Fundamenta Informaticae 191(3-4) : 231–237 (2024)
Available at IOS Press through:
https://doi.org/10.3233/FI-242180

Descriptional Complexity of Finite Automata – Selected Highlights

Arto Salomaa
Department of Mathematics and Statistics
University of Turku, 20014 Turku, Finland
asalomaa@utu.fi

Kai Salomaa*
School of Computing
Queen's University, Kingston, Ontario, Canada
salomaa@queensu.ca

Taylor J. Smith
Department of Computer Science
St. Francis Xavier University, Antigonish, Nova Scotia, Canada
tjsmith@stfx.ca

In honor of the 60th birthday of Iiro Honkala.

Abstract. The state complexity, respectively, nondeterministic state complexity of a regular language L is the number of states of the minimal deterministic, respectively, of a minimal nondeterministic finite automaton for L. Some of the most studied state complexity questions deal with size comparisons of nondeterministic finite automata of differing degree of ambiguity. More generally, if for a regular language we compare the size of description by a finite automaton and by a more powerful language definition mechanism, such as a context-free grammar, we encounter non-recursive trade-offs. Operational state complexity studies the state complexity of the language resulting from a regularity preserving operation as a function of the complexity of the argument languages. Determining the state complexity of combined operations is generally challenging and for general combinations of operations that include intersection and marked concatenation it is uncomputable.

Keywords: finite automaton, state complexity, degree of ambiguity, regularity preserving operation, undecidability

*Address for correspondence: School of Computing, Queen's University, Kingston, Ontario, Canada.

1. Introduction

Descriptional complexity studies the relative succinctness between different representations of formal languages [1]. For a quantitative understanding of regular languages the commonly used size measures count the number of states or, in the case of nondeterministic finite automata, the number of transitions. Early work on descriptional complexity of finite automata includes [2, 3, 4, 5, 6].

One of the most important open problems in descriptional complexity was originally raised by Sakoda and Sipser in 1978 [7]. The question asks whether any two-way nondeterministic finite automaton M has an equivalent deterministic two-way automaton with a number of states bounded by a polynomial on the number of states of M. Already Sakoda and Sipser conjectured a negative answer to this question. Berman [8] and Sipser [9] showed that if one proves an exponential gap in the succinctness of nondeterministic and deterministic two-way automata and the strings involved in the separation have polynomial length, this implies that deterministic logarithmic space is a proper subset of nondeterministic logarithmic space. Also Sipser [9] introduced a restricted version of two-way automata, called sweeping automata, where the reading head may reverse only at the end-markers and proved an exponential separation between nondeterministic and deterministic sweeping automata. While the original question on the succinctness comparison of nondeterministic and deterministic two-way automata remains open, an exponential separation has been established, besides sweeping automata, for several other restricted variants [10, 11, 12].

This brief survey discusses three specific descriptional complexity topics: non-recursive trade-offs, the state complexity trade-off between nondeterministic finite automata with differing degrees of ambiguity, and, the state complexity of language operations. We generally focus on finite automaton models, however, non-recursive trade-offs naturally deal with succinctness comparisons with more powerful models. Descriptional complexity is a large and active research area and more information can be found e.g. in the surveys [13, 14, 15, 16, 17, 18, 12, 19].

2. Non-recursive trade-offs

In a seminal work Stearns [6] studied the relative succinctness of regular languages represented by deterministic finite automata (DFA) and deterministic pushdown automata (PDA). He showed that if a deterministic PDA recognizes a regular language it can be a simulated by a DFA of triple-exponential size. The work establishes also the decidability of regularity of the language of a deterministic PDA. Generally the difference in succinctness of description between different representations of a regular language can be arbitrary. This phenomenon is referred to as a non-recursive trade-off.

More formally, a family of languages $\mathcal{L}$ is represented by a descriptional system S if $\mathcal{L} = \{L(\mathcal{R}) \mid \mathcal{R} \in S\}$. Here $L(\mathcal{R})$ is the language represented by descriptor $\mathcal{R}$. A complexity measure is a total recursive function $c : S \to \mathbb{N}$. For descriptional systems S_1 and S_2 equipped with a complexity measure c, a function f is said to be an upper bound for the size blow-up for changing from system S_1 to S_2 if for every language L that has representations in both systems, for every representation $\mathcal{R}_1$ of L in S_1, the language L has a representation $\mathcal{R}_2$ in S_2 where $c(\mathcal{R}_2) \leq f(c(\mathcal{R}_1))$. Note that, for example, when considering the trade-off between finite automata and pushdown automata we consider only the size of representations of regular languages. The trade-off between two descriptional systems is said to be *non-recursive* if it is not upper bounded by any recursive function.

