Acta Cybernetica **23** (2017) 141–157.

Ambiguity, Nondeterminism and State Complexity of Finite Automata

Yo-Sub Han[a], Arto Salomaa[b], and Kai Salomaa[c]

Abstract

The degree of ambiguity counts the number of accepting computations of a nondeterministic finite automaton (NFA) on a given input. Alternatively, the nondeterminism of an NFA can be measured by counting the amount of guessing in a single computation or the number of leaves of the computation tree on a given input. This paper surveys work on the degree of ambiguity and on various nondeterminism measures for finite automata. In particular, we focus on state complexity comparisons between NFAs with quantified ambiguity or nondeterminism.

Keywords: finite automata, nondeterminism, degree of ambiguity, state complexity

Dedicated to the memory of Zoltán Ésik (1951–2016).

1 Introduction

Finite automata are a fundamental model of computation that has been systematically studied since the 1950's. At the same time many important questions on finite automata and regular languages remain open [7, 18, 52]. The last decades have seen much work on the descriptional complexity, or state complexity, of regular languages [10, 13, 15, 16, 17, 28]. The state complexity (respectively, nondeterministic state complexity) of a regular language L is the optimal size of a deterministic finite automaton (DFA) (respectively, a nondeterministic finite automaton (NFA)) recognizing L. The effect of a regularity preserving operation on the minimal DFA (or alternatively a minimal NFA) is called the state complexity of the operation. The state complexity of basic operations on regular languages was considered first by Maslov [34] and further references can be found in the survey by Gao et al. [9].

[a] Department of Computer Science, Yonsei University, 50, Yonsei-Ro, Seodaemum-Gu, Seoul 120-749, Republic of Korea, E-mail: `emmous@yonsei.ac.kr`

[b] Department of Mathematics and Statistics and Turku Centre for Computer Science, University of Turku, 20014 Turku, Finland, E-mail: `salomaaenator@gmail.com`

[c] School of Computing, Queen's University, Kingston, Ontario K7L 2N8, Canada, E-mail: `ksalomaa@cs.queensu.ca`

DOI: 10.14232/actacyb.23.1.2017.9

Yu and co-authors have considered also the state complexity of combined operations and in a sequence of papers culminating with [6] have determined the precise worst-case state complexity of all combinations of two basic language operations. Establishing the precise state complexity of combined language operations is often quite involved, and for general combinations of operations that include marked concatenation and intersection the question is even undecidable [45]. Ésik et al. [8] have introduced techniques to estimate the state complexity of combined operations.

Ambiguity is a fundamental concept in grammar derivations. The ambiguity of regular expressions and finite state machines was first systematically considered by Book et al. [3]. A regular expression is unambiguous if it denotes each string in at most one way. A nondeterministic finite automaton (NFA) is unambiguous if each string has at most one accepting computation. Book et al. [3] show that the Glushkov automaton construction preserves ambiguities of a regular expression. A more restrictive notion of one-unambiguity was introduced by Brüggemann-Klein and Wood [4]: every regular language can be denoted by an unambiguous regular expression but not, in general, by a one-unambiguous regular expression.

The degree of ambiguity of an NFA A on a string w is the number of accepting computations of A on w. The degree of ambiguity of A is the maximal degree of ambiguity of A on any input string, if the maximum exists, and in this case A is said to be finitely ambiguous. Otherwise the degree of ambiguity of A can be measured as a function of the length of the inputs. Ravikumar and Ibarra [42] have first studied systematically the size trade-offs between the unambiguous, finitely ambiguous, polynomially ambiguous and exponentially ambiguous NFAs. The celebrated separation result of Leung [30] establishes that there exist (exponentially ambiguous) n-state NFAs such that any equivalent polynomially ambiguous NFA needs $2^n - 1$ states. Hromkovič et al. [19, 20] have used powerful techniques from communications complexity for state complexity separations for NFAs with different degree of ambiguity.

The degree of ambiguity is defined in terms of the number of accepting computations, and does not directly limit the amount of nondeterminism, or the amount of guessing, used by an automaton. In an unambiguous NFA, even though an accepting computation is unique, the computation may include any number of nondeterministic steps – unambiguity implies just that at any nondeterministic step at most one choice can lead to acceptance. In order to develop a quantitative understanding of the power of nondeterminism, one can directly measure the number of nondeterministic steps used by an NFA.

Nondeterminism measures for Turing machine computations were originally considered by Kintala and Fischer [25]. Kintala and Wotschke [26] first quantified the amount of nondeterminism in a finite automaton computation and showed, roughly speaking, that there is a significant difference in the determinization size blow-up between NFAs allowing different finite numbers of nondeterministic choices in a computation (where the number of nondetermistic steps is at most the logarithm of the number of states). The hierarchy result has been refined in the spectrum result of Goldstine et al. [11] that will be discussed in section 4.3.

Commonly used nondeterminism measures count the number of nondeterministic steps (or the amount of guessing in bits of information) on a best accepting computation [11], or the number of leaves of the entire computation tree [19, 38]. Further variants limit the amount of nondeterminism on a worst computation [19, 39]. Some interesting relationships between the degree of ambiguity and various nondeterminism measures have been established by Goldstine et al. [12] and Hromkovič et al. [19].

This paper surveys work on the growth rates of the degree of ambiguity and the various nondeterminism measures, and on algorithms to determine the growth rate for a given NFA. In particular, we focus on state completity comparisons between NFAs having different degrees of ambiguity or allowing different amounts of nondeterminism. Strong separation results are known for succinctness comparisons between NFAs of different ambiguity growth rates (finite, polynomial or exponential). However, in the case of limited nondeterminism, practically all existing work on state complexity is restricted to comparisons between different finite amounts of nondeterminism, that is, the amount of nondeterminism on any input is at most a given constant. State complexity of NFAs with limited nondeterminism that is measured as a function of input length is a topic for future study.

