Probabilism versus Alternation for Automata

Georg Schnitger(✉)

Institut für Informatik, Goethe-Universität Frankfurt, Frankfurt, Germany
georg.schnitger@googlemail.com

Abstract. We compare the number of states required for one-way probabilistic finite automata with a positive gap ($1P_2FAs$) and the number of states required for one-way alternating automata (1AFAs). We show that a $1P_2FA$ P can be simulated by 1AFA with an at most polynomial increase in the number $s(P)$ of states of P, provided only inputs of length at most poly$(s(P))$ are considered. On the other hand we gather evidence that the number of states grows super-polynomially if the number of alternations is bounded by a fixed constant. Thus the behavior of one-way automata seems to be in marked contrast with the behavior of polynomial-time computations.

Many thanks to Juraj for many ideas, many questions, many answers and lots of hiking.

1 Introduction

Sipser [16] showed in 1983 that the complexity class BPP of languages recognizable in polynomial time by probabilistic Turing machines with bounded error probability is contained in the polynomial hierarchy. Later Peter Gacz and independently Lautemann [15] strengthened his results to imply that $\mathsf{BPP} \subseteq \Sigma_2^p \cap \Pi_2^p$ holds.

What is the relation between probabilism and alternation for one-way automata if we consider the number of states as a resource? We follow the notation in [14] and compare the number of states required by

(a) $1P_2FAs$, i.e., one-way ***p***robabilistic ***f***inite ***a***utomata with two-sided bounded error,
(b) $1\exists_k FAs$ and $1\forall_k FAs$, i.e., one-way alternating ***f***inite ***a***utomata which start in an existential resp. universal state and alternate at most $k-1$ times between existental and universal states,
(c) 1AFAs, i.e., one-way ***a***lternating ***f***inite ***a***utomata.

(See Sect. 2 for formal definitions.) For an automata M let $s(M)$ be the number of states of M. To investigate the state complexity we again borrow from [14] and

© Springer Nature Switzerland AG 2018
H.-J. Böckenhauer et al. (Eds.): Hromkovič Festschrift, LNCS 11011, pp. 113–126, 2018.
https://doi.org/10.1007/978-3-319-98355-4_8

define complexity classes not consisting of languages but of families of languages. In particular, for a computation mode χ define the class

$$\{(L_n)_{n\geq 1} \mid \text{there are one-way } \chi\text{-automata } M_n \text{ with } L(M_n) = L_n \text{ such that } s(M_n) \text{ is polynomial in } n\}.$$

As a consequence, by choosing the computation mode accordingly, we may introduce $1\mathrm{P}_2$ as the automata-version of BPP and $1\Sigma_k, 1\Pi_k, 1H, 1A$ as the automata versions of Σ_k^p, Π_k^p, PH and AP respectively.

We gather evidence that a *fixed* number of alternations is insufficient to simulate $1\mathrm{P}_2$FAs efficiently, i.e., that $1\mathrm{P}_2$ is not a subset of $1\Sigma_k$ for any $k \in \mathbb{N}$. To do so we investigate "circuit automata", a class of 1AFAs whose state transitions depend – for all but one input position – on the position and not on the symbol found in this position. Circuit automata with few alternations are quite powerful since, if states and circuit size are compared, they turn out to be "equivalent" to alternating circuits of AND, OR and NOT-gates with small depth (see Proposition 1). But for the same reason circuit automata are quite weak since they are incapable of recognizing parity efficiently. We construct languages $L_{n,k}$ implementing the idea of "nested equalities" (see Definition 2) such that the family $(L_{n,k})_{n\geq 1}$ belongs to $1\mathrm{P}_2$ but show in Theorem 1 that circuit automata with $k-1$ alternations require a number of states which is super-polynomial in n. At least intuitively, alternations and not automata-specific abilities seem to matter when recognizing $L_{n,k}$ and hence we conjecture that $(L_{n,k})_{n\geq 1}$ does not belong to $1\Sigma_{o(k)}$.

Geffert [6] has separated all levels of the "polynomial" hierarchy $1H$ and was even able to do so for two-way automata. However, we could not apply his methods to languages which are easy probabilistically, but hard when using not too many alternations. Instead we apply methods of circuit complexity and in particular the Switching Lemma for bounded depth alternating circuits [7].

On the other hand we show in Theorem 2 that 1AFAs with an unbounded number of alternations simulate $1\mathrm{P}_2$FAs efficiently when input of "too large" length are excluded. This result is made possible by a normal form for probabilistic automata (see Proposition 2), namely as a weighted majority of DFAs.

Hromkovič has made, among others, many fundamental contributions to the understanding of the state complexity of nondeterministic and probabilistic automata. In [9] the surprising result is shown that DFAs require at most quadratically more states than Las Vegas automata and that this gap is largest possible. This result was made possible by investigations into communication complexity [4,5,10].

He showed how to apply communication complexity to the state complexity of various automata models, predominantly NFAs [8,12] and followed this approach systematically. For instance it was possible to show a rather fine-grained state hierarchy for NFAs with bounded ambiguity, where the ambiguity of an NFA is the maximal number of accepting computations for a given input size. Another example of the power of communication arguments is the proof in [11] that the

standard construction of converting NFAs with ε-transitions into NFAs without ε-transitions is almost optimal.

We formally introduce $1\mathrm{P}_2\mathrm{FAs}$, $1\Sigma_k\mathrm{FAs}$, $1\Pi_k\mathrm{FAs}$, 1AFAs and circuit automata in Sect. 2. Theorems 1 and 2 are shown in Sects. 3 and 4 respectively. Conclusions are given in Sects. 5.

