Randomization in Non-Uniform Finite Automata

Pavol Ďuriš
Comenius University in Bratislava, Slovakia
duris@dcs.fmph.uniba.sk

Rastislav Královič
Comenius University in Bratislava, Slovakia
kralovic@dcs.fmph.uniba.sk

Richard Královič
Google Inc., Zürich, Switzerland
ri.kralovic@gmail.com

Dana Pardubská
Comenius University in Bratislava, Slovakia
pardubska@dcs.fmph.uniba.sk

Martin Pašen
Comenius University in Bratislava, Slovakia
martin.pasen@fmph.uniba.sk

Peter Rossmanith
RWTH Aachen, Germany
rossmani@cs.rwth-aachen.de

Abstract

The non-uniform version of Turing machines with an extra *advice* input tape that depends on the length of the input but not the input itself is a well-studied model in complexity theory. We investigate the same notion of non-uniformity in weaker models, namely one-way finite automata. In particular, we are interested in the power of two-sided bounded-error randomization, and how it compares to determinism and non-determinism. We show that for unlimited advice, randomization is strictly stronger than determinism, and strictly weaker than non-determinism. However, when the advice is restricted to polynomial length, the landscape changes: the expressive power of determinism and randomization does not change, but the power of non-determinism is reduced to the extent that it becomes incomparable with randomization.

2012 ACM Subject Classification Theory of computation → Automata extensions

Keywords and phrases finite automata, non-uniform computation, randomization

Digital Object Identifier 10.4230/LIPIcs.MFCS.2020.30

Funding This research has been partially supported by the grant 1/0601/20 of the Slovak Scientific Grant Agency VEGA.

1 Introduction

Computational models traditionally employed in algorithmics and formal language theory (e. g. Turing machines, automata, ...) are usually uniform in the sense that a single, finitely described, device is used to process an infinite number of words in a given language L. On the other hand, complexity theory often studies non-uniform models (e. g. circuits) where a separate device is considered for each slice $L \cap \Sigma^n$.

Since the uniformity of the model seems to have a significant impact on the complexity of computational problems, there has been a lot of effort in comparing the uniform and non-uniform classes. For one way, the inherently non-uniform models can be made uniform by requiring that, for each n, the device used to process the slice $L \cap \Sigma^n$ must be generated

by some resource-constrained Turing machine on input 1^n (see e. g. [2]). For the other way, one can consider non-uniform versions of models that are inherently uniform by allowing a family of devices, each used to process a single slice $L \cap \Sigma^n$. Since the Turing machine model is powerful enough to contain the universal machine, this is equivalent to the standard definition of Karp and Lipton [11], where the input word is prefixed with an advice string that depends only on the length of the input (which may be the description of the TM used to process this particular slice). Note that since every slice is a finite language, non-uniform Turing machines can recognize, with sufficient advice, all languages. One then studies which languages can be recognized with advice of limited size.

We are interested in non-uniform versions of simpler models, namely finite automata. Since finite automata are much more limited when compared to Turing machines (in particular, they don't possess a universal machine), the way how the non-uniform version is defined matters, and the various definitions are not equivalent.

Ibarra and Ravikumar [9] considered families of finite automata with bounded rate of growth of their state complexity. Damm and Holzer [3] used a definition analogous to [11] where the input word is prefixed by an advice string. Obviously, in this model only advice of constant size is relevant, but there still is a hierarchy with increasing advice size (e. g. unary languages are trivially recognized with 1-bit advice). In order to utilize non-constant advice, the advice string must be made accessible to the automaton during various moments in the computation. Tadaki et al. [16] considered the advice written in a separate track (i. e., one advice symbol was assigned to every symbol of the input word). Freivalds [7] introduced a model where the advice may be split into several tapes (and the measure is the sum of the lengths of all advice tapes) with a prefix property, i. e., on each tape the string used as advice for words of length n must be a valid advice for all words of lengths up to n. Finally, Küçük et al. [13] introduced the model we use, with the advice written on a dedicated tape.

The introduction of non-uniformity into finite automata dramatically changes their expressive power, and non-uniform versions of different types of automata which are equivalent in the uniform version have different expressiveness. For example, one-way non-deterministic automata (NFA) can recognize any language with sufficient advice, and the same holds also for two-way deterministic automata (DFA), or one-way deterministic automata with two advice tapes [13]. On the other hand, the language $L_{ww} = \{ww \mid w \in \{a, b\}^\star\}$ cannot be recognized by a one-way DFA with any advice [4]. It has also been known [4] that for any growing function f, there is a language that can be recognized by a one-way NFA with advice $O(f(n))$, but cannot be recognized by one-way DFA regardless of advice.

In [13] it has been shown that one-way DFA with advice n^{k+1} recognize strictly more languages than the one-way DFA with advice n^k. A similar hierarchy was proved in [4] for non-deterministic automata: one-way NFA with advice $g(n)$ recognize strictly more languages that one-way NFA with advice $f(n)$ for any two functions $f(\cdot)$, $g(\cdot)$ such that $f(n)\log(f(n)) = o(g(n))$ and $g(n) \leq n2^{\frac{n}{2}}$.

The relation between determinism and randomization is a central question in complexity theory, and it seems that uniformity plays an important role here. For example, it is not known whether randomization increases the power of polynomial-time Turing machines, i.e., whether BPP $\supsetneq$ P, but for the non-uniform case the answer is negative, since BPP/*poly* = P/*poly* [1]. For space bounded classes it is known that randomized Turing machines with space complexity $f(n)$ can be simulated by deterministic ones with space complexity $f(n)^2$ [10]. On the other hand, [8] non-deterministic Turing machines can be simulated by randomized ones in the same space complexity.

When considering simpler models of finite automata, Rabin [15] proved that a one-way finite automaton with bounded error can recognize only regular languages. On the other hand, Freivalds [6] showed that the non-regular language $\{a^n b^n \mid n \in \mathbb{N}\}$ can be accepted by a 2-way finite automaton with bounded error.

Very little has been known so far for non-uniform randomized automata. Notably, Küçük [13] mentions that the language $L_{\mathsf{eq3}} = \{w \in \{a,b,c\}^\star \mid |w|_a = |w|_b = |w|_c\}$ can be recognized with linear advice by a 1-sided-error randomized automaton, but cannot be recognized with linear advice by a deterministic automaton.