First we fix some notation in section 2. Work on the degree of ambiguity is described in section 3 and section 4 deals with the various nondeterminism measures for NFAs.

2 Definitions

Here we recall and introduce some basic notation and definitions. More information on finite automata and regular languages can be found e.g. in [44, 47, 51]. General background on degrees of ambiguity and limited nondeterminism for finite automata can be found in [10, 13, 16, 41].

The set of positive integers is $\mathbb{N}$ and the cardinality of a finite set F is $|F|$. The set of strings over a finite alphabet Σ is Σ^* and ε is the empty string. A *bounded language* is a subset of $a_1^* a_2^* \cdots a_k^*$, where a_i, $1 \leq i \leq k$, are (not necessarily distinct) elements of the alphabet Σ.

A *nondeterministic finite automaton* (NFA) is a tuple $A = (Q, \Sigma, \delta, q_0, F)$ where Q is the finite set of states, Σ is the input alphabet, $\delta : Q \times \Sigma \to 2^Q$ is the transition function, $q_0 \in Q$ is the initial state and $F \subseteq Q$ is the set of final states. The transition function δ is in the usual way extended as a function $Q \times \Sigma^* \to 2^Q$ and the language recognized by A is $L(A) = \{w \in \Sigma^* \mid \delta(q_0, w) \cap F \neq \emptyset\}$. If $|\delta(q, b)| \leq 1$ for all $q \in Q$ and $b \in \Sigma$, the automaton A is a *deterministic finite automaton* (DFA). Note that we allow DFAs to have undefined transitions.

It is well known that DFA's and NFA's both recognize the class of regular languages. For a regular language L, the state complexity of L (respectively, the nondeterministic state complexity of L) is the number of states of the state minimal DFA (respectively, of a state minimal NFA[1]) recognizing L.

[1] An NFA with the smallest number of states recognizing a language L need not be unique.

Consider an NFA $A = (Q, \Sigma, \delta, q_0, F)$. The *branching* of a transition from state $q \in Q$ on input symbol $b \in \Sigma$ is $|\delta(q, b)|$. A *computation* of A on a string $w = b_1 b_2 \cdots b_k$, $b_i \in \Sigma$, $i = 1, \ldots, k$, $k \geq 0$, is a sequence of states $(p_1, \ldots, p_\ell)$, where $p_1 \in \delta(q_0, b_1)$, $p_{j+1} \in \delta(p_j, b_{j+1})$, $j = 1, \ldots \ell - 1$, and either $\ell = k$, or, $\ell < k$ and $\delta(p_\ell, b_{\ell+1}) = \emptyset$.

The sequence of states $(p_1, \ldots, p_\ell)$ is a *complete computation* on $b_1 b_2 \cdots b_k$ if $\ell = k$ and an *accepting computation* is a complete computation that ends in an accepting state of F. The set of all computations (respectively, all accepting computations) of A on the string w is denoted $\mathrm{comp}_A(w)$ (respectively, $\mathrm{comp}_A^{\mathrm{acc}}(w)$).

Intuivively, a computation of A on a string w is a sequence of states that A reaches when started with the initial state and the symbols of w are read one by one. A complete computation ends with a state reached after consuming all symbols of w. A computation may also end with a state where the transition on the next symbol of w is undefined.

3 Degree of ambiguity

Book et al. [3] first considered systematically the ambiguity of regular expressions and NFAs, and the relationship between these notions. A regular expression is unambiguous if it denotes each string in at most one way. A more restrictive notion of 1-unambiguity, or 1-determinism, was introduced by Brüggemann-Klein and Wood [4]. A regular expression is 1-unambiguous if its position automaton is deterministic. Every regular language has an unambiguous regular expression but the 1-unambiguous expressions define a strict subclass of regular languages [4, 14].

An NFA is unambiguous if any string has at most one accepting computation. Formally, the degree of ambiguity of an NFA A on a string w, $\mathrm{da}_A(w)$, is the number of accepting computations of A on w. The *degree of ambiguity* of A on strings of length m is defined as

$$\mathrm{da}_A(m) = \max\{\mathrm{da}_A(w) \mid w \in \Sigma^m\}.$$

Strictly speaking, we use the symbol da_A to denote two different functions: it denotes a function $\Sigma^* \to \mathbb{N}$ and a function $\mathbb{N} \to \mathbb{N}$.

The degree of ambiguity of A is said to be finite (or bounded) if the values $\mathrm{da}_A(m)$, $m \in \mathbb{N}$ are bounded, and in this case we denote

$$\mathrm{da}_A^{\sup} = \sup_{m \in \mathbb{N}} \mathrm{da}_A(m).$$

The NFA A is *unambiguous* if $\mathrm{da}_A^{\sup} = 1$. Clearly every DFA is unambiguous.

Following Ravikumar and Ibarra [42], with respect to the degree of ambiguity we consider five different classes of NFAs: DFAs, unambiguous NFAs (UFA), finitely ambiguous NFAs (FNFA), polynomially ambiguous NFAs (PNFA) and general (potentially exponentially ambiguous) NFAs. An NFA A is strictly polynomially ambiguous if A is not finitely ambiguous and there is a polynomial $p(\cdot)$ such that $\mathrm{da}_A(m) \leq p(m)$ for all $m \in \mathbb{N}$. The polynomial degree of growth of A is the minimal degree of a polynomial $p'(m)$ that upper bounds the function $\mathrm{da}_A(m)$. An NFA A is strictly exponentially ambiguous if it is not polynomially ambiguous.

