 10(4):385–397, 1963.

[55] Andrzej Ehrenfeucht and H. Paul Zeiger. Complexity measures for regular expressions. *Journal of Computer and System Sciences*, 12(2):134–146, 1976.

[56] Samuel Eilenberg. *Automata, Languages, and Machines*, volume 59A/B of *Pure and Applied Mathematics—A Series of Monographs and Textbooks*. Academic Press, 1974. Two volumes (Vols. A/B). Vol. B includes 2 chapters by Bret Tilson.

[57] Calvin C. Elgot. Decision problems of finite automata design and related arithmetics. *Transactions of the American Mathematical Society*, 98(1):21–51, 1961.

[58] Keith Ellul. Descriptional complexity measures of regular languages. Master's thesis, University of Waterloo, Ontario, Canada, 2004.

[59] Keith Ellul, Bryan Krawetz, Jeffrey Shallit, and Ming-Wei Wang. Regular expressions: New results and open problems. *Journal of Automata, Languages and Combinatorics*, 10(4):407–437, 2005.

[60] Unary finite automata vs. arithmetic progressions. Anthony widjaja to. *Information Processing Letters*, 2009. To appear.

[61] Camille Flye Sainte-Marie. Question 48. *L'intermédiaire des mathématiciens*, 1:107–110, 1894. Also reproduced as appendix in [43].

[62] Martin Fürer. The complexity of the inequivalence problem for regular expressions with intersection. In Jaco W. de Bakker and Jan van Leeuwen, editors, *7th Colloquium on Automata, Languages and Programming*, volume 85 of *Lecture Notes in Computer Science*, pages 234–245. Springer, 1980.

[63] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* A Series of Books in the Mathematical Sciences. W. H. Freeman, 1979.

[64] Wouter Gelade. Succinctness of regular expressions with interleaving, intersection and counting. In Edward Ochmański and Jerzy Tyszkiewicz, editors, *33rd International Symposium on Mathematical Foundations of Computer Science*, volume 5162 of *Lecture Notes in Computer Science*, pages 363–374. Springer, 2008.

[65] Wouter Gelade and Frank Neven. Succinctness of the complement and intersection of regular expressions. In Susanne Albers and Pascal Weil, editors, *25th Annual Symposium on Theoretical Aspects of Computer Science*, volume 08001 of *Dagstuhl Seminar Proceedings*, pages 325–336. Schloss Dagstuhl - Leibniz Center for Informatics, 2008.

[66] Dennis Geller and Saul Stahl. The chromatic number and other functions of the lexicographic product. *Journal of Combinatorial Theory, Series B*, 19(1):87–95, 1975.

[67] Ian Glaister and Jeffrey Shallit. A lower bound technique for the size of nondeterministic finite automata. *Information Processing Letters*, 59(2):75–77, 1996.

[68] Georges Gonthier. Formal proof—The four-color theorem. *Notices of the American Mathematical Society*, 55(1):1382–1393, 2008.

[69] Gregor Gramlich. Probabilistic and nondeterministic unary automata. In Branislav Rovan and Peter Vojtás, editors, *28th International Symposium on Mathematical Foundations of Computer Science*, volume 2747 of *Lecture Notes in Computer Science*, pages 460–469. Springer, 2003.

[70] Gregor Gramlich and Georg Schnitger. Minimizing nfa's and regular expressions. *Journal of Computer and System Sciences*, 73(6):908–923, 2007.

[71] David A. Gregory and Norman J. Pullman. Semiring rank: Boolean rank and nonnegative rank factorizations. *Journal of Combinatorics, Information & System Sciences*, 8(3):223–233, 1983.

[72] David Gries. Describing an algorithm by Hopcroft. *Acta Informatica*, 2(2):97–109, 1973.

[73] Michelangelo Grigni and Michael Sipser. Monotone separation of logarithmic space from logarithmic depth. *Journal of Computer and System Sciences*, 50(3):433–437, 1995.