2 Our contribution

We focus on one-way automata only. It has been known that the expressive power of NFA increases with increased advice size up to $n2^{\frac{n}{2}}$, which is essentially a tight bound, since an advice of size $O(n2^n)$ is sufficient to recognize all languages. For DFA, a similar hierarchy has been only known for polynomials. We prove that this is in fact the best possible result, since a DFA cannot utilize more than polynomial-sized advice (Theorem 4): every language can either be recognized by a DFA with polynomial advice, or cannot be recognized by a DFA with any advice. Using this fact, we prove the same statement for PFA (Theorem 5). We further investigate the relationship between determinism, randomization, and non-determinism. We show that L_{eq3} cannot be recognized by a DFA with any advice (Theorem 7), from which we conclude that both randomization, and non-determinism require only linear advice in order to recognize languages that cannot be recognized deterministically at all.

Since non-determinism with unlimited advice can recognize all languages, we have a strict separation between determinism, randomization, and non-determinism.

We then focus on advice of polynomial length. From Theorems 4 and 5 we know that the power of determinism and randomization does not change. To consider the non-determinism, it has been known [4] that the language of repeated words, L_{ww}, cannot be recognized by a NFA with polynomial advice. We show (Lemma 11) that L_{ww} can be recognized by a PFA with cubic advice. For the other direction, we show (Corollary 15) that there is a language that cannot be recognized by a PFA, but can be recognized by a NFA with linear advice.

Some technical proofs have been omitted due to space constraints.

3 Model and preliminaries

We briefly summarize some standard notions from automata theory we shall use; for more detailed definitions see the respective references. We use the model of multi-tape automata (see e. g. [5]): a non-deterministic two-tape one-way automaton (NFA) A is a tuple $A = (Q, \Sigma_1, \Sigma_2, \delta, q_0, F)$, where Q is a finite set of states, Σ_1, Σ_2 are finite alphabets of the two tapes, q_0 is the initial state, $F \subseteq S$ is the set of accepting states, and the transition function is $\delta : Q \times (\Sigma_1 \cup \{\triangleleft\}) \times (\Sigma_2 \cup \{\triangleleft\}) \mapsto 2^{OUT}$ where $OUT = Q \times \{\rightarrow, \perp\} \times \{\rightarrow, \perp\}$. The meaning is the usual one: the transition is based on the current state of the automaton, and the symbols scanned by both heads (or end-delimiter $\triangleleft \notin \Sigma_1 \cup \Sigma_2$, if the head is already past the end of the input word). The transition results in an action that changes the state, and possibly moves each of the heads independently to the next symbol. The automaton A

accepts a word $(w_1, w_2) \in \Sigma_1 \times \Sigma_2$ if there is an accepting computation of A starting from $(q_0, w_1\triangleleft, w_2\triangleleft)$. If $|\delta(q,a,b)| = 1$ for all $q \in Q$, $a \in \Sigma_1 \cup \{\triangleleft\}$, $b \in \Sigma_2 \cup \{\triangleleft\}$ the automaton is called deterministic (DFA).

We focus our attention on randomized computation, in particular on two-sided bounded-error computations. A probabilistic automaton (PFA) is an extension of a NFA in the sense that the action $OUT = \mathbb{R} \times Q \times \{\rightarrow, \perp\} \times \{\rightarrow, \perp\}$ contains also a real number, and for all fixed $a \in \Sigma_1 \cup \{\triangleleft\}$, $b \in \Sigma_2 \cup \{\triangleleft\}$, $q \in Q$, the respective numbers in $\delta(q,a,b)$ form a probability distribution. For $(p, q', d_1, d_2) \in \delta(q,a,b)$ we say that p is the probability of action (q', d_1, d_2).

Since the introduction of probabilistic finite automata by Rabin [15], the standard way of defining their acceptance is in terms of isolated cut-point λ: a PFA A accepts a language L_A if there exists an $\varepsilon > 0$ such that for each $w \in L_A$ the probability of accepting w is at least $\lambda + \varepsilon$, and for each $w \notin L_A$ the probability of accepting w is at most $\lambda - \varepsilon$. In this paper we assume $\lambda = 1/2$.

For any automaton A, the recognized language is denoted by $\mathscr{L}(A)$. The symbol $\mathcal{X}$ denotes any class of considered automata: DFA, NFA or PFA. The non-uniformity is modelled according to [4, 13], the class of languages recognized by automata of type $\mathcal{X}$ with advice of size $f(n)$ is denoted by $\mathscr{L}(\mathcal{X})/f(n)$:

▶ **Definition 1.** *Let $\mathcal{X}$ be a class of automata. Let $\alpha : \mathbb{N} \mapsto \Sigma_2^\star$ be a function such that $\forall n, |\alpha(n)| = f(n)$. Let $\Sigma_1^{\star\alpha} = \{(w, \alpha(n)) \mid w \in \Sigma_1^\star, n = |w|\} \subseteq (\Sigma_1^\star \times \Sigma_2^\star)$. For a language $L \subseteq \Sigma_1^\star$, let $L_\alpha = \{(w, \alpha(n)) \mid w \in L, n = |w|\} \subseteq (\Sigma_1^\star \times \Sigma_2^\star)$. Then a language L is recognized by an $\mathcal{X}$ automaton A with advice α, if each word $(w_1, w_2) \in L_\alpha$ is accepted, and each word $(w_1, w_2) \in \Sigma_1^{\star\alpha} - L_\alpha$ is rejected. The class of recognized languages is*

$$\mathscr{L}(\mathcal{X})/f(n) := \{L \subseteq \Sigma_1^\star \mid \exists \alpha, \text{ and an } \mathcal{X} \text{ automaton } A \text{ which recognizes } L \text{ with advice } \alpha\}$$

We write $\mathscr{L}(\mathcal{X})/\star$ if the size of the advice is unlimited, $\mathscr{L}(\mathcal{X})/exp$ if it is at most exponential, and $\mathscr{L}(\mathcal{X})/poly$ if it is at most polynomial in the input length.