It is known that for a fixed $k \in \mathbb{N}$ the equivalence of FNFAs with degree of ambiguity k can be tested in polynomial time [27, 49]. This is significant because, as we will see, the determinization of even UFAs can cause an exponential size blow-up. Also, different variants of the minimization problem for UFAs remain intractable, see [16] for references.

The syntactic definition of an NFA A does not directly tell us what is the degree of ambiguity of A. It was shown by Mandel and Simon [33] and Reutenauer [43], and by others independently, that it is decidable whether a given NFA is finitely ambiguous or polynomially ambiguous. Reutenauer [43] also gave an algorithm to compute the polynomial degree of growth of an NFA.

Building on charaterizations by Ibarra and Ravikumar [21] and Reutenauer [43], Weber and Seidl [50] gave a simpler structural characterization of finitely ambiguous and polynomial ambiguous NFAs that yields a polynomial time algorithm for the corresponding decision problems. The characterization implies also that, for an NFA with unbounded ambiguity, the degree of ambiguity must grow at least linearly.

Theorem 1 (Weber and Seidl [50]). *It can be decided in polynomial time whether a given NFA A is finitely ambiguous, strictly polynomially ambiguous or strictly exponentially ambiguous. Furthermore, the polynomial degree of growth of A can be computed in polynomial time.*

For the question of determining the exact finite degree of ambiguity, the complexity depends essentially on whether the finite degree of ambiguity is a constant or considered part of the input. For a fixed k, it can be tested in polynomial time whether the degree of ambiguity of an NFA is greater than k [49], but when k is part of the input the complexity is essentially worse.

Theorem 2 (Chan and Ibarra [5]). *For a given NFA A and $k \in \mathbb{N}$, testing whether the degree of ambiguity of A is at least k is PSPACE-complete.*

A relevant question is also how large can be the degree of ambiguity of an n-state FNFA. A double exponential upper bound was given already by Mandel and Simon [33] and this was impoved to $2^{\Theta(n^3)}$ by Reutenauer [43]. The bound was further improved by Weber and Seidl [50] who also show that for some subclasses of NFAs the maximal finite degree of ambiguity is exactly $2^{\Theta(n)}$.

Theorem 3 (Weber and Seidl [50]). *The degree of ambiguity of an n-state FNFA is at most $5^{\frac{n}{2}} \cdot n^n$.*

3.1 Ambiguity and state complexity

Clearly every regular language can be recognized by an unambiguous NFA, but the succinctness of the description depends significantly on the degree of ambiguity. Schmidt [46] first developed methods to prove lower bounds for the size of UFAs and also showed that the determinization of UFAs causes, in the worst case, an exponential size blow-up. The lower bound was improved by different authors and the precise worst case size blow-up was determined by Leung [32]. Leiss [29] constructed n-state UFAs with multiple initial states where any equivalent DFA needs $2^n - 1$ states and Leung [32] gave a construction for the same exponential size blow-up using UFAs with only one initial state.

Theorem 4 (Leiss [29], Leung [32])**.** *For each $n \in \mathbb{N}$, there exists an UFA with n states such that the minimal equivalent DFA has $2^n - 1$ states. For each $n \in \mathbb{N}$, there exists an FNFA with n states such that any equivalent UFA has $2^n - 1$ states.*

Note that because our definition allows DFAs to be incomplete the above bound for the UFA determinization differs by one from the bound stated in [32]. Input-driven pushdown automata (IDPDA) define a subclass of deterministic context-free languages that retains many of the desirable properties of the regular languages. In particular, an n-state nondeterministic IDPDA has an equivalent deterministic machine with $2^{\Theta(n^2)}$ states [1]. Recently, Okhotin and Salomaa [36] have shown that, analogously with Theorem 4, determinizing an unambiguous IDPDA and converting a general nondeterministic IDPDA to an unambiguous one both cause, in the worst case, the same $2^{\Theta(n^2)}$ size blow-up.

For state complexity comparisons between NFAs with different growth rates of ambiguity we use the following terminology. Consider classes X and Y of devices (the classes we consider are DFAs, UFAs, FNFAs, PNFAs and general NFAs, possibly with additional restictions). We say that class Y is *(super-polynomially) separated* from class X, if there exists a collection of languages L_n, $n \in \mathbb{N}$, such that L_n is recognized by a device from class Y having n states, but for any polynomial $p(n)$ and for sufficiently large values of n, a device from class X for L_n must have more than $p(n)$ states. This means, roughly speaking, that simulation of devices of class Y by devices of class X, in the worst case, causes a super-polynomial size blow-up.

Ravikumar and Ibarra [42] first considered systematically succinctness comparisons between FNFAs, PNFAs and general NFAs. In particular, they established the following result for NFAs accepting bounded languages.

Theorem 5 (Ravikumar and Ibarra [42])**.** *Any NFA accepting a bounded language can be converted to an FNFA with at most polynomial size blow-up. The class of FNFAs (respectively, the class of UFAs) recognizing a bounded language is super-polynomially separated from the corresponding class of UFAs (respectively, of DFAs).*

The descriptional complexity comparison between the classes FNFA, PNFA and NFA recognizing general regular languages was left open in [42]. Although in the case of bounded languages, NFAs of exponential ambiguity can be simulated by PNFAs and FNFAs of polynomial size, it was conjectured that for general regular languages the classes are super-polynomially separated. Leung [30] and Hromkovič et al. [19] have established that general NFAs can be super-polynomially more succinct than PNFAs.

Theorem 6 (Leung [30], Hromkovič et al. [19]). *The class of NFAs is super-polynomially separated from the class of PNFAs.*

The communication complexity techniques used by Hromkovič et al. [19] to prove Theorem 6 yield a substantially simplified proof. However, their proof does not give the optimal size blow-up $2^n - 1$ for the NFA–to–PNFA transformation that is obtained in the original ad hoc proof where Leung [30] shows that any PNFA for the family of languages $L_n = (0 + (01^*)^{n-1}0)^*$, $n \geq 1$, cannot be smaller than an incomplete DFA. It is easy to give an n-state NFA of exponential ambiguity that recognizes L_n.

