Available online at www.sciencedirect.com
SCIENCE DIRECT®
Theoretical Computer Science 330 (2005) 135–144
Theoretical Computer Science
www.elsevier.com/locate/tcs

On the power of randomized multicounter machines☆

Juraj Hromkovič[a,*], Georg Schnitger[b]

[a] *Department of Computer Science, Swiss Federal Institute of Technology ETH Zurich, ETH Zentrum, RZ F2, CH-8092 Zurich, Switzerland*
[b] *Institut für Informatik, Johann Wolfgang Goethe University, Robert Mayer Straße 11–15, 60054 Frankfurt am Main, Germany*

Abstract

One-way two-counter machines represent a universal model of computation. Here we consider the polynomial-time classes of multicounter machines with a constant number of reversals and separate the computational power of nondeterminism, randomization and determinism. For instance, we show that polynomial-time one-way multicounter machines, with error probability tending to zero with growing input length, can recognize languages that cannot be accepted by polynomial-time nondeterministic two-way multicounter machines with a bounded number of reversals. A similar result holds for the comparison of determinism and one-sided-error randomization, and of determinism and Las Vegas randomization.

Keywords: Problem complexity; Randomness; Determinism; Nondeterminism; Multicounter machines

1. Introduction

Although randomization is by now a standard tool for making computations and communication more efficient or for building simpler systems, we are far from fully understanding the power of randomized computing. Hence it is advisable to study randomization for

☆ The work of this paper has been supported by the DFG Projects HR 14/6-1 and SCHN 503/2-1. This is an essentially revisited and extended version of results presented at ICALP'03.

* Corresponding author.
E-mail address: jh@cs.rwth-aachen.de (J. Hromkovič).

0304-3975/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2004.09.015

restricted models of computation. This research has started with the study of simple models like one-way finite automata and two-party communication protocols and continues by investigating the power of randomization for more and more complex models of computation.

The goal of this paper is to establish new results separating randomization from determinism and nondeterminism as well as to contribute to the development of proof techniques for this purpose. The computing models considered here are multicounter machines.

To separate nondeterminism, randomization and determinism for polynomial-time computation is probably the central question of theoretical computer science. Because of the enormous hardness of this problem many researchers try to separate determinism from randomization and randomization from nondeterminism at least for restricted models of computations (see, for instance, [2–7,9–13,15,17–19,21,23–25]) in order to gain further insight into the computational power of these modes of computation.

Polynomial-time one-way multicounter machines are one of the most powerful computing models for which one tries to separate determinism, randomization and nondeterminism. It is a well-known fact that even one-way two-counter machines can simulate Turing machines and so multicounter machines represent a universal machine model. We consider the stronger model of two-way multicounter machines with a constant number of reversals.

In what follows, let **mcm** denote a multicounter machine and let **1mcm** denote a one-way mcm. If we speak about reversals we always mean the reversals of the reading head on the input tape.

In this paper we succeed in answering most of the basic questions about the relative power of determinism, randomization and nondeterminism for polynomial-time one-way (two-way with a constant number of reversals) multicounter machines. Let **1DMC(poly) [1NMC(poly)]** be the class of languages accepted by polynomial-time one-way deterministic [nondeterministic] multicounter machines. Let **2cDMC(poly) [2cNMC(poly)]** denote the class of languages accepted by deterministic [nondeterministic] two-way mcm with a constant number of reversals.

Definition 1. Let A be a randomized mcm with three final states q_{accept}, q_{reject} and q_{neutral}. We say that A is a **Las Vegas mcm (LVmcm)** recognizing a language L if the following conditions hold:

(i) For each $x \in L$, $\text{Prob}(A \text{ accepts } x) \geqslant \frac{1}{2}$ and $\text{Prob}(A \text{ rejects } x) = 0$.

(ii) For each $x \notin L$, $\text{Prob}(A \text{ rejects } x) \geqslant \frac{1}{2}$ and $\text{Prob}(A \text{ accepts } x) = 0$.

We say that A is a **one-sided-error Monte Carlo mcm, Rmcm** for L iff

(iii) For each $x \in L$, $\text{Prob}(A \text{ accepts } x) \geqslant \frac{1}{2}$.