Often it will be useful to consider the advice tape to be *real-time*, i. e., the automaton advances the advice head in every step. We show that we can assume this without loss of generality (note that real-time *input* head severely reduces the expressive power, since only advice of linear length can be effectively used).

▶ **Lemma 2.** *Let A be an $\mathcal{X}$ automaton. Then there exists an $\mathcal{X}$ automaton B such that B advances the head on the advice tape in every step, and $\mathscr{L}(B) = \mathscr{L}(A)$. Moreover, if A works with advice α, B works with advice β where $|\beta(n)| = O(n\,|\alpha(n)|)$.*

Proof. Let $A = (Q, \Sigma_1, \Sigma_2, \delta, q_0, F)$. We distinguish three cases. First, let A be a DFA with $k = |Q|$ states. Clearly, A can perform at most k steps without moving any of its heads. Fix n, and let the A's advice be $\alpha(n) = a_1 a_2 \cdots$. Let b be a new symbol. Let the advice be $\beta(n) = a_1 b^{kn} a_2 b^{kn} \cdots$. Construct a DFA B that uses advice β, and simulates A in rounds. At the beginning of each round, B's advice head is positioned on a symbol a_i. B scans the symbol on the advice tape, and remembers it in its state. Then it simulates A until A moves the advice head, with the difference that B moves the advice head in each step (the head is positioned on a b symbol). A may perform at most kn steps before moving its advice head, so B can simulate A with its advice head in the block of b's. Once A moves the advice head, B remembers A's state, and moves the advice head to the next a_{i+1}, and a new round begins.

If A is an NFA then the situation is similar: there may be some computations of A that perform more than k steps without moving any head, however, for each accepted word there is an accepting computation that makes at most k steps without moving any head. Hence, the same simulation as in the deterministic case works with one exception: if B encounters the end of the block of b's before the simulated computation of A moves the advice head, B rejects.

Finally, let A be a PFA. This case requires more care than the NFA since all accepting computations contribute to the probability of acceptance. We first transform A into a PFA A' that always moves at least one head, i. e., contains no actions of the form $(p, q, \bot, \bot)$, and then use the padding technique from the DFA case. ◀

For some of the proofs we shall use a normal form of automata with real-time advice tape which doesn't scan the input tape when the input head is not moved, as follows:

▶ **Lemma 3.** *Let $A = (Q_A, \Sigma_1, \Sigma_2, \delta_A, q_0, F_A)$ be a* DFA *(or* NFA*) with real-time advice tape. There exists a* DFA *(or* NFA*) $B = (Q_B, \Sigma_1, \Sigma_2, \delta_B, q_0, F_B)$ with real-time advice tape using the same advice as A such that $\mathscr{L}(A) = \mathscr{L}(B)$ and the following holds: The states Q_B can be partitioned into $Q_B = Q_B^{\rightarrow} \cup Q_B^{\varepsilon}$ such that*

1. *if $(q', \bot, \rightarrow) \in \delta_B(q, a, b)$ for some $q \in Q_B$, $a \in \Sigma_1$, $b \in \Sigma_2$ then $q' \in Q_B^{\varepsilon}$, and*
2. *for any $q \in Q_B^{\varepsilon}$, $b \in \Sigma_2$ it holds $\delta_B(q, a, b) = \delta_B(q, a', b)$ for all $a, a' \in \Sigma_1$.*

4 Upper bound on advice for DFA and PFA

Our first contribution is an upper bound on the advice that can be utilized by DFA and PFA. It has been proven in [4] that there are languages that cannot be recognized by DFA with any amount of advice, which is in contrast with NFA that can recognize all languages with advice $O(n2^n)$. Here we show that DFA cannot utilize more than polynomial advice.

▶ **Theorem 4.** $\mathscr{L}(\text{DFA})/poly = \mathscr{L}(\text{DFA})/\star$.

Proof. Consider a DFA A with the form of Lemma 3 with $k = |Q|$ states, working with advice $a_1, \ldots, a_\ell$. A pair (i, q), $1 \le i \le \ell$, $q \in Q$ is called a *point*. We say a point (i, q) is *active*, if $q \in Q_A^{\rightarrow}$, and *passive* otherwise. The computation on any input word of length n defines a *trace*, which is a sequence of points containing exactly n active points [1]. Altogether, there are 2^n traces. Let P be the set of points that are included in some trace.

We prove that there may be at most $2k(n+1)^k$ active points in P. In particular, we prove by induction on m the following claim:

Consider an interval $j_1, \ldots, j_2$ of positions on the advice tape. Suppose that for some m, $1 \le m \le k$, there are at least m traces that are point-wise disjoint on the interval $j_1, \ldots, j_2$, i. e., they don't share any point of the form (i, q), $j_1 \le i \le j_2$. Moreover, suppose that none of these m traces contain an active point of the form (i, q) for $j_1 < i < j_2$. Then there are at most $2k(n+1)^{k-m}$ active points from P of the form (i, q), $j_1 \le i \le j_2$.

[1] We assume without loss of generality that A reads the whole input.

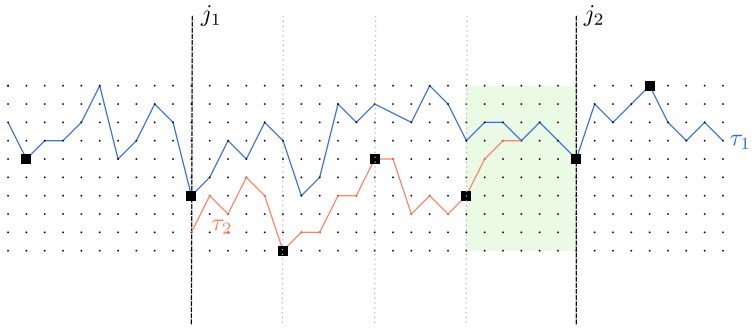

Figure 1 The trace τ_1 has no active point (square) in the interior of the $j_i, \ldots, j_2$ interval. The trace τ_2 splits the interval into $n+1$ sub-intervals; each of them apart from the last one has two point-wise disjoint traces.