Ravikumar and Ibarra [42] also conjectured that polynomially ambiguous NFAs can be significantly more succinct than finitely ambiguous NFAs. This question remained open for over 20 years. After about 10 years Hromkovič et al. [19] gave a partial result showing that there exist $(n+2)$-state PNFAs (with linear degree of ambiguity) such that any equivalent FNFA with degree of ambiguity k must have at least $2^{\frac{n-2}{k}} - 2$ states. The question was solved affirmatively by Hromkovič and Schnitger [20] using the powerful communication complexity techniques.

Theorem 7 (Hromkovič and Schnitger [20]). *For $n \in \mathbb{N}$ there exists PNFA A with number of states polynomial in n such that any FNFA recognizing the language $L(A)$ has at least $2^{\Omega(n^{\frac{1}{3}})}$ states.*

Theorem 7 is obtained as a special case of the more technical statement given next in Theorem 8 by setting there the parameter k to be one and, in fact, the degree of ambiguity of the PNFA A is only linear. The general result by Hromkovič and Schnitger [20] gives a super-polynomial succinctness separation between NFAs with degree of ambiguity $O(m^k)$ and $O(m^{k-1})$, $k \in \mathbb{N}$.

Theorem 8 (Hromkovič and Schnitger [20]). *Let r and $t = (r/k^2)^{\frac{1}{3}}$ be positive integers. There exist languages $L_{r,k}$ having an NFA with degree of ambiguity $O(m^k)$ and $k \cdot \mathrm{poly}(r)$ states such that any NFA for $L_{r,k}$ with degree of ambiguity $o(m^k)$ has at least $2^{\Omega(r^{(\frac{1}{3}}/k^{\frac{5}{3}})}$ states.*

Theorem 7 and Theorem 8 give a super-polynomial separation, respectively, between PNFAs and FNFAs and between NFAs having different polynomial degree of growth for ambiguity. The statement of Theorem 8 defines the languages $L_{r,k}$ only for restricted values of the subindices, but for the separation result it is sufficient that $L_{r,k}$ exists for infinitely many values of r and k. However, the lower bounds are not of the order $2^{\Theta(n)}$ as is known in the separation of general NFAs and PNFAs [30]. In fact, Hromkovič and Schnitger [20] suspect that the lower bound of Theorem 8 may not be optimal even for the languages used in the lower bound construction.

To conclude this section, we mention that Okhotin [35] has studied the state complexity of determinization of unary UFAs and Jirasek et al. [22] recently studied the state complexity of operations on UFAs.

4 Limited nondeterminism

Nondeterminism measures can be based on the amount of nondeterminism used in a best accepting computation of an NFA A on a given string w, on the amount of nondeterminism in a worst computation of A on w or on the size of the computation tree of A on w [10, 11, 19, 38].

In the following, $A = (Q, \Sigma, \delta, q_0, F)$ is always an NFA. Consider a string $w = b_1 b_2 \cdots b_k$, $b_i \in \Sigma$, $i = 1, \ldots, k$, and a computation of A on w,

$$C = (p_1, \ldots, p_\ell), \;\; p_i \in Q, 1 \le i \le \ell \le k.$$

Recall that $\ell < k$ is possible only if $\delta(p_\ell, b_{\ell+1}) = \emptyset$, that is, a computation reads the entire string w unless it encounters an undefined transition.

The *guessing* of the computation C, $\gamma_A(C)$ [11], is

$$\gamma_A(C) = \log_2 |\delta(q_0, b_1)| + \sum_{i=1}^{\ell-1} \log_2 |\delta(p_i, b_{i+1})|.$$

The branching of the first step of the computation C is $|\delta(q_0, b_1)|$, and after the first step the state is p_1. The branching of the second step is then $|\delta(p_1, b_2)|$, and the branching of the ith step is $|\delta(p_{i-1}, b_i)|$, $3 \le i \le \ell - 1$. Thus, intuitively, $\gamma_A(C)$ represents the amount of guessing, in bits of information, that occurs during the computation C. If A is a DFA, the amount of guessing in any computation of A is zero.

The *branching of the computation C*, $\beta_A(C)$ [11], is defined as the product of the branchings of the individual transitions of C, or in other words, $\beta_A(C) = 2^{\gamma_A(C)}$.

The amount of guessing an NFA uses on a string can be defined either as a best case or a worst case measure. The *guessing of a string* $w \in L(A)$ [11] is the amount of guessing of the best accepting computation:

$$\gamma_A(w) = \min\{\; \gamma_A(C) \mid C \in \mathrm{comp}_A^{\mathrm{acc}}(w) \;\},$$

and the *maximum guessing* of A on a string $w \in \Sigma^*$ [39] is

$$\gamma_A^{\max}(w) = \max\{\; \gamma_A(C) \mid C \in \mathrm{comp}_A(w) \;\}.$$

Note that the best case measure is defined as the amount of guessing on the best accepting computation while the maximum guessing considers all, not necessarily complete, computations. Instead of counting the amount of guessing in bits of information, Hromkovič et al. [19] use the *advice measure* that counts the number of nondeterministic steps on the worst computation on a given input and Leung [31] uses a corresponding best case measure. These measures are within a multiplicative constant (depending only on the NFA A) of the $\gamma_A^{\max}$ and γ_A measures, respectively.

The *branching* (respectively, the *trace*) of A on the string w is then $\beta_A(w) = 2^{\gamma_A(w)}$ (respectively, $\tau_A(w) = 2^{\gamma_A^{\max}(w)}$ [39, 41]).