(iv) For each $x \notin L$, $\text{Prob}(A \text{ rejects } x) = 1$.

We say that A is a **bounded-error probabilistic mcm, BPmcm** for L, if there is an ε such that

(v) For each $x \in L$, $\text{Prob}(A \text{ accepts } x) \geqslant \frac{1}{2} + \varepsilon$.

(vi) For each $x \notin L$, $\text{Prob}(A \text{ rejects } x) \geqslant \frac{1}{2} + \varepsilon$.

We denote by 1LVMC(poly)[1RMC(poly), 1BPMC(poly)] the class of languages accepted by a polynomial-time one-way LVmcm [Rmcm, BPmcm]. Let 2cLVMC(poly)

[2cRMC(poly), 2cBPMC(poly)] denote the class of languages accepted by polynomial-time two-way LVmcm [Rmcm, BPmcm] with a constant number of reversals.

All probabilistic classes possess amplification: We can reduce the error arbitrarily by simulating independent runs in parallel with an appropriately increased number of counters. Here the interesting question is whether an error probability tending to zero is reachable. Therefore for any probabilistic class A we define the class

$$A^* = \{L(M) \mid \text{M is a machine of type A with error probability tending towards 0 with increasing input length}\}.$$

(In the case of Las Vegas randomization we consider the probability of giving the answer "?" as error probability.) We obtain the following separations:

(a) Bounded-error randomization and nondeterminism are incomparable, since $1\text{BPMC}^*(\text{poly}) - 2\text{cNMC}(\text{poly}) \neq \emptyset$ and $1\text{NMC}(\text{poly}) - 2\text{cBPMC}(\text{poly}) \neq \emptyset$.

(b) $1\text{BPMC}^*(\text{poly}) - 2\text{cRMC}(\text{poly}) \neq \emptyset$,
i.e., bounded error randomization with an arbitrary small error is more powerful than one-sided-error randomization.

(c) $1\text{RMCM}^*(\text{poly}) - 2\text{cLVMC}(\text{poly}) \neq \emptyset$,
i.e., one-sided-error randomization is more powerful than Las Vegas randomization, and

(d) $2\text{cLVMC}^*(\text{poly}) - 2\text{cDMC}(\text{poly}) \neq \emptyset$ and
$2\text{cLVMC}^*(2^{\text{O}(\sqrt{n}\log^2 n)}) - 2\text{cDMC}(2^{\text{o}(n)}) \neq \emptyset$,
i.e., Las Vegas randomization is more powerful than determinism.

These results show a proper hierarchy between LVmcc, Rmcc and BPmcc resp. nondeterministic mcc, where the weaker computation mode cannot reach the stronger mode, even when restricting the stronger mode to one-way computations and additionally demanding error probability approaching zero. The proof even shows that allowing $\text{o}(n/\log n)$ reversals on inputs of size n does not help the weaker mode.

It is not unlikely that determinism and Las Vegas randomization are equivalent for one-way computations. However, the separation $2\text{cLVMC}^*(2^{\text{O}(\sqrt{n}\log^2 n)}) - 2\text{cDMC}(2^{\text{o}(n)}) \neq \emptyset$ also holds for $\text{o}(n/\log n)$ reversals of the deterministic machine.

2. Preliminaries

Before presenting our results we give some basic knowledge about elementary actions that can be efficiently executed by one-way multicounter machines. Let, for any counter C, $num(C)$ denote the size of the counter (i.e., the nonnegative integer represented by the unary content of the counter).

Fact 2. *Let A be a 1mcm with 5 counters $C_1, C_2, \ldots, C_5$. Then A can compute $num(C_1) \cdot num(C_2)$ without loosing the values $num(C_1)$ and $num(C_2)$ in time $\text{O}(num(C_1) \cdot num(C_2))$.*

Fact 3. *Let A be a 1mcm with 5 counters $C_1, C_2, \ldots, C_5$. Then A can compute*

$$num(C_1) \bmod num(C_2)$$

and

$$num(C_1) \operatorname{div} num(C_2)$$

without loosing the values $num(C_1)$ *and* $num(C_2)$ *in time* $O(num(C_1))$.

Let $Number(x)$ denote the nonnegative integer whose binary representation is x^R.