First note that there are k points of the form (i, q) for a fixed i, so in any interval there may be at most k point-wise disjoint traces. For the base of the induction, assume that $m = k$. Then each point of the $j_1, \ldots, j_2$ interval belongs to exactly one of the k disjoint traces. Since the interior points are not active, there are at most $2k$ active points in the interval.

For the induction step consider an interval $j_1, \ldots, j_2$ with m point-wise disjoint traces. If there are no active points in the interval, the claim holds. Otherwise, take the trace τ that contains the right-most active point of the form (i, q) for $j_1 < i < j_2$. The trace τ contains at most n active points in the interval, so it splits the interval into at most $n+1$ sub-intervals. By the choice of τ, the last sub-interval does not contain any active points. Note that, since the automaton is deterministic, once two traces share a point, they follow the same sequence of points until an active point is reached, at which they may diverge. Since none of the m traces contain an active point in the interior of the interval, τ is disjoint with all of them on all but the last sub-interval. Hence, the last sub-interval does not contain any active points, and each of the remaining sub-intervals contains at least $m+1$ disjoint traces. Applying the induction hypothesis on all but the last sub-interval, we get the claim.

Now we have proved that there are only polynomially many active points. To finish the proof consider an interval $j_1, \ldots, j_2$ of positions on the advice tape that does not contain an active point. This means that no computation on any word queries the input tape in this window; the automaton only considers the state and the advice symbol. Hence, the advice in this window can be replaced by an advice of constant length specifying the state-to-state transition relation. ◀

We have just proved that a DFA cannot utilize more than polynomial advice. We use this fact, and show that the same is true also for PFA.

▶ **Theorem 5.** $\mathscr{L}(\mathrm{PFA})/\star \subseteq \mathscr{L}(\mathrm{PFA})/poly$

Proof. Consider a PFA A with real-time advice tape, working with advice α. Let Σ_1, Σ_2 be the input, and advice alphabet, respectively. We first construct a DFA B with input alphabet Σ_1, and advice alphabet $\overline{\Sigma}_2$. Also, for each n we construct a probability distribution π_n over strings $\overline{\Sigma}_2^{|\alpha(n)|}$ such that for each $w \in \Sigma_1^n$ the probability that A accepts w with advice $\alpha(n)$ is the same as the probability that B accepts w with an advice that is selected at random

from π_n. The overall idea is that B's advice contains an encoding of the choices made by A. Since A has real-time advice tape, it makes one probabilistic decision per advice symbol in every computation. B shall use a two-track advice tape, with the second track used to simulate the random decisions of A. However, A may read a given advice symbol in different states and different positions on the input tape in each of its computations. For our purposes we need to encode the probabilistic decisions is a uniform way. Consider a particular $q \in Q_A$, $a \in \Sigma_1$, $b \in \Sigma_2$. Let $p_1, \ldots, p_z$ be the probabilities of actions in $\delta_A(q, a, b)$. Let $S_{q,a,b}$ be a subdivision of the unit interval $(0, p_1), (p_1, p_1+p_2), (p_1+p_2, p_1+p_2+p_3), \ldots, (\sum_{i=1}^{z-1} p_i, 1)$. Let $S = (s_0, s_1), (s_1, s_2), \ldots, (s_{c-1}, s_c)$ for some c, with $s_0 = 0$, $s_c = 1$ be a common subdivision of all $S_{q,a,b}$'s. Let $\overline{\Sigma}_2 = \Sigma_2 \times \mathbb{Z}_c$. A string from π_n is obtained by first taking $\alpha(n)$, and then adding to each symbol a value $i \in \mathbb{Z}_c$ such that each i is taken with probability $s_{i+1} - s_i$.

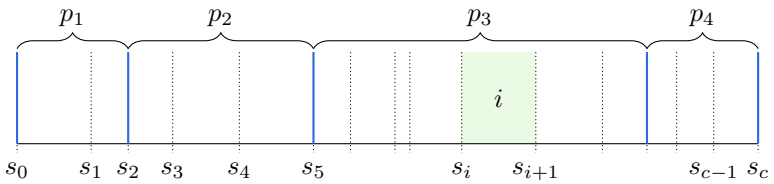

Figure 2 Specifying the transition probabilities of A. If the current advice symbol contains $i \in \mathbb{Z}_c$, which corresponds to the random choice of interval (s_i, s_{i+1}), B selects the third action in $\delta_A(q, a, b)$. If each $i \in \mathbb{Z}_c$ is selected with probability $s_{i+1} - s_i$, the probability of choosing the third action is p_3.

B uses the advice from π_n to simulate A with the advice $\alpha(n)$ as follows (see Figure 2): in each step when A is in state q, and symbols $a \in \Sigma_1$, $b \in \Sigma_2$ are on the input, and advice tapes, respectively, B considers also the symbol $i \in \mathbb{Z}_c$ from the advice tape, and based upon where the interval (s_i, s_{i+1}) is located within the subdivision $S_{q,a,b}$ selects the appropriate action from $\delta_A(q, a, b)$.

Consider a computation ξ of A. The overall probability of ξ is $p = \prod p_i$ where p_i is the probability of choosing the corresponding action from $\delta_A(q, a, b)$. Since the probability that B chooses the same action as A in the i-th step is p_i, the probability that B simulates ξ with random advice is p. Hence, for any word w the probability of acceptance of w is the same for A with advice $\alpha(n)$, and B with random advice from π_n.

The distribution π_n is defined over an exponential set of strings from $\overline{\Sigma}_2$, each of them of possibly super-polynomial length. We can use the construction from the proof of Theorem 4 to shorten each of them to polynomial length without affecting B's behavior. Note that the proof modifies the automaton (when an interval of the advice without active points is removed, it is replaced by a state-to-state transition for this interval; the modified automaton has to read this specification, and change the state accordingly), but the modification to the automaton does not depend on the content of the advice tape. Hence, we can consider a modified automaton B' working with a shortened advice, selected randomly from a distribution that ranges over the shortened strings, and maintains the probabilities from π_n (we shall abuse the notation and refer to this distribution by π_n).

Now we have an exponential number of polynomially-sized advice strings, such that for each input word, an advice string randomly chosen from π_n is *good* (i. e., if the word is in the language, B' accepts, and if not then B' rejects) with probability $1/2 + \varepsilon$ for some constant ε.