[74] Hermann Gruber and Markus Holzer. Finding lower bounds for nondeterministic state complexity is hard. In Oscar H. Ibarra and Zhe Dang, editors, *10th International Conference on Developments in Language Theory*, volume 4036 of *Lecture Notes in Computer Science*, pages 363–374. Springer, 2006.

[75] Hermann Gruber and Markus Holzer. Results on the average state and transition complexity of finite automata accepting finite languages (extended abstract). In Hing Leung and Giovanni Pighizzini, editors, *8th Workshop on Descriptional Complexity of Formal Systems*, Computer Science Technical Report NMSU-CS-2006-001, pages 267–275, 2006.

[76] Hermann Gruber and Markus Holzer. Computational complexity of NFA minimization for finite and unary languages. In Remco Loos, Szilárd Zsolt Fazekas, and Carlos Martín-Vide, editors, *1st International Conference on Language and Automata Theory and Applications*, Technical Report 35/07, pages 261–272. Research Group on Mathematical Linguistics, Universitat Rovira i Virgili, 2007.

[77] Hermann Gruber and Markus Holzer. Inapproximability of nondeterministic state and transition complexity assuming P $\neq$ NP. In Tero Harju, Juhani Karhumäki, and Arto Lepistö, editors, *11th International Conference on Developments in Language Theory*, volume 4588 of *Lecture Notes in Computer Science*, pages 205–216. Springer, 2007.

[78] Hermann Gruber and Markus Holzer. On the average state and transition complexity of finite languages. *Theoretical Computer Science*, 387(2):167–176, 2007.

[79] Hermann Gruber and Markus Holzer. Finite automata, digraph connectivity, and regular expression size. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walkuwiewicz, editors, *35th International Colloquium on Automata, Languages and Programming (Part II)*, volume 5126 of *Lecture Notes in Computer Science*, pages 39–50. Springer, 2008.

[80] Hermann Gruber and Markus Holzer. Language operations with regular expressions of polynomial size. In Cezar Câmpeanu and Giovanni Pighizzini, editors, *10th International Workshop on Descriptional Complexity of Formal Systems*, pages 182–193. CSIT University of Prince Edward Island, 2008.

[81] Hermann Gruber and Markus Holzer. Provably shorter regular expressions from deterministic finite automata (Extended abstract). In Masami Ito and Masafumi Toyama, editors, *12th International Conference on Developments in Language Theory*, volume 5257 of *Lecture Notes in Computer Science*. Springer, 2008.

[82] Hermann Gruber and Markus Holzer. Language operations with regular expressions of polynomial size. *Theoretical Computer Science*, 2009. To appear.

[83] Hermann Gruber and Markus Holzer. Tight bounds on the descriptional complexity of regular expressions. In Volker Diekert and Dirk Nowotka, editors, *13th International Conference on Developments in Language Theory*, volume 5583 of *LNCS*, pages 276–287. Springer, 2009. To appear.

[84] Hermann Gruber and Jan Johannsen. Optimal lower bounds on regular expression size using communication complexity. In Roberto Amadio, editor, *11th International Conference Foundations of Software Science and Computation Structures*, volume 4962 of *Lecture Notes in Computer Science*, pages 273–286. Springer, 2008.

[85] Igor Grunsky, Oleksiy Kurganskyy, and Igor Potapov. On a maximal NFA without mergible states. In Dima Grigoriev, John Harrison, and Edward A. Hirsch, editors, *First International Computer Science Symposium in Russia*, volume 3967 of *Lecture Notes in Computer Science*, pages 202–210. Springer, 2006.

[86] Stefan Gulan and Henning Fernau. An optimal construction of finite automata from regular expressions. In Ramesh Hariharan, Madhavan Mukund, and V. Vinay, editors, *28th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, volume 08004 of *Dagstuhl Seminar Proceedings*. Schloss Dagstuhl - Leibniz Center for Informatics, 2008.