The total amount of nondeterminism used by A in all computations on a string w is represented by the number of leaves of the computation tree of A on w. The number of leaves is the same as the number of computations of A on w, $|\mathrm{comp}_A(w)|$, and this value is called the *tree width* of A on w, $\mathrm{tw}_A(w)$. The tree width measure is called 'leaf size' in [19].

Similarly as we did with the degree of ambiguity, the tree width, the (maximum) guessing, the branching and the trace of an NFA A defines a function on naturals by taking the maximum value of the measure on strings of length m ($m \in \mathbb{N}$). If χ is any of tw, γ, $\gamma^{\max}$, β, or τ then $\chi_A : \mathbb{N} \to \mathbb{N}$ is defined as

$$\chi_A(m) = \max\{\ \chi_A(w) \mid w \in \Sigma^m\ \},\ \ m \in \mathbb{N}.$$

We say that the χ-function of A is *finite* (or *bounded*) if the value $\chi_A^{\sup} =^{\mathrm{def}} \sup_{m\in\mathbb{N}} \chi_A(m)$ is finite.

Hromkovič et al. [19] have characterized the possible growth rates of the tree width of an NFA. As for degree of ambiguity, the tree width of an NFA cannot be unbounded and sublinear.

Theorem 9 (Hromkovič et al. [19]). *For any NFA A, the function $\mathrm{tw}_A(m)$ is either bounded by a constant, or between linear and polynomial in m, or otherwise in $2^{\Theta(m)}$.*

The above characterization can be effectively decided. An NFA A has unbounded tree width if and only if some cycle of A contains a nondeterministic transition and this observation yields a simple polynomial time algorithm to test whether $\mathrm{tw}_A(m)$ is bounded [38]. On the other hand, there is no efficient algorithm to determine whether the guessing of an NFA is bounded.

Theorem 10 (Leung [31]). *For a given NFA A, it is PSPACE-complete to decide whether $\gamma_A(m)$ is bounded.*

Interestingly it is known that the guessing of an NFA may be unbounded and grow sublinearly.

Theorem 11 (Simon [48], Goldstine et al. [12]). *For each $k \in \mathbb{N}$, there is an NFA A such that $\gamma_A(m) = \Theta(\sqrt[k]{m})$.*

Due to the exponential correspondence between the guessing and branching measures, Theorem 11 implies that, for each $k \in \mathbb{N}$, there exists an NFA A such that $\beta_A(m) = 2^{\Theta(\sqrt[k]{m})}$. It is not known whether the branching of an NFA can be polynomially bounded but infinite [39].

Open 1. *If NFA A has unbounded branching does this imply that the growth rate of $\beta_A(m)$ must be superpolynomial?*

It is known that, for a unary NFA A, $\beta_A(m)$ is always either bounded or in $2^{\Theta(m)}$ [41] and the possible growth rates of a variant of the branching measure considered in [37] are similarly restricted. For the worst-case branching measure trace, Palioudakis et al. [39] have shown that, for an n-state NFA A, $\tau_A(m)$ is either bounded or $\tau_A(m) \geq 2^{\lfloor \frac{m}{n} \rfloor}$.

4.1 NFAs with large finite nondeterminism

Similarly as in the case of degree of ambiguity [50], for an n-state NFA A with bounded guessing (respectively, bounded tree width) we can ask how large can the guessing (respectively, the tree width) of A be. Leung [31] has shown that an n-state NFA with limited nondeterminism in any computation can make at most $2^n - 2$ nondeterministic transitions and has constructed a family of NFAs with bounded nondeterminism that is considerably larger than the number of states.

Theorem 12 (Leung [31])**.** *If A is an n-state NFA with bounded guessing, then $\gamma_A(m) = O(2^n)$. There exist n-state NFAs B_n, $n \in \mathbb{N}$ such that $\gamma_{B_n}^{\sup} = 2^{\frac{n}{3}} - 2$.*

A limitation of the above result is that in the NFAs B_n are defined over a growing alphabet and a large number of the nondeterministic moves are redundant. It remains open whether there exist n-state NFAs A with bounded guessing that is larger than n and where the language $L(A)$ cannot be recognized by an NFA of same size and less nondeterminism [31]. The notion of “less nondeterminism” could be formalized analogously as is done below with the notion of optimality in the case of tree width.

Hromkovič et al. [19] observed that the tree width of an n-state NFA, if bounded, is at most n^n. Palioudakis et al. [38] improved this bound and, furthermore, gave a construction of n-state NFAs with all possible values of bounded tree width that do not have “redundant” nondeterminism.

The notion of avoiding redundant nondeterminism is formalized as follows. A finite tree width NFA A with n states is said to have *optimal tree width* if $L(A)$ cannot be recognized by any NFA B with n_1 states where $n_1 \leq n$ and $\mathrm{tw}_B^{\sup} \leq \mathrm{tw}_A^{\sup}$ and at least one of the inequalities is strict.

Theorem 13 (Palioudakis et al. [38])**.** *The tree width of an n-state finite tree width NFA is at most 2^{n-2}. For every $n \geq 2$ and $1 \leq k \leq 2^{n-2}$ there exists an n-state NFA over a binary alphabet having optimal tree width k.*

Note that the above bound is less than the upper bound for the finite ambiguity of an n-state NFA (from Theorem 3). Naturally, for any NFA A and string w, the degree of ambiguity of A on w is at most the tree width of A on w (and usually much smaller than the tree width). However, an upper bound for the finite tree width of an n-state NFA does not imply a corresponding bound for the degree of ambiguity because an NFA may have finite ambiguity and unbounded tree width.

4.2 Comparing nondeterminism measures and ambiguity

Directly from the definitions it follows that if an NFA A has finite tree width, then the guessing (and branching) of A is also finite, but the converse implication does not need to hold. The tree width of A is finite if and only if the trace of A is finite.