Consider a set of (not necessarily distinct) κ advice strings sampled from π_n. Let w be a fixed input word, and let X_i be the selector random variable indicating whether the i-th advice string from the set is good for w. Since $\Pr[X_i = 1] \geq 1/2 + \varepsilon$, we can use Chernoff bound to conclude that with probability $1 - c^{-\kappa}$ for some $c > 1$, at least $\kappa(1/2 + \varepsilon')$ of the advice strings are good for w for an arbitrary $\varepsilon' < \varepsilon$.

Using union bound we can argue that the probability that at least $\kappa(1/2 + \varepsilon')$ strings from the set are good for any of the 2^n input words is at least $1 - 2^n c^{-\kappa}$. For $\kappa > \frac{n}{\log_2 c}$ this probability is non-zero, so we can conclude that for any $\kappa > \frac{n}{\log_2 c}$ there exists a set of at most κ advice strings such that for any input word w, at least $\kappa(1/2 + \varepsilon')$ of them are good.

Finally, consider a PFA C that works as follows. The advice string is formed by the κ strings from the set described above, divided by a sequence of $\log_2 \kappa$ delimiters. C selects one of the strings uniformly at random, and simulates B' on that string. C can select one of the strings uniformly at random by making a random choice on each of the delimiters: the probability of the event that all choices are positive is $1/\kappa$. Since for any input word w, at least a fraction $1/2 + \varepsilon'$ of the strings are good, simulating B' on a randomly selected advice string yields a probability $1/2 + \varepsilon'$ of the computation to be good. ◀

▶ **Corollary 6.** PFA *with unlimited advice have a normal form where the advice consists of polynomially many blocks of polynomial size. The automaton simulates a fixed* DFA *on a block of advice chosen uniformly at random.*

5 Separation of $\mathscr{L}$(DFA), $\mathscr{L}$(PFA), and $\mathscr{L}$(NFA)

In this section we develop results that separate various classes. We start with determinism. Our first aim is to show that $\mathscr{L}(\text{DFA})/\star$ is strictly contained in both $\mathscr{L}(\text{PFA})/poly$ and $\mathscr{L}(\text{NFA})/poly$. From Theorem 4 we know that $\mathscr{L}(\text{DFA})/\star \subseteq \mathscr{L}(\text{PFA})/poly$, and similarly $\mathscr{L}(\text{DFA})/\star \subseteq \mathscr{L}(\text{NFA})/poly$. There have been known examples of languages than cannot be recognized by DFA (with any advice), e. g. $L^{\mathsf{c}}_{ww} \in \mathscr{L}(\text{NFA})/O(n^2) - \mathscr{L}(\text{DFA})/\star$, see [4][2], however, it was not clear whether they are accepted by a PFA.

We not only ask the question whether $\mathscr{L}(\text{DFA})/\star \subsetneq \mathscr{L}(\text{PFA})/\star$, but more precisely, we ask what is the smallest advice a PFA or NFA needs to be able to recognize a language that is not in $\mathscr{L}(\text{DFA})/\star$. We can prove the following [3]:

▶ **Theorem 7.** $L_{eq3} \in \mathscr{L}(\text{PFA})/O(n) - \mathscr{L}(\text{DFA})/\star$.

Let an (i,j)-class of n-letter words be $[i,j]_n = \{w \in \{a,b,c\}^n \mid |w|_a = n-j,\ |w|_b = n-i,\}$, i. e., the set of n-letter words w such that $|w|_a - |w|_c = i$ and $|w|_b - |w|_c = j$. Suppose that some DFA recognizes L_{eq3}. Then, after reading the first k symbols, it must be in different configurations (i. e., different pair state, and position on the advice tape) for prefixes from different classes $[i,j]_k$. In the proof we shall consider intervals of positions on the advice tape, and we shall argue that the automaton must represent a growing number of classes in configurations with positions within this interval. To achieve this we use the following lemma

[2] recall that $L_{ww} = \{ww \mid w \in \{a,b\}^\star\}$
[3] recall that $L_{\mathsf{eq3}} = \{w \in \{a,b,c\}^\star \mid |w|_a = |w|_b = |w|_c\}$

concerning the growth of sets of classes: consider a set S of (i,j)-classes of n-letter words. Let S' (called the *boundary* of S) be the set of classes formed by words wx for each word $w \in [i,j]_n \in S$, and $x \in \Sigma$. Hence, for a set of classes S the boundary is (see Figure 3a):

$$S' = \left\{ [i,j]_{n+1} \mid [i-1,j]_n \in S \vee [i,j-1]_n \in S \vee [i+1,j+1]_n \in S \right\}$$

We argue that S' must be large enough:

▶ **Lemma 8.** *For a fixed n, let S be a set of classes of n-letter words, and let S' be the boundary of S. Then $|S'| \geq |S| + \sqrt{|S|}/3$.*

Proof. Let $I = \{i \mid \exists j : [i,j]_n \in S\}$, and $J = \{j \mid \exists i : [i,j]_n \in S\}$. Clearly at least one of $|I|$, $|J|$ is at least $\sqrt{|S|}$. Without loss of generality, let $|I| \geq \sqrt{|S|}$. Each class in $[i,j]_n \in S$ contributes to three classes in S', and each class $[i,j]_{n+1}$ can receive contributions from at most three classes from S. Let $C \subseteq S'$ be the classes from S' that receive contribution from at most two classes from S. Hence, for the overall contribution of S it holds $3|S| \leq 2|C| + 3|S' - C|$ from which $|S'| \geq |S| + |C|/3$. We show that $|C| \geq |I| \geq \sqrt{|S|}$ thus completing the proof.