2 Basics

We introduce probabilistic and alternating automata. Finally circuit automata are defined and their relation to unbounded fan-in circuits is made explicit.

Probabilistic Finite Automata. A one-way probabilistic finite automaton (1PFA) $P = (Q, \Sigma, \delta, \lambda, q_0, F)$ with *cut-point* λ is defined by a sequence

$$\delta = (\, \delta_a \,:\, a \in \Sigma)$$

of stochastic $|Q| \times |Q|$-matrices δ_a where $\delta_a[p, q]$ is the probability that P enters state q, when reading letter a in state p. If $w = w_1 \cdots w_k \in \Sigma^*$ is a word of length k over Σ, then the matrix product

$$\delta_w[p, q] := \big(\delta_{w_1} \cdots \delta_{w_k}\big)[p, q]$$

is the probability that P reaches state q when reading w in state p. We define the acceptance probability of w for any state $p \in Q$ as

$$\mathrm{prob}[P \text{ accepts input } w \text{ with } p \text{ as initial state}] := \sum_{q \in F} \delta_w[p, q]$$

and say that P accepts w with probability $\sum_{p \in F} \delta_w[q_0, p]$. The language accepted by P is

$$L(P) := \{w \in \Sigma^* \mid P \text{ accepts } w \text{ with probability at least } \lambda\}.$$

We say that P is a $1\mathrm{P}_2\mathrm{FA}$ iff it accepts $L(P)$ with *gap* $\gamma > 0$, i.e.,

$$\left| \sum_{q \in F} \delta_w[q_0, q] - \lambda \right| \geq \gamma$$

holds for all words $w \in \Sigma^*$.

Alternating Automata. We say that $A = (Q_\forall, Q_\exists, \Sigma, \delta, q_0, F)$ is an alternating automaton (1AFA) iff A has disjoint sets $Q_\exists$, $Q_\forall$ of *existential* resp. *universal* states. For the set $Q := Q_\exists \cup Q_\forall$ of all states the transition function

$$\delta \colon Q \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}(Q)$$

has the same structure as for NFAs with ε-moves. In particular we say that $q = (q_0, \ldots, q_k) \in Q^{k+1}$ is a *computation* of A on input $w = w_1 \cdots w_n \in \Sigma^n$ iff there is a sequence $v \in (\Sigma \cup \{\varepsilon\})^*$ which results from w by possibly introducing the empty word several times such that $q_i \in \delta(q_{i-1}, v_i)$ holds for all i $(1 \leq i \leq k)$.

The notion of acceptance is defined recursively:

(a) A accepts the empty word in initial state $q \in Q$ iff q belongs to F,
(b) Let $a \in \Sigma$ and $u \in \Sigma^*$. Then A accepts au with initial state $q \in Q$ iff
  – $q \in Q_\exists$ and A accepts u for *at least one* state in $\delta(q, a)$ as initial state
  – $q \in Q_\forall$ and A accepts u for *every* state in $\delta(q, a)$ as initial state.

We say that A accepts word $w \in \Sigma^*$ iff A accepts w with initial state q_0 and set $L(A) := \{u \in \Sigma^* \mid A \text{ accepts } u\}$. A is a Σ_k-automaton iff q_0 belongs to $Q_\exists$ and A, for all words $w \in \Sigma^*$ and all computations on w, alternates at most $k-1$ times between existential and universal states. A Π_k-automaton is defined analogously.

Circuit Automata. How difficult is it to derive lower bounds on the number of states of an alternating automaton? Geffert [6] gives non-trivial lower bounds for the state complexity of languages in the polynomial hierarchy 1H. However we were unable to apply these methods to languages which are "easy" for $1\mathrm{P}_2\mathrm{FA}$s but hard for $1\Sigma_k\mathrm{FA}$s and had to work with *finite* languages. Therefore, to rephrase the above question: how difficult is it to derive non-trivial lower bounds on the number of states of alternating automata for finite languages? To at least partially answer this question we compare automata models with alternating unbounded fan-in circuits.

We fix notation first. For a directed graph $H = (U, D)$ and a node $u \in U$ we define *fan-in*(u) as the number of edges in D which are directed into u. A *circuit* C is specified by a pair $C = (G, \text{gate})$, where $G = (V, E)$ is a directed acyclic graph and gate is a function which assigns to each node $v \in V$ an input position iff fan-in$(v) = 0$, respectively a boolean operation $f\colon \{0,1\}^k \to \{0,1\}$ iff fan-in$(v) = k > 0$. The *size* of C is the number of nodes of G and its *fan-in* is the maximal fan-in of a node of V.

Remark 1. ACC is the complexity class of all boolean functions computable by unbounded fan-in circuits with AND, OR, NOT and modulo-gates in bounded depth and polynomial size. There are well known connections between bounded depth circuit classes such as ACC and automata. For instance ACC is the family of languages accepted by a nonuniform DFA (NUDFA) over a monoid that does not contain an unsolvable group as a subsemigroup. (A NUDFA accepts iff the product of the input bits belongs to a given list of monoid elements [1].) Williams [17] showed that not all languages computable in quasi-polynomial nondeterministic time belong to ACC. Super-polynomial lower bounds for languages in P however are missing.

A Σ_k-circuit (Π_k-circuit) C is a circuit composed of AND-, OR- as well as NOT-gates. It has an OR-gate (AND-gate) as its top gate and its fan-in is unbounded. We require that NOT-gates appear only at the bottom of C and that there are at most k AND- resp. OR-gates on any path in C (and hence that the depth of C is $k-1$).