A central part of descriptional complexity research deals with non-recursive trade-offs. As a first non-recursive size blow-up, Meyer and Fischer [4] showed that between finite automata and general context-free grammars for regular languages the difference in economy of description can be arbitrary, that is, the trade-off is not upper bounded by any recursive function. Hartmanis [20] showed that the descriptional complexity trade-off between deterministic and nondeterministic PDA is non-recursive, even if the nondeterministic PDA is equipped with a proof in a formal system that it defines a deterministic language. The proof is based on the fundamental idea due to Hartmanis that invalid computations of a Turing machine can be encoded as a context-free language [21].

A couple of years later, based on Gödel's technique for non-recursive shortening of proofs of formal systems by additional axioms, Hartmanis [1] extended the method as a general technique for proving non-recursive succinctness trade-offs. We state the result on non-recursive trade-offs using the formulation by Kutrib [22] that is independent of a particular complexity measure.

Theorem 2.1. ([1, 22])
Let S_1 and S_2 be descriptional systems for recursive languages. The trade-off between S_1 and S_2 is non-recursive if the following conditions hold.

There exists a descriptional system S_3 and a property P that is not semi-decidable for languages with a representation in S_3 such that, given an arbitrary representation $\mathcal{R} \in S_3$, there exists an effective procedure to construct a representation in S_1 for some language $L_{\mathcal{R}}$ with the property that $L_{\mathcal{R}}$ has a representation in S_2 if and only if $L(\mathcal{R})$ does not have property P.

In fact, as noted in [22] most proofs appearing in the literature to establish non-recursive trade-offs rely on a technique analogous to Theorem 2.1, or are based on context-free language encodings of invalid Turing machine computations from [21].

3. Ambiguity of NFAs and state complexity

The state complexity $\mathrm{sc}(L)$ (respectively, nondeterministic state complexity $\mathrm{nsc}(L)$) of a regular language L is the minimal number of states of a DFA (respectively, of an NFA) for L. Already from [2, 4, 5] it is known that, for some regular languages L, $\mathrm{sc}(L) = 2^{\mathrm{nsc}(L)}$.

The degree of ambiguity of an NFA A on a string w is the number of accepting computations of A on w. The NFA A is *unambiguous* (UFA) if any string has at most one accepting computation. If the ambiguity of A on any string is bounded by a constant, A is *finitely ambiguous* (FNFA) and A is *polynomially ambiguous* (PNFA) if the degree of ambiguity of A on input w is bounded by a polynomial in the length of w.

Schmidt [23] developed methods to prove lower bounds for the size of UFAs and showed that there exists an n-state UFA where the smallest equivalent DFA requires $2^{\Omega(\sqrt{n})}$ states. The lower bound was improved by different authors and Leiss [24] gives a construction of an n-state UFA with multiple initial states where an equivalent DFA need 2^n states. Leung [25] established the lower bound

2^n for determinization of an UFA with one initial state, as well as, showed that there exists an n state FNFA for which any equivalent UFA needs $2^n - 1$ states.

Ravikumar and Ibarra [26] first considered systematically succinctness comparisons between FNFAs, PNFAs and general NFAs, and showed that any NFA recognizing a bounded language can be converted to an FNFA with polynomial size blow-up. Leung [27] gave an optimal separation between PNFAs and general NFAs. Using communication complexity Hromkovič et al. [28] give a significantly simplified proof for a super-polynomial separation of NFAs and PNFAs, however, their proof does not give the exact optimal size blow-up $2^n - 1$.

Theorem 3.1. ([27])
For $n \in \mathbb{N}$ there exists an n-state NFA A_n such that any PNFA for the language $L(A_n)$ needs $2^n - 1$ states.