We say that class the $[i,j]_n$ is located in column i and row j. Consider each $i \in I$ in decreasing order, and for each of them we find a unique class $c_i \in C$. The c_i will be located either in column i or $i-1$. Let j_i be the smallest number such that $[i,j_i]_n \in S$, or ∞ if no such class is in S. When considering a column i, we distinguish two cases: If $j_{i-1} \geq j_i - 1$ (see Figure 3b), we assign $c_i = [i-1, j_i - 1]_{n+1}$. Now suppose that $j_{i-1} < j_i - 1$ (see Figure 3c). The class $[i, j_{i-1}]_{n+1} \in S'$ because $[i-1, j_{i-1}]_n \in S$, and has at most two neighbors because $[i, j_{i-1} - 1]_n \notin S$, so we would like to assign $c_i = [i, j_{i_1}]_{n+1}$. However, it may happen that this class has already been assigned as c_{i+1}. In the latter case let k_i be the maximum j such that $[i,k_i]_n \in S$, and distinguish two sub-cases. If $k_{i-1} \leq k + i$ (see Figure 3d) we can assign $c_i = [i, k_i + 1]_{n+1}$, since there is only one c_{i+1}. Finally, if $k_{i-1} > k + i$ (see Figure 3e) we can assign $c_i = [i-1, k_{i-1} + 1]_{n+1}$.

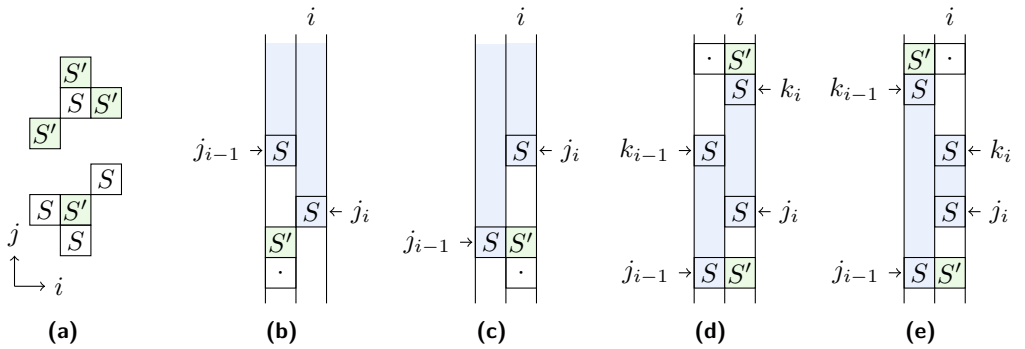

Figure 3 (a) A class $[i,j]_n \in S$ contributes to the border S' with classes $[i+1,j]_{n+1}$, $[i,j+1]_{n+1}$, $[i-1,j-1]_{n+1}$. Hence, a class in S' receives contributions from the three classes in S. (b) The case $j_{i-1} \geq j_i - 1$. The class $[i-1, j_i - 1]_{n+1}$ is in S' because of $[i,j_i]_n \in S$, and has at most two neighbors, since $[i-1, j_i - 2]_n \notin S$. (c) The basic case $j_{i-1} < j_i - 1$. (d) The sub-case $j_{i-1} < j_i - 1$, and $k_{i-1} \geq k_i$. (e) The sub-case $j_{i-1} < j_i - 1$, and $k_{i-1} > k_i$.

◀

Now we are able to prove the theorem.

Proof of Theorem 7. The fact that $L_{\mathsf{eq3}} \in \mathscr{L}(\mathrm{PFA})/O(n)$ comes from [13]. We show that $L_{\mathsf{eq3}} \notin \mathscr{L}(\mathrm{DFA})/\star$. For the sake of contradiction, let us suppose that a DFA A in the normal form from Lemma 3 recognizes L_{eq3}, and fix some n. Note that each word $u \in \Sigma^*$, $|u| < n/3$ is the prefix of some word from L_{eq3}. Consider all computations of A on all words from L_{eq3} of length n. For a given ℓ, consider the moment when A has just read a prefix u, $|u| = \ell$, i. e., the step when A for the first time moves the input head beyond u; we say that this is the situation after ℓ rounds.

A configuration of A is the pair containing the state and the position on the advice tape. For two words u, u', $|u| = |u'| = \ell < n/3$, if A is in the same configuration after reading u and u' in the first ℓ rounds, then u,u' belong to the same (i,j)-class; otherwise there would be some word of the form uz that is not recognized correctly by A. Hence, for each $\ell \leq n/3$, each configuration *belongs* to at most one class $[i,j]_\ell$, and each position on the advice tape belongs to at most k such classes, where k is the number of states of A. There are $\sum_{i=0}^{\ell}(\ell - i) = \binom{\ell+1}{2}$ classes of ℓ-letter words, and for $\ell < n/3$ all of them contain prefixes of some n-letter words from L_{eq3}.

Now consider the situation after ℓ rounds. Fix some interval $I = x_1, \ldots, x_2$ of positions on the advice tape. Suppose that the configurations with the advice head in I after ℓ rounds belong to more than $9k^2$ distinct classes $[i,j]_\ell$, i. e., for each such class c there is a word $u \in c$ such that A has the advice head in I when the input head is being moved beyond u. We say that I is *full* after ℓ rounds. Note that whenever an interval I is full after $\ell < n/3 - 1$ rounds, the number of classes it contains grows in the next round. To see this, note that for each class $c = [i,j]_\ell$ that is contained in I, there is a word $u \in c$ such that A has its advice head in I after reading u. Since ua, ub, uc are all prefixes of some words from L_{eq3} belonging to distinct classes, the respective classes must be represented by A after $\ell + 1$ rounds. Due to Lemma 8 if I contained $p > 9k^2$ classes after ℓ rounds, more than $p + k$ classes must be represented by A after $\ell + 1$ rounds. However, words from at most k classes may have computations in which the advice head ends outside of I: since A is in the normal form from Lemma 3, if two computations pass through the same configuration, they must continue together. Hence, the number of classes that must be contained in I after $\ell + 1$ rounds is more than p.

Since the number of classes belonging to a full interval grows, if an interval I is full after ℓ rounds, after $\ell + 9k^2 + k$ rounds $I = x_1, \ldots, x_2$ contains more than $18k^2 + k$ distinct classes. Let i be the first index in I such that the interval $x_1, \ldots, i$ is full. Since the interval $x_1, \ldots, i-1$ contained at most $9k^2$ classes, and at most k classes can share the advice-head position i, the interval $i+1, \ldots, x_2$ is full, too.

Finally, a simple induction yields that after $t(9k^2 + k) < n/3$ steps, the advice tape contains at least 2^{t-1} disjoint full intervals. Since k is constant, the length of the advice tape of A must be super-polynomial, which is a contradiction due to Theorem 4. ◀

From Theorem 4 and Theorem 7 we get

▶ **Corollary 9.** $\mathscr{L}(\mathrm{DFA})/\star \subsetneq \mathscr{L}(\mathrm{PFA})/poly$.