Circuit automata, a restricted version of alternating automata, are tailormade to simulate Σ_k- resp. Π_k-circuits.

Definition 1 (Circuit Automata). *Let A be a $1\Sigma_k$-automaton over the binary alphabet. We say that A is a $1\Sigma_k$-circuit automaton if for all inputs w, for all computations of A on input w and for all but one input position: all state transitions of A depend only on the position of the current input bit and not on its value.*

$1\Pi_k$-circuit automata are defined analogously.

Observe that circuit automata process only binary words.

Example 1. We later investigate "nested versions" of the language

$$L_{n,1} := \{uv \mid u, v \in \{0,1\}^n,\ u = v\}$$

of equality. $L_{n,1}$ can be recognized by a Π_2-circuit $C_{n,1}$: An AND-gate as the top gate of $C_{n,1}$ "checks" for all positions i $(1 \leq i \leq n)$ whether the two disjunctions $u_i \vee \neg v_i$ and $\neg u_i \vee v_i$ hold. Observe that $C_{n,1}$ has $\mathcal{O}(n)$ gates.

We build a Π_2-circuit automaton $A_{n,1}$ from $C_{n,1}$. Starting from its *universal* initial state, $A_{n,1}$ universally selects an input position i $(1 \leq i \leq n)$ *and* one of the two disjunctions. It then decides with a single alternation whether to check u_i or v_i. It accepts iff the selected disjunction is verified to be true.

Observe that $A_{n,1}$ is indeed a circuit automaton since in any computation only one input bit (namely u_i or v_i) is checked. Moreover $\mathcal{O}(n^2)$ states suffice, since any specific input location can be found with $\mathcal{O}(n)$ states.

It turns out that Σ_k-circuit automata and Σ_k-circuits are strongly related.

Proposition 1 (Circuit Automata and Circuits).

(a) Let C be a Σ_k-circuit of size s over the input space $\{0,1\}^n$. Then C can be simulated by a Σ_k-circuit automaton with $\mathcal{O}(n \cdot s)$ states.

(b) Let A be a Σ_k-circuit automaton with s states. Then A can be simulated – for all binary inputs of length n – by a Σ_k-circuit of size $\mathcal{O}((n \cdot s)^k)$.

Proof. (a) Simulate C by a Σ_k-circuit automaton A_C which has a state q_v for any node v of C. If the gate of v is an AND-gate (OR-gate), then q_v is a universal state (existential state). The initial state of A_C is the state of the sink of C and hence the initial state is existential. A_C may transition from q_v to q_u only in an ε-move and only if (u, v) is an edge of C.

Thus initially A_C selects a path of C in a series of $k-1$ alternating ε-moves beginning with an existential initial state. If an input gate $(\neg)x_i$ is reached, then A_C travels to the ith input bit. To make this possible attach a path of i new states to the "state of" $(\neg)x_i$. Hence A_C has at most $\mathcal{O}(n \cdot s)$ states, where s is the size of C.

So far A_C has made only input-independent moves. Finally A_C accepts if the ith input bit satisfies the corresponding input gate.

(b) Let A be a Σ_k-circuit automaton with s states. W.l.o.g. we may assume that all computations of A on inputs of length n become deterministic once the single input-dependent transition is performed.

Since A performs a single input-dependent move in any computation, all computations – before performing the input-dependent move – define a single computation tree T_n for all inputs in $\{0,1\}^n$. Whenever the input-dependent move is made, we terminate the computation and hence reach a leaf in T_n.

The leaf is labeled with x_i iff the computation has reached the ith input bit and is accepting. Otherwise the leaf is labeled with $\neg x_i$. Finally, a node v of T_n receives an AND-gate if the state of v is universal and an OR-gate otherwise.

Remember that A is a Σ_k-automaton and we may assume that the root of T_n has an OR-gate. The depth of T_n may be large, however there are at most $k-1$ "alternations" between AND- and OR-gates on any path of T_n. Connect the root directly with all nodes of T_n which are reachable by a path of OR-gates only. Once this is done, compress all AND-gates at the top of the new tree as far as possible and continue this process. At the end we obtain a Σ_k-circuit T with size $\mathcal{O}((n \cdot s)^k)$. □

Remark 2. How expressive are circuit automata? We show that the state complexity of DFAs and circuit automata are incomparable.

First observe that DFAs may require exponentially more states than circuit automata. Namely, as a consequence of Proposition 1, circuit automata and circuits – with alternations and depth coinciding – turn out to be "polynomially equivalent". As we have seen in Example 1 the language $L_{n,1}$ can be recognized by a Π_2-circuit automata and hence DFAs, when simulating Π_2- or Σ_2-circuit automata, require an exponential blowup in the number of states.

On the other hand, as a consequence of Proposition 1(b), Σ_k-circuit automata, for any $k \in \mathbb{N}$, are too weak to simulate DFAs efficiently, since n-bit parity requires alternating circuits of super-polynomial size in n if depth is bounded [7].

Remark 3. One may hope to obtain lower bounds for the number of states of alternating automata for finite languages using methods from circuit complexity. However Proposition 1(a) may be strengthened to allow for threshold gates[1] as bottom gates of a Σ_k-circuit. If C is an alternating circuit of size s and depth k containing threshold-gates with weight bound m as bottom gates, then C can be simulated by a Σ_k-automaton of size $\text{poly}(n, s^k, m)$.