Ravikumar and Ibarra [26] also conjectured that polynomially ambiguous NFAs can be significantly more succinct than finitely ambiguous NFAs. The question was solved affirmatively by Hromkovič and Schnitger [29]. The below theorem gives a simplified special case of the result in [29] that gives a superpolynomial succinctness separation between NFAs with degree of ambiguity, respectively, $O(m^{k-1})$ and $O(m^k)$, $k \in \mathbb{N}$. However, the lower bound for the separation of n-state PNFAs and FNFAs is not $2^{\Theta(n)}$ and the precise trade-off in the economy of description remains open.

Theorem 3.2. ([29])
For $n \in \mathbb{N}$ there exists a PNFA A_n with number of states polynomial in n such that any FNFA recognizing the language $L(A_n)$ has at least $2^{\Omega(n^{\frac{1}{3}})}$ states.

Besides ambiguity the degree of nondeterminism can be measured, roughly speaking, by counting the number of guesses in one computation [30] or by counting the number of all computations. The *tree width*, a.k.a. *leaf size* or *path size* of an NFA A on input w is the number of leaves of the computation tree of A on w [31, 32, 18, 33]. It is easy to see that an NFA with finite tree width can be determinized with polynomial size blow-up but very little is known about succinctness comparisons of NFAs with different non-constant tree width growth rates. For example, it remains open whether an NFA with polynomial tree width may, in the worst case, require super-polynomially more states than an equivalent unrestricted NFA. Similarly, the succinctness comparison between NFAs, respectively, of finite and polynomial tree width remains open.

4. Operational state complexity

The effect of a regularity preserving operation f on the size of the minimal DFA (respectively, on the size of a minimal NFA) is the *operational state complexity* of the operation. This is defined formally below.

Definition 4.1. If f is an m-ary regularity preserving language operation, a (deterministic) *state complexity upper bound* of f is a function $g : \mathbb{N}^m \to \mathbb{N}$ such that for any regular languages $L_1, \ldots, L_m$, the language $f(L_1, \ldots, L_m)$ has a DFA with at most $g(\mathrm{sc}(L_1), \ldots, \mathrm{sc}(L_m))$ states.

The nondeterministic state complexity of an operation f is defined similarly. A function $f_{\mathrm{sc}} : \mathbb{N}^m \to \mathbb{N}$ is the precise worst-case state complexity of f if f_{sc} is a state complexity upper bound of f and, furthermore, for any positive integers $n_1, \ldots, n_m$ there exist regular languages L_i with $\mathrm{sc}(L_i) = n_i$, $i = 1, \ldots, m$, and the minimal DFA for $f(L_1, \ldots, L_m)$ has $f_{\mathrm{sc}}(n_1, \ldots, n_m)$ states.

The state complexity of language operations was first considered by Maslov [3] but the paper remained unknown in the west. A systematic study of operational state complexity of regular languages was initiated by S. Yu in the 1990's [19, 34]. The operational state complexity of extensions of finite automata that have strong closure properties, such as input-driven pushdown automata, a.k.a. visibly pushdown automata, has also been considered [35, 36].

In a series of papers Yu and co-authors have investigated the state complexity of combined operations and have determined the precise worst-case state complexity of all combinations of two basic language operations [37, 13]. Establishing matching upper and lower bounds for the state complexity of combined language operations is often involved and Ésik et al. [38] have introduced techniques to estimate the state complexity of combined operations. For a general combination of operations that include marked concatenation and intersection, Yu et al. [39] have shown that the question whether a given integer function is a state complexity upper bound is undecidable in the following sense.

The marked concatenation of languages $L_1, L_2, \ldots, L_n$ is defined as $L_1 \sharp L_2 \sharp \cdots \sharp L_n$ where $\sharp$ is a new symbol not appearing in the languages L_i. A $(\cap, \sharp)$-composition over the set $\{L_1, L_2, \ldots, L_n\}$, $n \geq 2$, of language variables is an expression $\beta_1 \sharp \beta_2 \sharp \cdots \sharp \beta_r$, $r \geq 2$, where each β_i is of the form

$$\beta_i = K_1 \cap K_2 \cap \cdots \cap K_{t_i}, \quad 1 \leq t_i \leq n,$$

where K_j's are distinct among the language variables L_i, $i = 1, \ldots, n$. A sequence of $(\cap, \sharp)$-compositions C_i, $i = 1, 2, \ldots$, is effectively constructible if there is an algorithm that on input $i \in \mathbb{N}$ outputs C_i.