We show analogous result for NFA.

▶ **Theorem 10.** $L^{\mathsf{c}}_{\mathsf{eq3}} \in \mathscr{L}(\mathrm{NFA})/O(n) - \mathscr{L}(\mathrm{DFA})/\star$.

Proof. Note that since for each $\mathcal{F}$, $\mathscr{L}(\mathrm{DFA})/\mathcal{F}$ is closed under complement, the fact that $L^{\mathsf{c}}_{\mathsf{eq3}} \notin \mathscr{L}(\mathrm{DFA})/\star$ comes directly from Theorem 7.

To show that $L^{\mathsf{c}}_{\mathsf{eq3}} \in \mathscr{L}(\mathrm{NFA})/O(n)$, note that $L^{\mathsf{c}}_{\mathsf{eq3}} = \{w \in \Sigma^\star \mid \exists x \in \Sigma : |w|_x > |w|/3\}$. The following NFA A with real-time advice tape can recognize $L^{\mathsf{c}}_{\mathsf{eq3}}$ with linear advice: A checks in state that $|w| = 3m$, and rejects otherwise. In parallel, A guesses $x \in \Sigma$, and moves the advice head twice if the scanned symbol is x, and once otherwise. The length of the advice is $4n/3$. If the length of the advice tape is reached before the end of the input, it holds that $|w|_x > n/3$. ◀

In the rest of this section we deal with the relationship of randomization and non-determinism. For the following results, we use standard notions from communication complexity; for a general primer see e. g. [14]. In particular, we use the notation from [12]: for a boolean function $f : X \times Y \mapsto \{0,1\}$, we denote $R^{A\to B}_{\varepsilon}(f)$ the randomized private-coin one-round communication complexity of f with two-sided error probability ε. When speaking about protocols, we always mean this type of protocols. It is easy to see using Chernoff bound that running some constant number of copies of the protocol and then deciding based upon majority reduces the error probability below an arbitrary constant. Since we are interested only in $O(R^{A\to B}_{\varepsilon}(f))$, we can leave the ε out and write only $R^{A\to B}(f)$. We start by showing how a PFA can accept L_{ww}.

▶ **Lemma 11.** $L_{ww} \in \mathscr{L}(\mathrm{PFA})/O(n^3)$.

Proof idea. The proof follows from standard communication complexity ideas (compare e. g. [14], example 3.13). A randomized protocol for equality function with public random bits can be transformed into a PFA. ◀

Since $L_{ww} \notin \mathscr{L}(\mathrm{NFA})/poly$, we have

▶ **Corollary 12.** $\mathscr{L}(\mathrm{PFA})/O(n^3) \not\subseteq \mathscr{L}(\mathrm{NFA})/poly$

On the other hand, we present a language that can be recognized non-deterministically, but not probabilistically. The following simple observation states that it is possible to use a PFA to construct a communication protocol:

▶ **Lemma 13.** *Let $\ell : \mathbb{N} \mapsto \mathbb{N}$, and $\{f_n\}_{n\in\mathbb{N}}$ be a family of functions $f_n : \Sigma^n \times \Sigma^{\ell(n)} \mapsto \{0,1\}$. Let $L = \{u\#v \mid \exists n : f_n(u,v) = 1\}$. If $L \in \mathscr{L}(\mathrm{PFA})/s(n)$, then $R^{A\to B}(f) = O(\log s(n))$.*

Proof. Let A be the PFA that recognizes L with advice $\alpha(n)$. The protocol works as follows: The first party receives u, computes $n = |u| + \ell(|u|) + 1$, and starts to simulate A on the (prefix of the) input tape $u\#$ with advice $\alpha(n)$. When the # is reached in the input tape, the state of A, and the position on the advice tape are encoded in $O(\log|\alpha(n)|)$ bits, and transmitted to the second party that finishes the simulation. ◀

▶ **Theorem 14** ([12]). *For $x, y \subseteq \{1,\ldots,n\}$, $DISJ(x,y)$ is defined to be 1 if and only if $x \cap y = \emptyset$. The communication complexity is $R^{A\to B}(DISJ) = \Omega(n)$.*

▶ **Corollary 15.** $\mathscr{L}(\mathrm{NFA})/O(n) \not\subseteq \mathscr{L}(\mathrm{PFA})/\star$

Proof. For $x, y \subseteq \{1, \ldots, n\}$, $\overline{DISJ}(x, y)$ is defined to be 1 if and only if $x \cap y \neq \emptyset$. Due to symmetry, we have from Theorem 14 that $R^{A\to B}(\overline{DISJ}) = \Omega(n)$. Consider a language $L = \{u\#v \mid u, v \in \mathbb{Z}_2^\star,\ |u| = |v| \wedge\ \exists i :\ u_i \neq v_i\}$. If $L \in \mathscr{L}(\mathrm{PFA})/poly$, then Lemma 13 would yield $R^{A\to B}(\overline{DISJ}) = O(\log n)$ – a contradiction. Hence, $L \notin \mathscr{L}(\mathrm{PFA})/poly$.

On the other hand, it is easy to see that $L \in \mathscr{L}(\mathrm{NFA})/O(n)$: the advice $\alpha(n) = a^n b^n a^n$. The automaton (with real-time advice tape) first moves the advice and input heads in synchrony, and non-deterministically decides to check a given position. It remembers the current input symbol, moves the advice head to the beginning of the block of b's, and moves both heads in synchrony for the next n times (until the advice head encounters an a). Finally, it checks whether the input symbols on the corresponding positions are the same. ◀

6 Conclusion

We presented the first systematic attempt at mapping the power of randomization in non-uniform finite automata. First, we showed the strict separation $\mathscr{L}(\mathrm{DFA})/\star \subsetneq \mathscr{L}(\mathrm{PFA})/\star \subsetneq \mathscr{L}(\mathrm{NFA})/\star$. We also showed that both PFA and NFA need only linear advice to be able to recognize languages not in $\mathscr{L}(\mathrm{DFA})/\star$. Finally, we showed that if the advice is restricted to polynomial size, the power of determinism and randomization does not change. However, $\mathscr{L}(\mathrm{DFA})/\star \subsetneq \mathscr{L}(\mathrm{NFA})/poly \subsetneq \mathscr{L}(\mathrm{NFA})/\star$, and $\mathscr{L}(\mathrm{NFA})/poly$ is incomparable to $\mathscr{L}(\mathrm{PFA})/poly$.