[87] Rudolf Halin. S-functions for graphs. *Journal of Geometry*, 8(1–2):171–186, 1976.

[88] Yo-Sub Han and Derek Wood. Obtaining shorter regular expressions from finite-state automata. *Theoretical Computer Science*, 370(1-3):110–120, 2007.

[89] Godfrey H. Hardy and Edward M. Wright. *An Introduction to the Theory of Numbers*. Oxford University Press, 1979.

[90] Juris Hartmanis and Richard E. Stearns. On the computational complexity of algorithms. *Transactions of the American Mathematical Society*, 117(5):285–306, 1965.

[91] Kosaburo Hashiguchi. Algorithms for determining relative star height and star height. *Information and Computation*, 78(2):124–169, 1988.

[92] Kosaburo Hashiguchi and Namio Honda. Homomorphisms that preserve star height. *Information and Control*, 30(3):247–266, 1976.

[93] Markus Holzer and Martin Kutrib. Nondeterministic finite automata - Recent results on the descriptional and computational complexity. In Oscar H. Ibarra and Bala Ravikumar, editors, *13th International Conference on Implementation and Applications of Automata*, volume 5148 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2008.

[94] Markus Holzer and Martin Kutrib. Descriptional and computational complexity of finite automata. In Adrian H. Dediu, Armand-Mihai Ionescu, and Carlos Martín-Vide, editors, *3rd International Conference on Language and Automata Theory and Applications*, volume 5457 of *Lecture Notes in Computer Science*, pages 23–42. Springer, 2009.

[95] John E. Hopcroft. An $n \log n$ algorithm for minimizing the states in a finite automaton. In Zvi Kohavi and Azaria Paz, editors, *The Theory of Machines and Computations*, pages 189–196. Academic Press, 1971.

[96] John E. Hopcroft and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages and Computation.* Addison-Wesley Series in Computer Science. Addison-Wesley, 1979.

[97] Juraj Hromkovič. *Communication Complexity and Parallel Computing.* Texts in Theoretical Computer Science. An EATCS series. Springer, 1997.

[98] Juraj Hromkovič, Holger Petersen, and Georg Schnitger. On the limits of communication complexity technique for proving lower bounds on the size of minimal NFA's. *Theoretical Computer Science*, 2009. To appear.

[99] Juraj Hromkovič and Georg Schnitger. Comparing the size of NFAs with and without epsilon-transitions. *Theoretical Computer Science*, 380(1-2):100–114, 2007.

[100] Juraj Hromkovič and Georg Schnitger. On the hardness of determining small NFA's and of proving lower bounds on their sizes. In Masami Ito and Masafumi Toyama, editors, *12th International Conference on Developments in Language Theory*, volume 5257 of *Lecture Notes in Computer Science*, pages 34–55. Springer, 2008.

[101] Juraj Hromkovič and Georg Schnitger. Ambiguity and communication. In Susanne Albers and Jean-Yves Marion, editors, *26th International Symposium on Theoretical Aspects of Computer Science*, volume 09001 of *Dagstuhl Seminar Proceedings*, pages 553–564. Schloss Dagstuhl - Leibniz Center for Informatics, 2009.

[102] Juraj Hromkovič, Sebastian Seibert, Juhani Karhumäki, Hartmut Klauck, and Georg Schnitger. Communication complexity method for measuring nondeterminism in finite automata. *Information and Computation*, 172(2):202–217, 2002.

[103] Juraj Hromkovič, Sebastian Seibert, and Thomas Wilke. Translating regular expressions into small ε-free nondeterministic finite automata. *Journal of Computer and System Sciences*, 62(4):565–588, 2001.

[104] Juraj Hromkovič. Descriptional complexity of finite automata: Concepts and open problems. *Journal of Automata, Languages and Combinatorics*, 7(4):519–531, 2002.