Theorem 4.2. ([39])
A sequence of $(\cap, \sharp)$-compositions C_i, can be effectively constructed such that, given $i \in \mathbb{N}$ and a polynomial with positive integer coefficients P over the same number of variables as C_i, it is undecidable whether or not P is a state complexity upper bound for the composition C_i (as defined in Definition 4.1).

References

[1] Hartmanis J. On Gödel speed-up and succinctness of language representations. *Theoretical Computer Science*, 1983. **26**(3):335–342. doi:10.1016/0304-3975(83)90016-6.

[2] Lupanov OB. A comparison of two types of finite sources (O sravnenii dvukh tipov konechnykh istochnikov). *Problemy Kibernetiki*, 1963. **9**:321–326.

[3] Maslov AN. Estimates of the number of states of finite automata. *Soviet Math. Dokl.*, 1970. **11**(5):1373–1375.

[4] Meyer AR, Fischer MJ. Economy of description by automata, grammars, and formal systems. In: Proceedings of SWAT 1971. IEEE, 1971 pp. 188–191. doi:10.1109/SWAT.1971.11.

[5] Moore FR. On the bounds for state-set size in the proofs of equivalence between deterministic, nondeterministic, and two-way finite automata. *IEEE Trans. Comput.*, 1971. **C-20**(10):1211–1214. doi: 10.1109/T-C.1971.223108.

[6] Stearns RE. A regularity test for pushdown machines. *Information and Control*, 1967. **11**(3):323–340. doi:10.1016/S0019-9958(67)90591-8.

[7] Sakoda WJ, Sipser M. Nondeterminism and the size of two way finite automata. In: Proceedings of STOC 1978. ACM, 1978 pp. 275–286. doi:10.1145/800133.804357.

[8] Berman P. A note on sweeping automata. In: Proceedings of ICALP 1980, volume 85 of *LNCS*. Springer, 1980 pp. 91–97. doi:10.1007/3-540-10003-2_62.

[9] Sipser M. Lower bounds on the size of sweeping automata. *Journal of Computer and System Sciences*, 1980. **21**(2):195–202. doi:10.1016/0022-0000(80)90034-3.

[10] Hromkovič J, Schnitger G. Nondeterminism versus determinism for two-way finite automata: Generalizations of Sipser's separation. In: Proceedings of ICALP 2003, volume 2719 of *LNCS*. Springer, 2003 pp. 439–451. doi:10.1007/3-540-45061-0_36.

[11] Kapoutsis CA. Two-way automata versus logarithmic space. *Theory of Computing Systems*, 2014. **55**:421–447. doi:10.1007/s00224-013-9465-0.

[12] Pighizzini G. Two-way finite automata: Old and recent results. *Fundamenta Informaticae*, 2013. **126**(2–3):225–246. doi:10.3233/FI-2013-879.

[13] Gao Y, Moreira N, Reis R, Yu S. A survey on operational state complexity. *Journal of Automata, Languages and Combinatorics*, 2017. **21**(4):251–310. doi:10.25596/jalc-2016-251.

[14] Goldstine J, Kappes M, Kintala CMR, Leung H, Malcher A, Wotschke D. Descriptional complexity of machines with limited resources. *Journal of Universal Computer Science*, 2002. **8**(2):193–234. doi: 10.3217/jucs-008-02-0193.

[15] Gruber H, Holzer M. From finite automata to regular expressions and back — A summary on descriptional complexity. *International Journal of Foundations of Computer Science*, 2015. **26**(8):1009–1040. doi: 10.1142/S0129054115400110.

[16] Gruber H, Holzer M, Kutrib M. Descriptional complexity of regular languages. In: Pin JÉ (ed.), Handbook of Automata Theory, volume 1, pp. 411–457. EMS Press, 2021. doi:10.4171/AUTOMATA-1/12.

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

[18] Hromkovič J. Descriptional complexity of finite automata: Concepts and open problems. *Journal of Automata, Languages and Combinatorics*, 2002. **7**(4):519–531. doi:10.25596/jalc-2002-519.

[19] Yu S. Regular languages. In: Rozenberg G, Salomaa A (eds.), Handbook of Formal Languages, volume 1, pp. 41–110. Springer, 1997. doi:10.1007/978-3-642-59136-5_2.