Fact 4. *Let A be a 1mcm with at least* 3 *counters. If A has a word* $x \in \{0,1\}^*$ *on the input tape, then A can compute and save the number* $Number(x)$ *in time* $O(Number(x))$.

Lemma 5. *Let A be a 1mcm with at least* 4 *counters* C_1, C_2, C_3, *and* C_4. *Let a word* $x \in \{0,1\}^*$ *be on the input tape of A. Then A can compute the integer*

$$Number(x) \bmod num(C_1)$$

in time $O(|x|\, num(C_1))$.

Proof. The computation of A is based on the fact that, for all positive integers i and k,

$$2^{i+1} \bmod k = 2(2^i \bmod k) \bmod k.$$

Thus, if a counter contains the value $2^i \bmod num(C_1)$, then doubling its content and computing modulo $num(C_1)$ the 1mcm A computes the value $2^{i+1} \bmod num(C_1)$. In this way, starting with the least significant bit of x_1 of $x = x_1, \ldots, x_n$ the machine A can compute the value

$$\left(\sum_{i=1}^{n} x_i 2^{i-1}\right) \bmod num(C_1),$$

as the value

$$\left(\sum \left(x_i 2^{i-1}\right) \bmod num(C_1)\right) \bmod num(C_1). \qquad \square$$

Lemma 6. *Let A be a randomized 1mcm with counters* C_1, C_2, C_3, C_4, C_5. *Let* $n = num(C_1)$ *and* $n^2 = num(C_2)$ *for a positive integer n. Then without any movement on the input tape, A can generate and save a random prime from* $\{2, 3, \ldots, n^2\}$ *or enter a special state saying "I was not successful in generating a prime" in time* $O(n^5)$ *with a probability of success above* $1 - e^{-n^2/2 \ln n}$ *for sufficiently large n.*

Proof. First of all we observe that A can generate a random number from $\{2, 3, \ldots, n^2\}$ in time $O(n^2)$. A simply computes and saves, consecutively, the values $2^0, 2^1, 2^2, \ldots, 2^{\lceil \log_2 (n^2+1) \rceil}$ by doubling the previous value. For any 2^i A toss a coin in order to decide whether 2^i has to be summed to the created number. When the value 2^j is larger then n^2, then A stops this part of this procedure, which can be performed in $O(n^2)$ time.

After that A deterministically verifies whether the generated random number $m \in \{2, 3, \ldots, n^2\}$ is a prime by computing

$$m \bmod r$$

for all $r \in \{2, 3, \ldots, n\}$. Following Fact 3, this can be done in time $O(nm) \subseteq O(n^3)$.

If m is a prime, then A halts. If not, A tries again to generate a new random number, but A does it at most n^2 times. After counting n^2 unsuccessful attempts A finishes in a special state.

The Prime Number Theorem says that there are approximately $n^2/2 \ln n$ primes smaller than n^2. Hence the probability of generating a prime in one attempt is

$$\frac{1}{2 \ln n}.$$

The probability to be not successful in $2n^2$ attempts is then

$$\left(1 - \frac{1}{2 \ln n}\right)^{2n^2} = \left(\left(1 - \frac{1}{2 \ln n}\right)^{2 \ln n}\right)^{n^2/2 \ln n} \leqslant e^{-n^2/2 \ln n}. \qquad \square$$

3. Main results

Our first two results compare nondeterminism and randomness. Let

$$EQ = \{0^n \# w \# w \mid w \in \{0, 1\}^n, n \in \mathbb{N}\}.$$

Theorem 7. $EQ \in 2BPMC^*(\text{poly}) - 2cNMC(\text{poly})$.

Proof. First, we show that $EQ \in$2BPMC*(poly), by describing a 1mcm M that accepts EQ with an error probability tending to zero with the input length. For any input $0^n \# w \# y$ the 1mcm M works as follows. Reading 0^n it saves the value n in a counter and the value n^2 in another counter (by computing $n \cdot n$ (Fact 2)). Following the strategy described in Lemma 6 M generates a random prime p from $\{2, 3, \ldots, n^2\}$ with a probability at least $1 - e^{-n^2/2 \ln n}$ and stops with a probability at most $e^{-n^2/2 \ln n}$ without deciding about the membership of the input in EQ.