Proposition 1 (Palioudakis et al. [39])**.** *If A is an NFA with finite tree width, then*

$$\mathrm{tw}_A^{\sup} \leq \tau_A^{\sup} \leq 2^{\mathrm{tw}_A^{\sup} - 1}.$$

It is known that the above inequalities cannot be improved in general, that is, there are NFAs for which either of the inequalities of Proposition 1 becomes and equality [39].

Hromkovič et al. [19] have established relationships between the tree width, maximum guessing and degree of ambiguity in a minimal NFA. They use the name 'leaf size' for tree width and instead of maximum guessing they use an "advice" measure that is within a constant factor of maximum guessing. The advice of an NFA A on a string w counts the largest number of nondeterministic steps in any computation of A on w.

Theorem 14 (Hromkovič et al. [19])**.** *If A is a minimal NFA, then for all $m \in \mathbb{N}$,*

$$\max(\gamma_A^{\max}(m), \mathrm{da}_A(m)) \leq \mathrm{tw}_A(m) = O(\mathrm{da}_A(m) \cdot \gamma_A^{\max}(m)).$$

Goldstine et al. [12] have established a subtle relationship between ambiguity and guessing for NFAs where all states are final. They define the ambiguity of a string w as the number of complete computations on w. To avoid confusion, we call the number of complete computations of an NFA A on a string w the *complete ambiguity*[2] of A on w. Note that if all states of A are final, the complete ambiguity of A coincides with the degree of ambiguity as defined in section 3 and if A has no undefined transitions then the complete ambiguity of A coincides with the tree width of A.

By definition, the guessing function $\gamma_A(m)$ of an NFA grows at most linearly. If the guessing is bounded or grows linearly, then the complete ambiguity may be either bounded or unbounded but, in the intermediate case, where the guessing is unbounded but sublinear, then ambiguity must always be unbounded. Recall from Theorem 11 that there exist NFAs with unbounded and sublinear growth rate of the guessing function.

Theorem 15 (Goldstine et al. [12])**.** *Let A be an NFA. If $\gamma_A(m)$ is non-constant and sublinear, then the complete ambiguity of A must be unbounded. On the other hand, if $\gamma_A(m)$ is in $O(1)$ or in $\Theta(m)$, then the complete ambiguity may be either bounded or unbounded.*

4.3 Limited nondeterminism and state complexity

An important descriptional complexity question is the succinctness comparison of NFAs employing different amounts of nondeterminism and, in particular, the determinization size blow-up of NFAs with limited nondeterminism. Goldstine et al. [11] have shown that converting a general NFA to an NFA with finite branching involves, in the worst case, an exponential size blow-up.

Theorem 16 (Goldstine et al. [11])**.** *For each $n \in \mathbb{N}$, there exists an n-state NFA A such that any finite branching NFA recognizing the language $L(A)$ needs at least 2^{n-1} states.*

[2] Keeler [24] calls this the string path width of A on w.

Also, Goldstine et al. [11] have shown that there exist regular languages for which different finite amounts of nondeterminism yield incremental savings in the number of states. The following two theorems give the "spectrum" result of [11] stated in a slightly simplified form.

Theorem 17 (Goldstine et al. [11])**.** *Let A be a minimal DFA of size $2^n - 1$, $n \geq 2$. Then*

(i) for $2 \leq k \leq \frac{n}{\log_2 n}$, the optimal size of an NFA with branching k for $L(A)$ is at least $2^{\frac{n}{k}}$, and,

(ii) for $k \geq \frac{n}{\log_2 n}$, an NFA with branching k for $L(A)$ has size at least n.

Furthermore, they show that the bounds of Theorem 17 are close to best possible:

Theorem 18 (Goldstine et al. [11])**.** *For $n \geq 2$ there exists a minimal NFA A_n with $n+1$ states such that if we denote by $\sigma_n[k]$ the optimal size of an NFA with branching k recognizing $L(A_n)$ then the following relations hold:*

(i) $\sigma_n[1] = 2^n$, and, $2^{\frac{n}{k}} \leq \sigma_n[k] < 2k \cdot 2^{\frac{n}{k}}$ when $2 \leq k < \frac{n}{\log_2 n}$,

(ii) $n + 1 \leq \sigma_n[k] < 2k \cdot 2^{\frac{n}{k}}$ when $\frac{n}{\log_2 n} \leq k < n$,

(iii) $n + 1 \leq \sigma_n[k] < 4n$, when $k \geq n$.

Recall that tree width is more restrictive than branching in the sense that an NFA with finite tree width necessarily has finite branching, but the converse implication does not hold, in general. Contrasting the result of Theorem 16, every finite tree width NFA has an equivalent DFA of polynomial size.

Theorem 19 (Palioudakis et al. [38])**.** *For an NFA A with n states having tree width at most $k \leq n - 1$, the language $L(A)$ has a DFA of size $1 + \sum_{j=1}^{k} \binom{n-1}{j}$. Furthermore, for every $1 \leq k \leq n - 1$, there exists an n state NFA $A_{n,k}$ with tree width k over a binary alphabet such that the minimal DFA for $L(A_{n,k})$ has $1 + \sum_{j=1}^{k} \binom{n-1}{j}$ states.*

Palioudakis et al. [38] gives also an upper bound $1+\sum_{i=1}^{k-\ell+1} \binom{n-1}{i}$ for converting an n state NFA with tree width k to an NFA with tree width $2 \leq \ell < k$, but a corresponding lower bound is missing. Also no spectrum result for tree width analogous to the spectrum result for branching (Theorem 18) is known. That is, there is no result that yields good bounds, for a given sequence of languages, for the succinctness of NFAs over a range of different tree width values.