[105] David A. Huffman. The synthesis of sequential switching circuits. *Journal of The Franklin Institute*, 257(3):161–190, 1954. See [106] for the second part of this article.

[106] David A. Huffman. The synthesis of sequential switching circuits. *Journal of The Franklin Institute*, 257(4):257–303, 1954. See [105] for the first part of this article.

[107] Paul Hunter and Stephan Kreutzer. Digraph measures: Kelly decompositions, games, and orderings. *Theoretical Computer Science*, 399(3):206–219, 2008.

[108] Harry B. Hunt III. The equivalence problem for regular expressions with intersection is not polynomial in tape. Technical Report Report TR 73-161, Department of Computer Science, Cornell University, 1973.

[109] Lucian Ilie, Roberto Solis-Oba, and Sheng Yu. Reducing the size of NFAs by using equivalences and preorders. In Alberto Apostolico, Maxime Crochemore, and Kunsoo Park, editors, *16th Annual Symposium on Combinatorial Pattern Matching*, volume 3537 of *Lecture Notes in Computer Science*, pages 310–321. Springer, 2005.

[110] Lucian Ilie and Sheng Yu. Follow automata. *Information and Computation*, 186(1):140–162, 2003.

[111] Lucian Ilie and Sheng Yu. Reducing NFAs by invariant equivalences. *Theoretical Computer Science*, 306(1-3):373–390, 2003.

[112] Tao Jiang, Edward McDowell, and Bala Ravikumar. The structure and complexity of minimal NFA's over unary alphabet. *International Journal of Foundations of Computer Science*, 2(2):163–182, 1991.

[113] Tao Jiang and Bala Ravikumar. A note on the space complexity of some decision problems for finite automata. *Information Processing Letters*, 40(1):25–31, 1991.

[114] Tao Jiang and Bala Ravikumar. Minimal NFA problems are hard. *SIAM Journal on Computing*, 22(6):1117–1141, 1993.

[115] Galina Jirásková. Note on minimal automata and uniform communication protocols. In Carlos Martín-Vide and Victor Mitrana, editors, *Grammars and Automata for String Processing. From Mathematics and Computer Science to Biology, and Back*, volume 9 of *Topics in Computer Mathematics*, pages 163–170. Taylor and Francis, 2003.

[116] Thor Johnson, Neil Robertson, Paul D. Seymour, and Robin Thomas. Directed tree-width. *Journal of Combinatorial Theory, Series B*, 82(1):138–154, 2001.

[117] Tsunehiko Kameda and Peter Weiner. On the state minimization of nondeterministic finite automata. *IEEE Transactions on Computers*, C-19(7):617–627, 1970.

[118] Christos A. Kapoutsis. Removing bidirectionality from nondeterministic finite automata. In Joanna Jedrzejowicz and Andrzej Szepietowski, editors, *30th International Symposium on Mathematical Foundations of Computer Science*, volume 3618 of *Lecture Notes in Computer Science*, pages 544–555. Springer, 2005.

[119] Mauricio Karchmer and Avi Wigderson. Monotone circuits for connectivity require super-logarithmic depth. *SIAM Journal on Computing*, 3:255–265, 2 1990.

[120] Meir Katchalski, William McCuaig, and Suzanne M. Seager. Ordered colourings. *Discrete Mathematics*, 142(1-3):141–154, 1995.

[121] Sanjeev Khanna, Rajeev Motwani, Madhu Sudan, and Umesh V. Vazirani. On syntactic versus computational views of approximability. *SIAM Journal on Computing*, 28(1):164–191, 1998.

[122] V. M. Khrapchenko. Methods for determining lower bounds for the complexity of Π-schemes (English translation). *Mathematical Notes of the Academy of Sciences of the USSR*, 10:474–479, 1972.