Reading the input part (suffix) $\# w \# y$ the machine M computes the values

$$\mathit{Number}(w) \bmod p \text{ and } \mathit{Number}(y) \bmod p$$

in time $O(n^3)$ by the strategy described in Lemma 5. Simultaneously, M checks whether $n = |w| = |y|$ and if not then M rejects the input.

If $\mathit{Number}(w) \bmod p = \mathit{Number}(y) \bmod p$, then M accepts and reject otherwise.

Now, let us analyse the error probability of M. We distinguish two possibilities with respect to the membership of the input in EQ.

(i) Let $0^n \# w \# y \in EQ$, i.e., $w = y$.

Then $Number(w) \bmod p = Number(y) \bmod p$ for every positive integer p and so M accepts with certainty.

(ii) Let $0^n\#w\#y \in EQ$.

If $n = |w| = |y|$ does not hold, then M rejects with certainty. When $n = |w| = |y|$ and $w \neq y$, then M can err when

$$Number(w) \bmod p = Number(y) \bmod p. \tag{1}$$

Let us bound the number or primes leading to the wrong decision. If (1) holds, then p divides the number

$$d = |Number(w) - Number(y)|.$$

But $d < 2^n$ and so there are at most $n-1$ different primes in the factorisation of d (for details, see for instance [8]). Since, due to the Prime Number Theorem we know that the number of primes smaller than n^2 at least

$$n^2/2 \ln n$$

for $n \geqslant 9$, the error probability is bounded by

$$\frac{n-1}{n^2/2 \ln n} < \frac{2 \ln n}{n}.$$

Hence, M is successful with a probability at least

$$\left(1 - e^{-n^2/2 \ln n}\right)\left(1 - \frac{2 \ln n}{n}\right)$$

that tends to 1 with growing n.

Thus, we have proved that $EQ \in 2BPMC^*(poly)$.

To show that $EQ \notin$ 2cNMC(poly) we use an argument from communication complexity theory. Assume the opposite, i.e., that there is a polynomial-time nondeterministic mcm D that accepts EQ and uses at most c reversals in any computation. Let D have k counters for a positive integer k and let D work in time at most n^r, $r \in \mathbb{N}$, for any input of length n. Consider the work of D on an input $O^n\#x\#y$ with $|x| = |y| = n$. D is always in a configuration where the content of each counter is bounded by n^r. Each such configuration can be represented by a sequence of $O(kr \log_2 n)$ bits and so the whole crossing sequence on any position can be stored by $O(ckr \log_2 n)$ bits. Thus, D can be simulated by a nondeterministic communication protocol that accepts EQ within communication complexity $O(\log_2 n)$. This contradicts the well-known fact that the nondeterministic communication complexity of EQ is in $\Omega(n)$ [1,8,16]. □

For showing that nondeterminism can be more powerful than bounded-error randomness, we consider the nondisjointness problem defined by

$$NDIS = \{x\#y \mid x, y \in \{0,1\}^n \text{ for an } n \in \mathbb{N} \text{ and } \exists j : x_j = y_j = 1\}.$$

Theorem 8. $NDIS \in 1NMC(poly) - 2cBPMC(poly)$.

Proof. By guessing a position j with $x_j = y_j = 1$ a nondeterministic 1mcm can accept *NDIS* with one counter in linear time.

The fact $NDIS \notin 2cBPMC(poly)$ can be proved by contradiction as follows. Assume there is a 2cBPmcm that accepts *NDIS*. Then similarly as in the proof of Theorem 7 one can construct a sequence of bounded-error two-party protocols that accept *NDIS* within communication complexity $O(\log_2 n)$. But this contradicts the result of [14,22] that the communication complexity of *NDIS* is in $\Omega(n)$. □

Observe that the lower bounds of Theorems 7 and 8 even work when allowing $o(n/\log n)$ reversals. Hence, Theorem 7 shows that bounded-error randomization with error probability approaching zero cannot be compensated for by nondeterminism and $o(n/\log n)$ increase of the allowed number of reversals and Theorem 8 shows that one-way nondeterminism cannot be compensated for by bounded-error randomization with $o(n/\log n)$ reversals.