Deterministic finite automata with multiple initial states (MDFA) can be viewed as a restricted type of automata with limited nondeterminism: the only nondeterminism consists of the choice of the initial state. With an elegant construction based on modular arithmetic, Kappes [23] has given an efficient simulation of an NFA with finite branching by an MDFA.

Theorem 20 (Kappes [23]). *An NFA with n states and branching k ($k \in \mathbb{N}$) can be simulated by an MDFA with $k \cdot n$ states and k initial states.*

The bound as stated in [23] is $k \cdot n + 1$ and the construction produces an MDFA with a dead state. Above in Theorem 20 we allow the possibility that an MDFA can have undefined transitions (following the definition of [40]). Palioudakis et al. [40] have established an almost matching lower bound by showing that for infinitely many values $n, k \in \mathbb{N}$ there exists an n-state NFA with branching k such that any equivalent MDFA needs at least $\frac{k}{1+\log k} \cdot n$ states.

To conclude we mention that limiting the nondeterminism of an NFA is not sufficient to make the minimization problem tractable. It is well known that minimization of general NFAs is PSPACE-complete. Björklund and Martens [2] have shown that minimization remains NP-hard, roughly speaking, for all finite automaton models that extend the class of DFAs. The hardness result is for the class of δNFAs which are a very restricted subclass of tree width two NFAs [2].

5 Conclusion and open problems

Descriptional complexity comparison of nondeterministic finite automata of different degrees of ambiguity and employing different amounts of nondeterminism is a foundational question in automata theory. The spectrum result of Goldstine et al. [11] (Theorems 17 and 18) establishes the existence of a sequence of languages for which different finite amounts of branching allow incremental savings in the number of states, and the succinctness comparisons are approximately the best possible.

On the other hand, very little is known about the state complexity of NFAs where the amount of nondeterminism is unbounded and measured as a function of input length. While there is a super-polynomial separation between the size of finitely ambiguous, polynomially ambiguous, and general NFAs, no succinctness comparisons between NFAs of different unbounded branching or unbounded tree width are known. This can be a topic for future research. In particular, it would be interesting to know whether the powerful communication complexity techniques used by Hromkovič et al. [19, 20] for succinctness comparisons of NFAs with different degrees of ambiguity can be used to establish good lower bounds and separation results for the state complexity of NFAs where the branching (or the tree width) is measured as a function of input length.

A further topic of interest could be the succinctness comparison of NFAs of given degree of ambiguity and NFAs of given branching (or tree width). Only a few tentative results are known [38] in this direction.

Acknowledgements

Han was supported by the Basic Science Research Program through NRF funded by MEST (2015R1D1A1A01060097) and the Yonsei University Future-leading Research Initiative of 2016. K. Salomaa was supported by the Natural Sciences and Engineering Research Council of Canada Grant OGP0147224.

References

[1] Alur, R. and Madhusudan, P. Adding nesting structure to words. *Journal of the ACM,* 56(3), 2009.

[2] Björklund, H. and Martens, W. The tractability frontier for NFA minimization. *J. Comput. System Sci.,* 78: 198–210, 2012.

[3] Book, R., Even, S., Greibach, S. and Ott, G. Ambiguity in graphs and expressions. *IEEE Transactions on Computers,* c-20(2):149–153, 1971.

[4] Brüggemann-Klein, A. and Wood, D. One-unambiguous regular languages. *Information and Computation,* 140(2):229–253, 1998.

[5] Chan, T.-H. and Ibarra, O. On the finite-valuedness problem for sequential machines. *Theoret. Comput. Sci.,* 23: 95–101, 1983.

[6] Cui, B., Gao, Y., Kari, L., Yu, S. State complexity of combined operations with two basic operations. Theoret. Comput. Sci., 437: 82–102, 2012.

[7] Ellul, K., Krawetz, B., Shallit, J. and Wang, M.-W. Regular expressions: New results and open problems. *Journal of Automata, Languages and Combinatorics,* 10: 407–437, 2005.

[8] Ésik, Z., Gao, Y, Liu, G., Yu, S. Estimation of state complexity of combined operations. *Theoret. Comput. Sci.,* 410: 3272–3280, 2009.

[9] Gao, Y., Moreira, N., Reis, R., Yu, S. A review on state complexity of individual operations. To appear in *Computer Science Review.* (Available at `www.dcc.fc.up.pt/dcc/Pubs/TReports/TR11/dcc-2011-08.pdf`)

[10] Goldstine, J., Kappes, M., Kintala, C.M.R., Leung, H., Malcher, A. and Wotschke, D. Descriptional complexity of machines with limited resources. *J. Universal Computer Science,* 8(2): 193–234, 2002.

[11] Goldstine, J., Kintala, C.M.R. and Wotschke, D. On measuring nondeterminism in regular languages. *Inform. Comput.,* 86: 179–194, 1990.

[12] Goldstine, J., Leung, H. and Wotschke, D. On the relation between ambiguity and nondeterminism in finite automata. *Inform. Comput.,* 100: 261–270, 1992.

[13] Gruber, H., Holzer, M. From finite automata to regular expressions and back — A summary of descriptional complexity. *Intern. J. Foundations Comput. Sci.,* 26: 1009–1040, 2015.

[14] Han, Y.-S. and Wood, D. Generalizations of 1-deterministic regular languages. *Information and Computation,* 206(9-1): 1117–1125, 2008.

[15] Holzer, M. and Kutrib, M. Nondeterministic finite automata – Recent results on the descriptional and computational complexity. *Intern. J. Foundations Comput. Sci.,* **20**(4): 563–580, 2009.