[123] Pekka Kilpeläinen and Rauno Tuhkanen. Regular expressions with numerical occurrence indicators—Preliminary results. In Pekka Kilpeläinen and Niina Päivinen, editors, *8th Symposium on Programming Languages and Software Tools*, pages 163–173. Department of Computer Science, University of Kuopio, Finland, 2003.

[124] Daniel Kirsten. Distance desert automata and the star height problem. *RAIRO – Theoretical Informatics and Applications*, 39(3):455–509, 2005.

[125] Stephen C. Kleene. Representation of events in nerve nets and finite automata. In Claude E. Shannon and John McCarthy, editors, *Automata Studies*, Annals of Mathematics Studies, pages 3–42. Princeton University Press, 1956.

[126] Donald E. Knuth. *The Art of Computer Programming, Volume 4, Fascicle 0: Introduction to Combinatorial Algorithms and Boolean Functions.* Addison-Wesley Professional, 2008.

[127] Donald E. Knuth, James H. Morris Jr., and Vaughan R. Pratt. Fast pattern matching in strings. *SIAM Journal on Computing*, 6(2):323–350, 1977.

[128] Sailesh Kumar, Sarang Dharmapurikar, Fang Yu, Patrick Crowley, and Jonathan S. Turner. Algorithms to accelerate multiple regular expressions matching for deep packet inspection. In Luigi Rizzo, Thomas E. Anderson, and Nick McKeown, editors, *ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pages 339–350. ACM, 2006.

[129] Eyal Kushilevitz and Noam Nisan. *Communication complexity.* Cambridge University Press, 1997.

[130] Quanzhong Li and Bongki Moon. Indexing and querying XML data for regular path expressions. In Peter M. G. Apers, Paolo Atzeni, Stefano Ceri, Stefano Paraboschi, Kotagiri Ramamohanarao, and Richard T. Snodgrass, editors, *27th International Conference on Very Large Data Bases*, pages 361–370. Morgan Kaufmann, 2001.

[131] Sylvain Lombardy. On the size of the universal automaton of a regular language. In Wolfgang Thomas and Pascal Weil, editors, *24th Annual Symposium on Theoretical Aspects of Computer Science*, volume 4393 of *Lecture Notes in Computer Science*, pages 85–96. Springer, 2007.

[132] Oleg Borisovich Lupanov. A comparison of two types of finite sources. *Problemy kibernetiki*, 9:321–326, 1963. (In Russian).

[133] Michael S. Mahoney. What was the question? The origins of the theory of computation. In Atsushi Akera and William Aspray, editors, *Using History to Teach Computer Science and Related Disciplines*, pages 225–232. Computing Research Association, 2004.

[134] Andreas Malcher. Minimizing finite automata is computationally hard. *Theoretical Computer Science*, 327(3):375–390, 2004.

[135] Wim Martens, Frank Neven, and Thomas Schwentick. Simple off the shelf abstractions for XML schema. *SIGMOD Record*, 36(3):15–22, 2007.

[136] A. N. Maslov. Estimates of the number of states of finite automata. *Soviet Mathematics Doklady*, 11(5):1373–1375, 1970.

[137] Alain J. Mayer and Larry J. Stockmeyer. Word problems—This time with interleaving. *Information and Computation*, 115(2):293–311, 1994.

[138] Warren S. McCulloch. *Embodiments of Mind.* MIT press, 1988.

[139] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133, 1943. Reprinted in [138].

[140] Robert McNaughton. The loop complexity of pure-group events. *Information and Control*, 11(1/2):167–176, 1967.

[141] Robert McNaughton. The loop complexity of regular events. *Information Sciences*, 1(3):305–328, 1969.

[142] Robert McNaughton and Hisao Yamada. Regular expressions and state graphs for automata. *IRE Transactions on Electronic Computers*, EC-9(1):39–47, 1960.

[143] Albert R. Meyer and Michael J. Fischer. Economy of description by automata, grammars, and formal systems. In *12th Annual IEEE Symposium on Switching and Automata Theory*, pages 188–191. IEEE Computer Society, 1971.