To separate one-sided error from Las Vegas we consider the language

$$NEQ = \{0^n\#x\#y \mid n \in \mathbb{N}, x, y \in \{0,1\}^n, x \neq y\},$$

which can be viewed as a complement of *EQ*.

Theorem 9. $NEQ \in 1RMC^*(poly) - 2cLVMC(poly)$.

Proof. To recognize *EQ* by a one-sided-error 1mcm M one can use almost the same randomized 1mcm as for *NEQ*. The only difference is that M rejects the input when it was not successful in generating a prime. Then M rejects with certainty all inputs $w \notin NEQ$, and for every input w from *NEQ* M accepts w with probability tending to zero with growing input length.

The membership of *NEQ* in 2cLVMC(poly) would imply the existence of Las Vegas two-party protocols accepting *NEQ* within communication complexity $O(\log_2 n)$. This would contradict to the lower bound $\Omega(n)$ [20] on the Las Vegas communication complexity of *NEQ*. Hence, $NEQ \notin 2cLVMC(poly)$. □

Since an one-sided-error mcm is a special version of a nondeterministic 1mcm, Theorems 7–9 yield

$$1DMC(poly) \subseteq 1LVMC(poly) \subset 1RMC(poly) \subset 1NMC(poly)$$

and

$$1DMC(poly) \subset 1RMC(poly) \subset 1BPMC(poly).$$

Clearly, these hierarchies may be formulated for two-way mcm machines with distinct bounds on the number of reversals as well as for $*$-randomized classes. The only relation we were not able to fix is the relation between determinism and Las Vegas randomization for polynomial-time one-way mcm. We let it as an open problem here. But, we are able to establish the following separations between Las Vegas and determinism.

Theorem 10. *There exist a language* $L \subseteq \{0,1,\#\}^*$ *such that*

(i) *L can be recognized by a LVmcm in time* $2^{O(\sqrt{n}\log n)}$ *with one reversal, and*

(ii) *each deterministic mcm that accepts L with a constant number of reversals must work in time* $2^{\Omega(n)}$.

Proof. Consider the language

$$L = \{\, w_1\#w_2\#\cdots\#w_m\#\#y_1\#y_2\#\cdots\#y_m \mid m \in \mathbb{N}\text{-}\{0\},$$

$$w_i, y_i \in \{0,1\}^m \quad \text{for } i = 1,\ldots,m \text{ and } \exists j : w_j = y_j \,\}.$$

First, we outline how to construct a LVmcm M that accepts L in time $2^{(\sqrt{n}/\log n)}$. Let $x \in \{0,1,\#\}^*$ be an input. Since the fact whether

$$x = w_1\#w_2\#\cdots w_m\#\#y_1\#y_2\#\cdots\#y_m \text{ and } x_i, y_i \in \{0,1\}^m \text{ for } i = 1,\ldots,m$$

can be verified in one run of M from the left to the right in linear time, let us focus on the work of M on words having this form. Similarly, as in the proof of Theorem 7 M performs at most m^2 attempts to randomly generate a prime smaller than m^3. If M does not succeed, than it will stop in the state q_{neutral}. If M generates a prime p, then M reading $w_1\#w_2\#,\ldots,\#w_m\#$ computes m numbers

$$a_i = \mathit{Number}(w_i) \bmod p \text{ for } i = 1,\ldots,m.$$

All these numbers $a_1,\ldots,a_m$ can be saved unary in a counter of size $2^{2m\lceil\log_2(m+1)\rceil}$. The crucial fact is that M can reconstruct the binary representation of all a_is in time that is linear in $2^{2m\lceil\log_2(m+1)\rceil}$. M does it when reading $y_1\#y_2\#,\ldots,\#y_m$ and compares a_i with $\mathit{Number}(y_i) \bmod p$ for every $i \in \{1,2,\ldots,m\}$. If $a_i \neq \mathit{Number}(y_i) \bmod p$ for all $i \in \{1,\ldots,m\}$, then M rejects the input x.

If M found a $j \in \{1,\ldots,m\}$ such that $a_j = \mathit{Number}(y_i) \bmod p$, then M saves $\mathit{Number}(y_j)$ in a counter. Observe, that the unary representation of y_j is of the size 2^m. Then M reverses the direction of the head and moves to w_j in order to check whether $w_j = y_j$. If $w_j = y_j$, then M accept x. If $w_j \neq y_j$, then M finishes the computation in q_{neutral}.