[16] Holzer, M. and Kutrib M. Descriptional complexity of (un)ambiguous finite state machines and pushdown automata. In: *Reachability Problems 2010,* Lect. Notes Comput. Sci. 6227, pp. 1–23, 2010

[17] Holzer, M. and Kutrib, M. Descriptional and computational complexity of finite automata — A survey. *Information and Computation,* 209: 456–470, 2011.

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

[19] Hromkovič, J., Seibert, S., Karhumäki, J., Klauck, H. and Schnitger, G. Communication complexity method for measuring nondeterminism in finite automata. *Inform. Comput.,* 172: 202–217, 2002.

[20] Hromkovič, J. and Schnitger, G. Ambiguity and communication. *Theory Comput. Syst.,* 48: 517–534, 2011.

[21] Ibarra, O. and Ravikumar, B. On sparseness, ambiguity and other decision problems for acceptors and transducers. In *Proc. of STACS'86,* Lect. Notes Comput. Sci. 210, Springer, pp. 171–179, 1986.

[22] Jirásek, J., Jirásková, G., Šebej, J. Operations on unambiguous finite automata. In *Developments in Language Theory, DLT,* Lect. Notes Comput. Sci. 9840, Springer, pp. 243–255, 2016.

[23] Kappes, M. Descriptional complexity of deterministic finite automata with multiple intial states. *J. Automata, Languages and Combinatorics,* 5: 269–278, 2000.

[24] Keeler, C. *New measures for finite automaton complexity and subregular language hierarchies.* MSc thesis, Queen's University, 2016.

[25] Kintala, C.M.R. and Fischer, P.C. Refining nondeterminism in relativized polynomial time computations. *SIAM J. Comput.,* 9(1): 46–53, 1980.

[26] Kintala, C.M.R and Wotschke, D. Amounts of nondeterminism in finite automata. *Acta Informatica,* 13(2): 199–204, 1980.

[27] Kuich, W. Finite automata and ambiguity. Rept. 253 of the IIG. Techinische Universität Graz, 1988.

[28] Kutrib, M. and Pighizzini, G. Recent trends in descriptional complexity of formal languages. *Bulletin of the EATCS,* 111: 70–86, 2013.

[29] Leiss, E. Succinct representation of regular languages by Boolean automata. *Theoret. Comput. Sci.,* 13: 323–330, 1981.

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

[31] Leung, H. On finite automata with limited nondeterminism. *Acta Informatica,* 35: 595–624, 1998.

[32] Leung, H. Descriptional complexity of NFA of different ambiguity. *Intern. J. Foundations Comput. Sci.,* 16(5): 975–984, 2005.

[33] Mandel, A. and Simon, I. On finite semi-groups of matrices. *Theoret. Comput. Sci.,* 5: 183–204, 1977.

[34] Maslov, A.N. Estimates of the number of states of finite automata. *Soviet Math. Dokl.,* 11: 1373–1375, 1970.

[35] Okhotin, A. Unambiguous finite automata over a unary alphabet. *Inform. Comput.,* 212: 15–36, 2012.

[36] Okhotin, A., Salomaa, K. Descriptional complexity of unambiguous input-driven pushdown automata. *Theoret. Comput. Sci.,* 566: 1–11, 2015.

[37] Palioudakis, A., Han, Y.-S. and Salomaa, K. Growth rate of minimum branching. Submitted for publication, February 2017.

[38] Palioudakis, A., Salomaa, K. and Akl, S.G. State complexity of finite tree width NFAs. *J. Automata, Languages and Combinatorics,* 17(2–4): 245–264, 2012.

[39] Palioudakis, A., Salomaa, K. and Akl, S.G.: Comparisons between measures of nondeterminism for finite automata. In *Proc. of DCFS'13,* Lect. Notes Comput. Sci. 8031, Springer, pp. 217–228, 2013.

[40] Palioudakis, A., Salomaa, K. and Akl, S.G. Lower bound for converting an NFA with finite nondeterminism into an MDFA. *J. Automata, Languages and Combinatorics,* 19: 251–264, 2014.

[41] Palioudakis, A., Salomaa, K. and Akl, S.G. Quantifying nondeterminism in finite automata. *Annals of the University of Bucharest,* No. 2, pp. 89–100, 2015.

[42] Ravikumar, B. and Ibarra, O.H. Relating the type of ambiguity of finite automata to the succinctness of their representation. *SIAM Journal of Computing,* 18(6): 1263–1282, 1989.

[43] Reutenauer, C. Propriétés arithmétiques et topologiques de séries rationnelles en variables non commutatives. *Thése troisiéme cycle,* Université Paris VI, 1977.

[44] Rozenberg, G., Salomaa, A. (Eds.) *Handbook of Formal Languages,* Vol. I–III, Springer-Verlag, 1997.

[45] Salomaa, A., Salomaa, K., and Yu, S. Undecidability of state complexity. *Internat. J. Comput. Math.,* 90: 1310–1320, 2013.

[46] Schmidt, E.M. *Succinctness of descriptions of context-free, regular and finite languages.* PhD thesis, Cornell University, Ithaca, NY, 1978

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

[48] Simon, I. The nondeterministic complexity of a finite automaton. In: Lothaire, M. (ed.) *Mots - mélanges offerts a M.P. Schützenberger,* Paris, Hermes, pp. 384–400, 1990.

[49] Stearns, R. and Hunt III, H. On the equivalence and containment problems for unambiguous regular expressions, regular grammars and finite automata. *SIAM J. Comput.,* 14: 598–611, 1985.

[50] Weber, A. and Seidl, H. On the degree of ambiguity of finite automata. *Theoret. Comput. Sci.,* 88: 325–349, 1991.

[51] Yu, S. Regular languages, in [44], Vol. I, pp. 41–110, 1997.

[52] Yu, S. State complexity of regular languages. *Journal of Automata, Languages and Combinatorics,* 6(2): 221–234, 2001.