[144] Albert R. Meyer and Larry J. Stockmeyer. The equivalence problem for regular expressions with squaring requires exponential space. In *13th Annual IEEE Symposium on Switching and Automata Theory*, pages 125–129. IEEE Computer Society, 1972.

[145] Edward F. Moore. Gedanken experiments on sequential machines. In Claude E. Shannon and John McCarthy, editors, *Automata Studies*, Annals of Mathematics Studies, pages 129–153. Princeton University Press, 1956.

[146] Frank R. Moore. On the bounds for state-set size in the proofs of equivalence between deterministic, nondeterministic, and two-way finite automata. *IEEE Transactions on Computers*, C-20(10):1211–1214, 1971.

[147] Nelma Moreira and Rogério Reis. Series-parallel automata and short regular expressions. *Fundamenta Informaticae*, 91(3-4):611–629, 2009.

[148] Paul H. Morris, Ronald A. Gray, and Robert E. Filman. GOTO removal based on regular expressions. *Journal of Software Maintenance*, 9(1):47–66, 1997.

[149] Jaroslav Nešetřil and Patrice Ossona de Mendez. Tree-depth, subgraph coloring and homomorphism bounds. *European Journal of Combinatorics*, 27(6):1022–1041, 2006.

[150] Jan Obdržálek. DAG-width: Connectivity measure for directed graphs. In *17th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 814–821. ACM Press, 2006.

[151] James B. Orlin. Contentment in graph theory: Covering graphs with cliques. *Indagationes Mathematicae*, 80(5):406–424, 1977.

[152] Christos H. Papadimitriou. *Computational complexity.* Addison-Wesley, 1994.

[153] Holger Petersen. The membership problem for regular expressions with intersection is complete in LOGCFL. In Helmut Alt and Afonso Ferreira, editors, *19th Annual Symposium on Theoretical Aspects of Computer Science*, volume 2285 of *Lecture Notes in Computer Science*, pages 513–522. Springer, 2002.

[154] Giovanni Pighizzini and Jeffrey Shallit. Unary language operations, state complexity and Jacobsthal's function. *International Journal of Foundations of Computer Science*, 13(1):145–159, 2002.

[155] Jean-Eric Pin. Syntactic semigroups. In Grzegorz Rozenberg and Arto Salomaa, editors, *Handbook of formal languages, Vol. 1: Word, language, grammar*, pages 679–746. Springer, 1997.

[156] Mark Semenovich Pinsker. On the complexity of a concentrator. In *7th International Teletraffic Conference*, pages 318/1–318/4, 1973.

[157] Michael O. Rabin and Dana Scott. Finite automata and their decision problems. *IBM Journal of Research and Development*, 3(2):114–125, 1959.

[158] Neil Robertson, Daniel P. Sanders, Paul D. Seymour, and Robin Thomas. The four-colour theorem. *Journal of Combinatorial Theory, Series B*, 70(1):2–44, 1997.

[159] Neil Robertson and Paul D. Seymour. Graph minors. III. Planar tree-width. *Journal of Combinatorial Theory, Series B*, 36(1):49–64, 1984.

[160] Neil Robertson and Paul D. Seymour. Graph minors. II. Algorithmic aspects of tree-width. *Journal of Algorithms*, 7(3):309–322, 1986.

[161] John Michael Robson. The emptiness of complement problem for semi extended regular expressions requires c^n space. *Information Processing Letters*, 9(5):220–222, 1979.

[162] Grigore Roşu. An effective algorithm for the membership problem for extended regular expressions. In Helmut Seidl, editor, *10th International Conference on Foundations of Software Science and Computational Structures*, volume 4423 of *LNCS*, pages 332–345. Springer, 2007.