However no super-polynomial lower bounds for alternating circuits – with threshold gates as bottom gates – seem to exist for languages in P.

3 Separating Probabilism from Few Alternations

We define a family $L_{n,k}$ of languages which turn out to be easy for $1P_2FAs$ and 1AFAs with a sufficiently large number of alternations. However $L_{n,k}$ is shown to be hard for circuit automata with too few alternations.

[1] A threshold gate with weight bound m has integral weights $-m \leq w_0, w_1, \ldots, w_m \leq m$ and binary inputs $y_1, \ldots, y_m$. It accepts iff $\sum_{i=1}^{m} w_i y_i \geq w_0$ and rejects otherwise.

3.1 Nested Equality

The language $L_{n,k} \subseteq \{0,1\}^{(2n)^k}$ of "nested equality" is defined recursively. A bottom-up view of the definition is as follows. Assume that the input $x \in \{0,1\}^{(2n)^k}$ is partitioned into $2(2n)^{k-1}$ binary words $u_1, v_1, \ldots, u_{(2n)^{k-1}}, v_{(2n)^{k-1}}$ of respective length n with

$$x = u_1 v_1 \cdots u_{(2n)^{k-1}} v_{(2n)^{k-1}}.$$

Compress $x^{(1)} := x$ into the word $x^{(2)}$ of length $(2n)^{k-1}$ by replacing $u_i v_i$ by one if $u_i = v_i$ and by zero otherwise. Apply this compression repeatedly, each time partitioning the current word $x^{(i)}$ into consecutive words of length n, and accept x iff $x^{(k+1)} = 1$. Here is the top-down view.

Definition 2 (The Language of Nested Equality).

(a) The functions $f_{n,k} \colon \{0,1\}^{(2n)^k} \to \{0,1\}$ *are defined recursively.*

- *For* $u, v \in \{0,1\}^n$ *set* $f_{n,1}(uv) := \begin{cases} 1 & \text{if } u = v, \\ 0 & \text{otherwise.} \end{cases}$
- *For* $x \in \{0,1\}^{(2n)^k}$ *set*

$$f_{n,k}(x) := f_{n,1}\Big(f_{n,k-1}(u_1) \cdots f_{n,k-1}(u_n), f_{n,k-1}(v_1) \cdots f_{n,k-1}(v_n)\Big),$$

where $x = u_1 \cdots u_n v_1 \cdots v_n$ *with* $|u_i| = |v_i| = (2n)^{k-1}$ *for* $i = 1, \ldots, n$.

(b) Set $L_{n,k} := \{x \in \{0,1\}^{(2n)^k} \mid f_{n,k}(x) = 1\}$.

$L_{n,k}$ turns out to be easy for bounded-error automata as well as for alternating circuit automata with a sufficient number of alternations.

Lemma 1 ($1P_2FA$s and 1AFAs for $L_{n,k}$). *For any* $k \geq 1$,

(a) There are $1P_2FA$*s for* $L_{n,k}$ *with* $\mathcal{O}\big(n^{2k^2+3k}\big)$ *states and gap* $1/4$.
(b) $L_{n,k}$ *can be simulated by a* Π_{2k}*-circuit automaton with* $\mathcal{O}((2n)^{2k})$ *states.*

Proof. We associate the $(2n)$-ary tree $T_{n,k}$ of depth k with $L_{n,k}$. $T_{n,k}$ helps to visualize the hierarchical decomposition of the input bits of x into the $1 + 2n + (2n)^2 + \cdots + (2n)^{k-1} = \frac{(2n)^k - 1}{2n-1}$ equality problems.

(a) We describe probabilistic automata $P_{n,k}$ for $L_{n,k}$ with cut-point $1/2$ and gap $1/4$ recursively. The automaton $P_{n,1}$ selects a prime $p \leq N_1$ at random, where N_1 is to be determined later and checks whether $\sum_{i=1}^n 2^{n-i} u_i \equiv \sum_{i=1}^n 2^{n-i} v_i \bmod p$ holds. $P_{n,1}$ accepts iff the answer is positive. Hence $P_{n,1}$ accepts $L_{n,k}$ with $\mathcal{O}(n \cdot N_1^2)$ states. It errs only if $uv \in \{0,1\}^{2n}$ does not belong to $L_{n,1}$ and in particular if it picked a prime divisor of the difference $D := \sum_{i=1}^n 2^{n-i} u_i - \sum_{i=1}^n 2^{n-i} v_i$. Hence the error probability of $P_{n,1}$ is bounded by the quotient of the number of prime divisors of D and the total number of

primes used by $P_{n,1}$. Hence, by the Prime Number Theorem, the error for the equality problem on n bits is bounded by

$$\mathcal{O}\left(\frac{n}{N_1/\log_2 N_1}\right). \tag{1}$$

We view $P_{n,k}$ as a vector $(P_{n,1}^{(1)},\ldots,P_{n,1}^{(k)})$ of k variants $P_{n,1}^{(1)},\ldots,P_{n,1}^{(k)}$ of $P_{n,1}$. If j is the current input position, then $P_{n,1}^{(i)}$ deals with its equality problem which is specified by the node of height $i-1$ in $T_{n,k}$ which is an ancestor of the leaf containing position j. In particular, $P_{n,1}^{(i)}$ waits until $P_{n,1}^{(i-1)}$ has completed its current equality problem, computes for one step after receiving the decision of $P_{n,1}^{(i-1)}$ and then waits for the next decision. If $P_{n,1}^{(i)}$ finishes its equality problem it sends the result to $P_{n,1}^{(i+1)}$ and begins with the next equality problem.