Since $n = |x| = m(m+1)$, M works in time $2^{O(\sqrt{n}\log n)}$. Clearly, M never errs. The following probabilistic analysis shows that the probability to reach q_{neutral} tends to zero with the growth of the input length n.

The probability that M stops because it was not successful in generating a prime is (as already observed in the previous proofs and in Lemma 6) negligible. As usual, we distinguish two cases with respect to the membership of x in L.

(i) Let $x \in L$.

Then $w_i \neq y_i$ for all $i \in \{1,2,\ldots,m\}$. If

$$\mathit{Number}(w_i) \equiv \mathit{Number}(y_i) \bmod p \tag{2}$$

for at least one $i \in \{1,2,\ldots,m\}$, then M stops in the state q_{neutral}. In the opposite case, M correctly rejects x.

Let us calculate the probability p_i that (2) happens for a fixed position i. Since we have at least $m^3/3 \ln m$ primes smaller than m^3 for $m \geqslant 9$ and at most $m-1$ primes with property (2),

$$p_i \leqslant \frac{m-1}{m^3/3 \ln m} < \frac{3 \ln m}{m^2}.$$

Let p_{neut} be the probability that (2) happens for at least one position $i \in \{1, 2, \ldots, m\}$. Clearly,

$$p_{\text{neut}} \leqslant \sum_{i=1}^{m} p_i < \sum_{i=1}^{m} \frac{3 \ln m}{m^2} = \frac{3 \ln m}{m}.$$

Thus, p_{neut} is tending to zero with growing input length.

(ii) Let $x \in L$.

Now, let j be the smallest integer from $\{1, \ldots, m\}$ such that

$$w_j = y_j.$$

Then M stops in q_{neutral} iff (2) happens for some $i \in \{1, \ldots, j-1\}$. But the probability of this event is at most $\sum_{i=1}^{j-1} p_i$, which is smaller than p_{neut}.

Thus, M is a LVmcm accepting L.

To prove that $2^{\Omega(n)}$ deterministic time is necessary to accept L by a mcm with a constant number of reversals, one can again use arguments from communication complexity theory, because it is known that the communication complexity of L is in $\Omega(n)$. □

Theorem 11. *$2cLVMC^*(poly) - 2cDMC(poly) \neq \emptyset$, i.e., Las Vegas randomization is more powerful that determinism for polynomial time multicounter machines with constant number of reversals.*

Proof. Consider the language

$$\begin{aligned} L_{\text{pad}} = \{\, & 0^n \,\#\#\# \, w_1 \# w_2 \# \cdots \# w_m \# \# y_1 \# y_2 \# \cdots \# y_m \mid \\ & n \in \mathbb{N} - \{0\}, m = \log_2 n / \log_2 \log_2 n, \\ & w_i, y_i \in \{0,1\}^m \text{ for } i \in 1, \ldots, m \text{ and } \exists j : w_j = y_j \} \end{aligned}$$

which can be viewed as padding the language L with exponentially many dummy symbols. Now, using the same calculation as in the proof of Theorem 10 the result follows. □

References

[1] A.V. Aho, J.E. Hopcroft, M. Yannakakis, On notions of information transfer in VLSI circuits, in: Proceedings of the 15th Annual ACM STOCS, ACM, 1983, pp. 133–139.

[2] L. Babai, Monte Carlo algorithms in graph isomorphism techniques, Research Report no. 79-10, Département de mathématiques et statistique, Université de Montréal, 1979.

[3] M. Dietzfelbinger, M. Kutylowski, R. Reischuk, Exact lower bounds for computing Boolean functions on CREW PRAMs, J. Comput. System Sci. 48 (1994) 231–254.