[20] Hartmanis J. On the succinctness of different representations of languages. *SIAM Journal of Computing*, 1980. **9**(1):114–120. doi:10.1137/0209010.

[21] Hartmanis J. Context-free languages and Turing machine computations. In: Mathematical Aspects of Computer Science, volume 19 of *Proc. Sympos. Appl. Math.* American Mathematical Society, 1967 pp. 42–51. doi:10.1090/psapm/019/0235938.

[22] Kutrib M. The phenomenon of non-recursive trade-offs. *International Journal of Foundations of Computer Science*, 2005. **16**(5):957–973. doi:10.1142/S0129054105003406.

[23] Schmidt EM. Succinctness of descriptions of context-free, regular, and finite languages. Ph.D. thesis, Cornell University, 1978.

[24] Leiss E. Succinct representation of regular languages by boolean automata. *Theoretical Computer Science*, 1981. **13**(3):323–330. doi:10.1016/S0304-3975(81)80005-9.

[25] Leung H. Descriptional complexity of NFA of different ambiguity. *International Journal of Foundations of Computer Science*, 2005. **16**(5):975–984. doi:10.1142/S0129054105003418.

[26] Ravikumar B, Ibarra OH. Relating the type of ambiguity of finite automata to the succinctness of their representation. *SIAM Journal of Computing*, 1989. **18**(6):1263–1282. doi:10.1137/0218083.

[27] Leung H. Separating exponentially ambiguous finite automata from polynomially ambiguous finite automata. *SIAM Journal of Computing*, 1998. **27**(4):1073–1082. doi:10.1137/S0097539793252092.

[28] Hromkovič J, Seibert S, Karhumäki J, Klauck H, Schnitger G. Communication complexity method for measuring nondeterminism in finite automata. *Information and Computation*, 2002. **172**(2):202–217. doi:10.1006/inco.2001.3069.

[29] Hromkovič J, Schnitger G. Ambiguity and communication. *Theory of Computing Systems*, 2011. **48**:517–534. doi:10.1007/s00224-010-9277-4.

[30] Goldstine J, Kintala CMR, Wotschke D. On measuring nondeterminism in regular languages. *Information and Computation*, 1990. **86**(2):179–194. doi:10.1016/0890-5401(90)90053-K.

[31] Björklund H, Martens W. The tractability frontier for NFA minimization. *Journal of Computer and System Sciences*, 2012. **78**(1):198–210. doi:10.1016/j.jcss.2011.03.001.

[32] Han YS, Salomaa A, Salomaa K. Ambiguity, nondeterminism and state complexity of finite automata. *Acta Cybernetica*, 2017. **23**(1):141–157. doi:10.14232/actacyb.23.1.2017.9.

[33] Palioudakis A, Salomaa K, Akl SG. State complexity of finite tree width NFAs. *Journal of Automata, Languages and Combinatorics*, 2012. **17**(2–4):245–264. doi:10.25596/jalc-2012-245.

[34] Yu S. State complexity of regular languages. *Journal of Automata, Languages and Combinatorics*, 2001. **6**(2):221–234. doi:10.25596/jalc-2001-221.

[35] Alur R, Madhusudan P. Visibly pushdown languages. In: Proceedings of STOC 2004. ACM, 2004 pp. 202–211. doi:10.1145/1007352.1007390.

[36] Okhotin A, Salomaa K. Complexity of input-driven pushdown automata. *ACM SIGACT News*, 2014. **45**(2):47–67. doi:10.1145/2636805.2636821.

[37] Cui B, Gao Y, Kari L, Yu S. State complexity of combined operations with two basic operations. *Theoretical Computer Science*, 2012. **437**:82–102. doi:10.1016/j.tcs.2012.02.030.

[38] Ésik Z, Gao Y, Liu G, Yu S. Estimation of state complexity of combined operations. *Theoretical Computer Science*, 2009. **410**(35):3272–3280. doi:10.1016/j.tcs.2009.03.026.

[39] Salomaa A, Salomaa K, Yu S. Undecidability of state complexity. *International Journal of Computer Mathematics*, 2013. **90**(6):1310–1320. doi:10.1080/00207160.2012.704994.