There is a total of $\frac{(2n)^k-1}{2n-1}$ equality problems to be solved. To enforce a gap of at least $\frac{1}{4}$, we choose the upper bound N_k for primes sufficiently large. In particular $N_k=\Theta(n^{k+1})$ will do since, by (1), $P_{n,k}$ errs with probability $1-o(1)$ on none of the $\Theta(n^{k-1})$ equality problems.

The states of $P_{n,k}$ are k-tuples with the ith component corresponding to a state of $P_{n,1}^{(i)}$. The number of states of $P_{n,k}$ is hence asymptotically bounded by $(nN_k^2)^k = n^k\cdot n^{2(k+1)k} = n^{2k^2+3k}$.

(b) We construct a Π_{2k}-circuit $C_{n,k}$ for $L_{n,k}$ recursively and then apply Proposition 1(a).

The Π_2-circuit $C_{n,1}$ (with $\mathcal{O}(n)$ gates) is described in Example 1. Inputs of $C_{n,k}$ have the form $x=u_1v_1\cdots u_{(2n)^{k-1}}v_{(2n)^{k-1}}$ with $|u_i|=|v_i|=n$. We obtain circuit $C_{n,k}$ after feeding the outputs of $(2n)^{k-1}$ copies of $C_{n,1}$ into a copy of $C_{n,k-1}$, where the ith copy of $C_{n,1}$ checks whether $u_i=v_i$ holds.

With an inductive argument one may verify that circuit $C_{n,k}$ has depth $2k$ and size $\mathcal{O}((2n)^k)$. Hence we obtain a Π_{2k}-circuit automaton with $\mathcal{O}((2n)^{2k})$ states for $L_{n,k}$, if we apply Proposition 1(a). □

3.2 Circuit Automata Require Many Alternations

We show that Σ_{k+1}-circuit automata for $L_{n,k}$ require a super-polynomial number $f_k(n)$ of states.

Theorem 1. *Let $k\in\mathbb{N}$ be even and assume that $n\in\mathbb{N}$ is sufficiently large. Then any Σ_k-circuit automaton for $L_{n,k-1}$ has to have size super-polynomial in n. However polynomial size is sufficient for $\Pi_{2(k-1)}$-circuit automata.*

Proof. $L_{n,k-1}$ is accepted by $\Pi_{2(k-1)}$-circuit automata of polynomial size in n as a consequence of Lemma 1(b). Hence the claim follows from Proposition 1 (b) if we show that any Σ_k-circuit $C_{n,k}$ for $L_{n,k-1}$ has to have size super-polynomial in n, provided $k\geq 2$.

We apply a variant of the Switching Lemma for bounded depth alternating circuits [7] to the two bottom layers of $C_{n,k}$. The original Switching Lemma,

applied to circuits with N input bits, is proven by selecting a random restriction $\rho\colon [N] \to \{0,1,*\}$, where a star has probability p. Positions are fixed with a zero or a one with respective probability $(1-p)/2$.

Remember that an input x for $C_{n,k}$ has the form $x = u_1 v_1 \cdots u_{(2n)^{k-1}} v_{(2n)^{k-1}}$ for strings u_i, v_i of length n. To obtain lower size bounds for $C_{n,k}$ we place stars with probability p in $u_1 \cdots u_{(2n)^{k-1}}$ and fix the remaining positions in u_i at random. Moreover *all* positions in $v_1 \cdots v_{(2n)^{k-1}}$ are fixed such that equality $u_i = v_i$ is still possible for all i $(1 \leq i \leq (2n)^{k-1})$. A straightforward argument shows that the results of the original Switching Lemma also hold with adjusted parameters in our situation. Finally, for any i, fix all but one star in u_i such that equality remains possible. Here we utilize that fixing inputs bits does not increase size or depth of the circuit.

If s is the size of $C_{n,k}$, then $p = \Theta(\frac{1}{\log_2 s})$ will do. Hence to show that super-polynomial size is required, we may by way of contradiction assume $p = \frac{\alpha}{\log_2 n}$ for an arbitrarily large constant α. As a consequence with high probability there will be exactly one star in each u_i.

We apply the variant of the Switching Lemma, observing that all bits of v_i are fixed, and obtain with high probability that the new circuit $C_{n,k-1}$ has depth $k-1$ and that its size is polynomial in the size of $C_{n,k}$.

Observe that $C_{n,k-1}$ accepts $L_{n,k-2}$ and we may repeat this procedure. Since k is even, the circuit $C_{n,2}$ is a Σ_2-circuit. Hence there are restrictions such that there is a Σ_2-circuit for $L_{n,1}$ with size polynomial in the size of $C_{n,k}$.

Remember that an implicant of a boolean function $f\colon \{0,1\}^N \to \{0,1\}$ is a conjunction of literals which implies f. But $L_{n,1}$, interpreted as a boolean function $f\colon \{0,1\}^{2n} \to \{0,1\}$, has only implicants of length $2n$, namely one conjunction for any pair uu. But $C_{n,2}$ is a disjunction of implicants and all implicants are required to appear. Hence $C_{n,2}$ has to have size 2^n and the claim follows. □

4 Simulating Probabilism with Alternations

Our goal is to efficiently simulate a $1\mathrm{P}_2\mathrm{FA}$ $P = (Q, \Sigma, \delta, \lambda, q_0, F)$ by an 1AFA A. The constructions of Sipser, Lautemann or Canetti [2,15,16] in some shape or form require the capability of performing and evaluating many simulations of the probabilistic machine. For instance, Lautemann assumes an error probability of $2^{-\Omega(n)}$ for input size n, a requirement which in general cannot be fulfilled for $1\mathrm{P}_2\mathrm{FA}$s. We therefore proceed differently and in particular do not impose any bound on the number of alternations.