[163] Jacques Sakarovitch. The language, the expression, and the (small) automaton. In Jacques Farré, Igor Litovsky, and Sylvain Schmitz, editors, *10th International Conference on Implementation and Application of Automata*, volume 3845 of *Lecture Notes in Computer Science*, pages 15–30, 2005.

[164] Kai Salomaa. Descriptional complexity of nondeterministic finite automata. In Tero Harju, Juhani Karhumäki, and Arto Lepistö, editors, *11th International Conference on Developments in Language Theory*, volume 4588 of *Lecture Notes in Computer Science*, pages 31–35. Springer, 2007.

[165] Georg Schnitger. Regular expressions and NFAs without ε-transitions. In Bruno Durand and Wolfgang Thomas, editors, *23rd Annual Symposium on Theoretical Aspects of Computer Science*, volume 3884 of *Lecture Notes in Computer Science*, pages 432–443, 2006.

[166] Marcel Paul Schützenberger. On finite monoids having only trivial subgroups. *Information and Control*, 8(2):190–194, 1965.

[167] Kyriakos N. Sgarbas, Nikos D. Fakotakis, and George K. Kokkinakis. Incremental construction of compact acyclic NFAs. In *39th Annual Meeting of the Association for Computational Linguistics*, pages 482–489. Association for Computational Linguistics, 2001.

[168] Jeffrey Shallit. *A Second Course in Formal Languages and Automata Theory*. Cambridge University Press, 2009.

[169] Peter W. Shor. A counterexample to the triangle conjecture. *Journal of Combinatorial Theory, Series A*, 38(1):110–112, 1985.

[170] Hans-Ulrich Simon. On approximate solutions for combinatorial optimization problems. *SIAM Journal on Discrete Mathematics*, 3(2):294–310, 1990.

[171] Richard E. Stearns and Harry B. Hunt III. On the equivalence and containment problems for unambiguous regular expressions, regular grammars and finite automata. *SIAM Journal on Computing*, 14(3):598–611, 1985.

[172] Larry J. Stockmeyer and Albert R. Meyer. Word problems requiring exponential time: Preliminary Report. In *5th Annual ACM Symposium on Theory of Computing*, pages 1–9. ACM, 1973.

[173] Ambuj Tewari, Utkarsh Srivastava, and Phalguni Gupta. A parallel DFA minimization algorithm. In Sartaj Sahni, Viktor K. Prasanna, and Uday Shukla, editors, *9th International Conference on High Performance Computing*, volume 2552 of *Lecture Notes in Computer Science*, pages 34–40. Springer, 2002.

[174] Boris A. Trakhtenbrot. Finite automata and the logic of monadic predicates. *Doklady Akademii Nauk SSSR*, 140:326–329, 1961.

[175] Luca Trevisan. *Reductions and (Non-)Approximability*. PhD thesis, University of Rome "La Sapienza", 1997. PhD Theses Series of Electronic Colloquium on Computational Complexity.

[176] Pál Turán. On an extremal problem in graph theory (in Hungarian). *Matematicko Fizicki Lapok*, 48:436–452, 1941.

[177] Volker Waizenegger. On the efficiency of description by regular expressions and finite automata (in German). Diploma thesis, Rheinisch-Westfälische Technische Hochschule Aachen, Fachbereich Informatik, 2000.

[178] Manfred K. Warmuth and David Haussler. On the complexity of iterated shuffle. *Journal of Computer and System Sciences*, 28(3):345–358, 1984.

[179] Sheng Yu. Regular languages. In Grzegorz Rozenberg and Arto Salomaa, editors, *Handbook of formal languages, Vol. 1: Word, language, grammar*, pages 41–110. Springer, 1997.

[180] Sheng Yu. A renaissance of automata theory? *Bulletin of the EATCS*, 72:270–272, 2000.

[181] Sheng Yu. State complexity: Recent results and open problems. *Fundamenta Informaticae*, 64(1-4):471–480, 2005.

[182] David Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory of Computing*, 3(1):103–128, 2007.

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

Y

Z