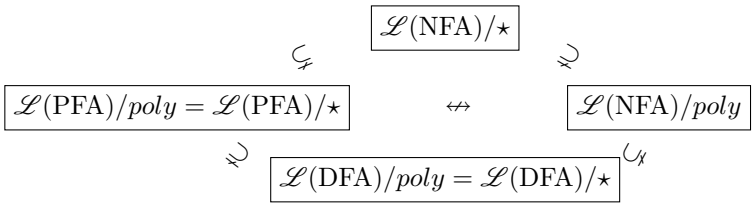

Figure 4 Relationship of the classes of non-uniform one-way finite automata.

Since there is a significant gap between $\mathscr{L}(\mathrm{NFA})/\star$ and $\mathscr{L}(\mathrm{NFA})/poly$, it might be beneficial to investigate the impact of advice size to the power of non-determinism in a more fine-grained way. In particular, we have shown that linear advice is sufficient for NFA to recognize languages outside of $\mathscr{L}(\mathrm{PFA})/\star$, and cubic advice is sufficient for PFA to recognize languages outside of $\mathscr{L}(\mathrm{NFA})/poly$. It would be interesting to know where are the precise boundaries when $\mathscr{L}(\mathrm{PFA})/f(n) \not\subseteq \mathscr{L}(\mathrm{NFA})/g(n)$ and vice-versa.

Also, our results hold for one-way automata. We expect that the situation for two-way automata would be significantly different. Since the relation of one-way versus two-way is in general important, we believe it would be beneficial to investigate it also in the non-uniform case.

References

1 Leonard M. Adleman. Two theorems on random polynomial time. In *19th Annual Symposium on Foundations of Computer Science, Ann Arbor, Michigan, USA, 16-18 October 1978*, pages 75–83. IEEE Computer Society, 1978. doi:10.1109/SFCS.1978.37.

2 Allan Borodin. On relating time and space to size and depth. *SIAM J. Comput.*, 6(4):733–744, 1977. doi:10.1137/0206054.

3 Carsten Damm and Markus Holzer. Automata that take advice. In Jirí Wiedermann and Petr Hájek, editors, *Mathematical Foundations of Computer Science 1995, 20th International Symposium, MFCS'95, Prague, Czech Republic, August 28 - September 1, 1995, Proceedings*, volume 969 of *Lecture Notes in Computer Science*, pages 149–158. Springer, 1995. doi:10.1007/3-540-60246-1_121.

4 Pavol Duris, Rafael Korbas, Rastislav Královic, and Richard Královic. Determinism and nondeterminism in finite automata with advice. In Hans-Joachim Böckenhauer, Dennis Komm, and Walter Unger, editors, *Adventures Between Lower Bounds and Higher Altitudes - Essays Dedicated to Juraj Hromkovič on the Occasion of His 60th Birthday*, volume 11011 of *Lecture Notes in Computer Science*, pages 3–16. Springer, 2018. doi:10.1007/978-3-319-98355-4_1.

5 Patrick C. Fischer and Arnold L. Rosenberg. Multitape one-way nonwriting automata. *J. Comput. Syst. Sci.*, 2(1):88–101, 1968. doi:10.1016/S0022-0000(68)80006-6.

6 Rusins Freivalds. Probabilistic two-way machines. In Jozef Gruska and Michal Chytil, editors, *Mathematical Foundations of Computer Science 1981, Strbske Pleso, Czechoslovakia, August 31 - September 4, 1981, Proceedings*, volume 118 of *Lecture Notes in Computer Science*, pages 33–45. Springer, 1981. doi:10.1007/3-540-10856-4_72.

7 Rusins Freivalds. Amount of nonconstructivity in deterministic finite automata. *Theor. Comput. Sci.*, 411(38-39):3436–3443, 2010. doi:10.1016/j.tcs.2010.05.038.

8 John Gill. Computational complexity of probabilistic turing machines. *SIAM J. Comput.*, 6(4):675–695, 1977. doi:10.1137/0206049.

9 Oscar H. Ibarra and Bala Ravikumar. Sublogarithmic-space turing machines, nonuniform space complexity, and closure properties. *Mathematical Systems Theory*, 21(1):1–17, 1988. doi:10.1007/BF02088003.

10 H. Jung. Relationships between probabilistic and deterministic tape complexity. In Jozef Gruska and Michal Chytil, editors, *Mathematical Foundations of Computer Science 1981, Strbske Pleso, Czechoslovakia, August 31 - September 4, 1981, Proceedings*, volume 118 of *Lecture Notes in Computer Science*, pages 339–346. Springer, 1981. doi:10.1007/3-540-10856-4_101.

11 Richard M. Karp and Richard J. Lipton. Turing machines that take advice. *Enseignement Mathématique*, 28(2):191–209, 1982.

12 Ilan Kremer, Noam Nisan, and Dana Ron. On randomized one-round communication complexity. *Comput. Complex.*, 8(1):21–49, 1999. doi:10.1007/s000370050018.

13 Ugur Küçük, A. C. Cem Say, and Abuzer Yakaryilmaz. Finite automata with advice tapes. *Int. J. Found. Comput. Sci.*, 25(8):987–1000, 2014. doi:10.1142/S012905411440019X.

14 Eyal Kushilevitz and Noam Nisan. *Communication complexity*. Cambridge University Press, 1997.

15 Michael O. Rabin. Probabilistic automata. *Information and Control*, 6(3):230–245, 1963. doi:10.1016/S0019-9958(63)90290-0.

16 Kohtaro Tadaki, Tomoyuki Yamakami, and Jack C. H. Lin. Theory of one-tape linear-time turing machines. *Theor. Comput. Sci.*, 411(1):22–43, 2010. doi:10.1016/j.tcs.2009.08.031.