We begin by deriving a normal form for P when restricted to "short" inputs. To do so we randomly fix P-transitions to obtain DFAs $D_1, \ldots, D_r$ as well as a distribution $\mu = (\mu_1, \ldots, \mu_r)$ on the DFAs. Define the $1\mathrm{P}_2\mathrm{FA}$

$$P_r^* := \mathrm{majority}_{\lambda,\mu}(D_1, \ldots, D_r)$$

to pick the DFA D_i with probability μ_i and to simulate the input w by D_i. The input w is accepted iff the combined acceptance probabilities, summed over all accepting DFAs D_i, is at least λ.

We show that P_r^* is equivalent with P on all inputs of approximate length at most $r \cdot \gamma^2$ and that P_r^* even has a gap of at least $\gamma/2$.

Proposition 2. *Let $P = (Q, \Sigma, \delta, \lambda, q_0, F)$ be a 1P$_2$FA with gap $\gamma > 0$ and let r be a natural number. Then there are DFAs $D_1, \ldots, D_r$, each of size $s(P)$, as well as a distribution $\mu = (\mu_1, \ldots, \mu_r)$ such that $P_r^* := \text{majority}_{\lambda,\mu}(D_1, \ldots, D_r)$ and P agree on all inputs of length at most K where*

$$K = \Omega\Big(\frac{r \cdot \gamma^2}{\ln(|\Sigma|)}\Big),$$

provided $|\Sigma| \geq 2$. If $|\Sigma| = 1$, then $K = \exp^{\Omega(r \cdot \gamma^2)}$ holds. Moreover P_r^* has a gap of least $\gamma/2$ on all inputs of length at most K.

Proof. For an integer K set $W_K := \bigcup_{i=0}^{K} \Sigma^i$. We randomly select DFAs $D_i = (Q^*, \Sigma, \delta_i, q_0, F^*)$ for $i = 1, \ldots, 2r+1$, where $Q^* := Q \times [K] \cup \{q_0\}$ and $F^* = F \times [K]$ – if $q_0 \in F$, then q_0 has to be inserted into F^*. The transition functions δ_i are obtained by setting $\delta_i((p,t),a) = (q, t+1)$ with probability $\delta_a[p,q]$.

Let $\mathcal{D}$ be the set of DFAs which can be build this way and let p_D be the probability for a DFA $D \in \mathcal{D}$ to be picked. We define the distribution μ by setting $\mu_i = p_{D_i} / \sum_{j=1}^{r} p_{D_j}$ and work with the 1P$_2$FA $P_r^* := \text{majority}_{\lambda,\mu}(D_1, \ldots, D_r)$.

Fix some input $w \in W_K$. Assume that w is accepted with probability p_w by P and with probability p_w^* by P_r^*. By how much does p_w^* deviate from p_w and with which probability does that happen?

Assume first that P accepts w and as a consequence $p_w \geq \lambda + \gamma$ follows. For any path $\mathcal{P}$ in P, which starts in q_0 and is consistent with w, the probability of $\mathcal{P}$ equals the probability of a DFA $D \in \mathcal{D}$ to possess path $\mathcal{P}$. Hence p_w coincides with the probability $p_w^{\mathcal{D}}$ that a DFA $D \in \mathcal{D}$, selected with probability p_D, accepts w and $p_w^{\mathcal{D}} = p_w \geq \lambda + \gamma$ follows.

How likely is it that p_w^* is significantly smaller than p_w, i.e., that $p_w^* \leq \lambda + \gamma/2$ holds? If r DFAs are selected in $\mathcal{D}$, then $r \cdot p_w$ is the expected number of DFAs in $\mathcal{D}$ accepting w. If P_r^* errs on w or if its gap is less than $\gamma/2$, then $r \cdot p_w^*$, as the result of r independent random trials of picking a DFA from $\mathcal{D}$, deviates from its expected value $r \cdot p_w$ by a factor μ with $\mu = \frac{\lambda+\gamma/2}{p_w} \leq \frac{\lambda+\gamma/2}{\lambda+\gamma} = 1 - \frac{\gamma/2}{\lambda+\gamma}$. We apply the Chernoff bound and obtain for $|\Sigma| \geq 2$,

$$\begin{aligned}
&\text{prob}[\text{there is } w \in W_K \text{ such that } p_w^* \leq \lambda + \gamma/2 \leq \lambda + \gamma \leq p_w] \\
&\quad \leq \sum_{w \in W_K, p_w \geq \lambda+\gamma} \text{prob}[p_w^* \leq \lambda + \gamma/2] \\
&\quad \leq |W_K| \cdot \exp^{-(\frac{\gamma/2}{\lambda+\gamma})^2 \cdot r \cdot p_w/2} \leq |W_K| \cdot \exp^{-\frac{\gamma^2}{\lambda+\gamma} \cdot r/8} \qquad \text{(Chernoff bound)} \\
&\quad \leq 2 \exp^{K \ln(|\Sigma|)} \cdot \exp^{-\frac{\gamma^2}{\lambda+\gamma} \cdot r/8} = 2 \exp^{K \ln(|\Sigma|) - \frac{\gamma^2}{\lambda+\gamma} \cdot r/8}.
\end{aligned}$$

To obtain a sufficiently failure probability it suffices to (asymptotically) demand $r = \frac{\lambda+\gamma}{\gamma^2} \cdot K \ln(|\Sigma|) = \mathcal{O}\big(\frac{K \ln(|\Sigma|)}{\gamma^2}\big)$, respectively $K = \Omega\big(\frac{r \cdot \gamma^2}{\ln(|\Sigma|)}\big)$. If $|\Sigma| = 1$

we obtain, again asymptotically, $r = \frac{\lambda+\gamma}{\gamma^2} \cdot \ln(K+1) = \mathcal{O}\big(\frac{\ln(K+1)}{\gamma^2}\big)$ and hence $K = \exp^{\Omega(r\gamma^2)}$ follows.