[4] P. Ďuriš, J. Hromkovič, K. Inone, A separation of determinism, Las Vegas and nondeterminism for picture recognition, in: Proceedings of the IEEE Conference on Computational Complexity, IEEE, 2000, pp. 214–228. Full Version: Electronic Colloquium on Computational Complexity, Report no. 27, 2000.
[5] P. Ďuriš, J. Hromkovič, J.D.P. Rolim, G. Schnitger, Las Vegas versus determinism for one-way communication complexity, finite automata and polynomial-time computations, in: Proceedings of the STACS'97, Lecture Notes in Computer Science, Vol. 1200, Springer, Berlin, 1997, pp. 117–128.
[6] R. Freivalds, Projections of languages recognizable by probabilistic and alternating multitape automata, Inform. Process. Lett. 13 (1981) 195–198.
[7] J. Gill, Computational complexity of probabilistic Turing machines, SIAM J. Comput. 6 (1977) 675–695.
[8] J. Hromkovič, Communication Complexity and Parallel Computing, Springer, Berlin, 1997.
[9] J. Hromkovič, Communication protocols—an exemplary study of the power of randomness, in: P. Pardalos, S. Kajasekaran, J. Reif, J. Rolim (Eds.), Handbook on Randomized Computing, Vol. 2, Kluwer Publisher, Dordrecht, 2001, pp. 533–596.
[10] J. Hromkovič, M. Sauerhoff, Tradeoffs between nondeterminism and complexity for communication protocols and branching programs, in: Proceedings of the STACS 2000, Lecture Notes in Computer Science, Vol. 1770, Springer, Berlin, 2000, pp. 145–156.
[11] J. Hromkovič, G. Schnitger, On the power of Las Vegas II, Two-way finite automata, in: Proceedings of the ICALP'99, Lecture Notes in Computer Science, Vol. 1644, Springer, Berlin, 1999, pp. 433–443 (extended version: Theoret. Comput. Sci. 262 (2001) 1–14).
[12] J. Hromkovič, G. Schnitger, On the power of Las Vegas for one-way communication complexity, OBDD's and finite automata, Inform. Comput. 169 (2001) 281–296.
[13] J. Hromkovič, G. Schnitger, On the power of randomized pushdown automata, in: Proceedings of the DLT'2001, Lecture Notes in Computer Science, Vol. 1770, Springer, Berlin, 2002, pp. 262–271.
[14] B. Kalyanasundaram, G. Schnitger, The probabilistic communication complexity of set intersection, SIAM J. Discrete Math. 5 (4) (1992) 545–557.
[15] J. Kaneps, D. Geidmanis, R. Freivalds, Tally languages accepted by Monte Carlo pushdown automata, in: RANDOM'97, Lecture Notes in Computer Science, Vol. 1269, pp. 187–195.
[16] E. Kushilevitz, N. Nisan, Communication Complexity, Cambridge University Press, Cambridge, 1997.
[17] I. Macarie, On the structure of log-space probabilistic complexity classes, Technical Report TR-506, Dept. of Computer Science, University of Rochester, 1994.
[18] I. Macarie, M. Ogihara, Properties of probabilistic pushdown automata, Technical Report TR-554, Dept. of Computer Science, University of Rochester, 1994.
[19] I.I. Macarie, J.I. Seiferas, Strong equivalence of nondeterministic and randomized space-bounded computations, Manuscript, 1997, Later version: Amplification of slight probabilistic advantage at absolutely no cost in space, Inform. Process. Lett. 72 (1999) 113–118.
[20] K. Mehlhorn, E. Schmidt, Las Vegas is better than determinism in VLSI and distributed computing, in: Proceedings of the 14th ACM STOC'82, ACM, 1982, pp. 330–337.
[21] Ch. Papadimitrou, M. Sipser, Communication complexity, in: Proceedings of the 14th ACM STOC, ACM, 1982, pp. 196–200; also in J. Comput. System Sci. 28 (1984) 260–269.
[22] A.A. Razborov, On the distributional complexity of disjointness, Theoret. Comp. Sci. 106 (2) (1992) 385–390.
[23] M. Sauerhoff, On nondeterminism versus randomness for read-once branching programs, Electronic Colloquium on Computational Complexity TR 97 - 030, 1997.
[24] M. Sauerhoff, On the size of randomized OBDDs and read-once branching programs for *k*-stable functions, in: Proceedings of the STACS'99, Lecture Notes in Computer Science, Vol. 1563, Springer, Berlin, 1999, pp. 488–499.
[25] A.C. Yao, Some complexity questions related to distributed computing, in: Proceedings of the 11th ACM STOC, ACM, 1979, pp. 209–213.