The case that P rejects w is treated analogously. We find a majority-of-DFAs with the required properties if the probability for P_r^* to err on an input w of length at most K or to have a gap smaller than $\gamma/2$ is less than 1. The claim follows. □

Let $P = (Q, \Sigma, \delta, \lambda, q_0, F)$ be a $1\mathrm{P}_2\mathrm{FA}$ with gap γ. We just found out that P can be simulated by the weighted majority

$$P_r^* = \mathrm{majority}_{\lambda,\mu}(D_1, \ldots, D_r)$$

with gap $\gamma/2$, provided correctness is required only for inputs of approximate length $K = \Omega\big(\frac{r\cdot\gamma^2}{\ln(|\Sigma|)}\big)$. We now show how to simulate P_r^* by a 1AFA

$$A_r = (Q_\forall, Q_\exists, \Sigma, \delta_A, q_0, F_A)$$

on inputs of length at most K. Assume that q_0 is the starting state of P_r^* and $Q \times \{i\}$ is the set of states of D_i. We begin the definition of A_r by choosing q_0 as its (existential) starting state and set

$$\begin{aligned} Q_\forall &:= \{!\} \times Q \times [r] \times [r^2], \\ Q_\exists &:= \{?\} \times Q \times [r] \times [r^2] \cup \{?\} \times \{\text{continue}\} \times [r] \times [r^2] \cup \{q_0\}. \end{aligned}$$

Only universal states are accepting, since we set

$$F_A = \{!\} \times F \times [r] \times [r^2].$$

Also, insert q_0 into F_A iff P accepts the empty word. We describe the transitions of A_r next.

Let w be an arbitrary input. The alternating automaton A_r guesses which DFAs accept w and then verifies its guess each time. In particular, A_r may guess that D_i accepts w and therefore introduces for each i $(1 \le i \le r)$, the ε-transitions

$$q_0 \xrightarrow{\varepsilon} (?, q_0, i, j) \xrightarrow{\varepsilon} (!, q_0, i, j),$$

where $j := \lfloor r^2 \cdot \mu_i \rfloor$ keeps track of the probability of D_i. The ε-transition from the existential to the universal version of (q_0, i, j) is the only applicable transition and therefore A_r immediately challenges its guess. It verifies its guess by simulating D_i with $(!, q_0, i, j)$ as the starting state.

But A_r may have to search for further DFAs $D_{i'}$ accepting w and therefore introduces all ε-transitions of the form

$$(!, q_0, i, j) \xrightarrow{\varepsilon} (?, \text{continue}, i, j) \xrightarrow{\varepsilon} (?, q_0, i', j+j') \xrightarrow{\varepsilon} (!, q_0, i', j+j'),$$

where $j' := \lfloor r^2 \cdot \mu_{i'} \rfloor$ keeps track of the probability of $D_{i'}$. However, to avoid that DFAs are selected more than once, $i < i'$ has to hold. Hence, again the guess is

immediately challenged and subsequently verified. This process continues until a universal state $(!, q_0, i^*, j^*)$ is reached with $\frac{1}{r^2} \cdot j^* \geq \lambda$.

In summary, A_r accepts an input w iff there are DFAs $D_{i_1}, \ldots, D_{i_s}$ accepting w such that $i_1 < i_2 < \cdots < i_s$ and

$$\frac{1}{r^2} \cdot \sum_{t=1}^{s} \lfloor r^2 \mu_i \rfloor \geq \lambda$$

hold.

Theorem 2 (1P$_2$FAs and 1AFAs). *Assume that $P = (Q, \Sigma, \delta, \lambda, q_0, F)$ is a 1P$_2$FA with gap γ and let r be a natural number. Then there is a 1AFA A_r such that A_r and P agree on all inputs of length at most K where*

$$K = \Omega\left(\frac{r \cdot \gamma^2}{\ln(|\Sigma|)}\right),$$

provided $|\Sigma| \geq 2$. If $|\Sigma| = 1$, then $K = \exp^{\Omega(r \cdot \gamma^2)}$ holds. The size of A_r is bounded by $\mathcal{O}(r^3 \cdot s(P))$.

Proof. We apply Proposition 2 and transform the 1P$_2$FA P into the weighted majority automaton P_r^* which agrees with P on all inputs of length at most K.

We may assume that $r \geq 2/\gamma^2$, since otherwise $K < 2$ and the claim is trivial. In particular $\gamma/2 \geq \gamma^2/2 \geq 1/r$ follows. We do a case analysis.

Case 1: P_r^* accepts w. But P_r^* has gap $\gamma/2$ and hence

$$\sum_{i:D_i \text{ accepts } w} \mu_i \geq \lambda + \gamma/2. \tag{2}$$

A_r accepts w as well, since

$$\begin{aligned} \frac{1}{r^2} \cdot \sum_{i:D_i \text{ accepts } w} \lfloor r^2 \mu_i \rfloor &\geq \frac{1}{r^2} \cdot \sum_{i:D_i \text{ accepts } w} \left(r^2 \mu_i - 1\right) \\ &\geq \sum_{i:D_i \text{ accepts } w} \left(\mu_i - \frac{1}{r^2}\right) \overset{(2)}{\geq} \lambda + \gamma/2 - \frac{1}{r} \geq \lambda. \end{aligned}$$

Case 2: P_r^* rejects w. We again utilize that P_r^* has gap $\gamma/2$ and obtain

$$\sum_{i:D_i \text{ accepts } w} \mu_i \leq \lambda - \gamma/2. \tag{3}$$

This time A_r rejects w, since

$$\frac{1}{r^2} \cdot \sum_{i:D_i \text{ accepts } w} \lfloor r^2 \mu_i \rfloor \leq \frac{1}{r^2} \cdot \sum_{i:D_i \text{ accepts } w} r^2 \mu_i \overset{(3)}{\leq} \lambda - \gamma/2.$$

Finally observe that the size of A_r is asymptotically bounded by $r^3 \cdot s(P)$. □

5 Conclusions

We gather evidence that a *fixed* number of alternations is insufficient to simulate a $1\mathrm{P}_2\mathrm{FA}$ P efficiently and thus the behavior of one-way automata is quite different from unrestricted computations in, say, polynomial time or at least logarithmic space. However such an efficient simulation exists for inputs of length at most polynomial in the number $s(P)$ of states of P, if an unbounded number of alternations is allowed.

Quite a few fundamental problems remain unresolved. First, showing nontrivial lower bounds on the number of states of 1AFAs for finite languages seems to be an important extension of the lower size bounds for circuits of bounded depth. For instance we conjecture that 1AFAs with $o(k)$ alternations may recognize $L_{n,k}$ only if their size is super-polynomial in n. It certainly seems also viable to construct infinite languages which are presumably easy for $1\mathrm{P}_2\mathrm{FA}$s and hard for $1\Sigma_k\mathrm{FA}$s for some k, since then automata specific arguments may be applied.

Second, does $1P_2 \subseteq 1A$ hold? We conjecture that the answer is positive. Observe that even $1P_2 \subseteq 1H$ may hold.

Third, is it possible to simulate arbitrary $1\mathrm{P}_2\mathrm{FA}$s with positive gap efficiently by a majority-of-DFAs *without* any restriction on the size of inputs? Or, in other words, can the restriction on input length in the claim Proposition 2 be dropped? Of course a positive answer implies that $1P_2 \subseteq 1A$ holds.

Acknowledgement. Many thanks to a referee for many valuable comments.

References

1. Barrington, D.A., Thérien, D.: Finite monoids and the fine structure of NC^1. J. ACM **35**(4), 941–952 (1988)
2. Canetti, R.: More on BPP and the polynomial-time hierarchy. Inf. Process. Lett. **57**, 237–241 (1996)
3. Chandra, A.K., Kozen, D.C., Stockmeyer, L.J.: Alternation. J. ACM **28**(1), 114–133 (1981)
4. Ďuriš, P., Hromkovič, J., Jukna, S., Sauerhoff, M., Schnitger, G.: On multi-partition communication complexity. Inf. Comput. **194**(1), 49–75 (2004)
5. Dietzfelbinger, M., Hromkovič, J., Schnitger, G.: A comparison of two lower-bound methods for communication complexity. Theor. Comput. Sci. **168**(1), 39–51 (1996)
6. Geffert, V.: An alternating hierarchy for finite automata. Theor. Comput. Sci. **445**, 1–24 (2012)
7. Håstad, J.: Almost optimal lower bounds for small depth circuits. In: STOC, pp. 6–20 (1986)
8. Hromkovič, J., Seibert, S., Karhumäki, J., Klauck, H., Schnitger, G.: Communication complexity method for measuring nondeterminism in finite automata. Inf. Comput. **172**(2), 202–217 (2002)
9. Hromkovič, J., Schnitger, G.: On the power of Las Vegas for one-way communication complexity, OBDDs, and finite automata. Inf. Comput. **169**(2), 284–296 (2001)
10. Hromkovič, J., Schnitger, G.: Nondeterministic communication with a limited number of advice bits. SIAM J. Comput. **33**(1), 43–68 (2003)

11. Hromkovič, J., Schnitger, G.: Comparing the size of NFAs with and without epsilon-transitions. Theor. Comput. Sci. **380**(1–2), 100–114 (2007)
12. Hromkovič, J., Petersen, H., Schnitger, G.: On the limits of the communication complexity technique for proving lower bounds on the size of minimal NFA's. Theor. Comput. Sci. **410**(30–32), 2972–2981 (2009)
13. Hromkovič, J., Schnitger, G.: Ambiguity and communication. Theory Comput. Syst. **48**(3), 517–534 (2011)
14. Kapoutsis, C.A.: Size complexity of two-way finite automata. In: Diekert, V., Nowotka, D. (eds.) DLT 2009. LNCS, vol. 5583, pp. 47–66. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02737-6_4
15. Lautemann, C.: BPP and the polynomial hierarchy. Inf. Process. Lett. **17**, 215–217 (1983)
16. Sipser, M.: A complexity theoretic approach to randomness. In: ACM Symposium on Theoretical Computer Science, pp. 330–335 (1983)
17. Williams, R.: Non-uniform ACC circuit lower bounds. In: IEEE Conference on Computational Complexity, pp. 115–125 (2011)