On Pseudorandomness and Resource-Bounded Measure

Vikraman Arvind
Institute of Mathematical Sciences Madras

Johannes Köbler
Universität Ulm

Nr. 97-05
Ulmer Informatik-Berichte
März 1997
On Pseudorandomness and Resource-Bounded Measure

V. Arvind
Institute of Mathematical Sciences
C. I. T. Campus
Madras 600 113, India

Johannes Köbler
Abteilung Theoretische Informatik,
Universität Ulm,
D-89069 Ulm, Germany

Abstract

In this paper we extend a key result of Nisan and Wigderson to the nondeterministic setting: for all \( \alpha > 0 \) we show that if there is a language in \( E = \text{DTIME}(2^\alpha(n)) \) that is hard to approximate by nondeterministic circuits of size \( 2^{\alpha n} \), then there is a pseudorandom generator that can be used to derandomize \( \text{BP} \cdot \text{NP} \) (in symbols, \( \text{BP} \cdot \text{NP} = \text{NP} \)). By applying this extension we are able to answer some questions left open by Lutz regarding the derandomization of the classes \( \text{BP} \cdot \Sigma_k^p \) and \( \text{BP} \cdot \Theta_k^p \) under plausible measure theoretic assumptions:

- For all \( k \geq 2 \), if \( \mu_p(\Delta_k^p) \neq 0 \), then \( \text{BP} \cdot \Sigma_k^p = \Sigma_k^p \).
- For all \( k \geq 2 \), if \( \mu_p(\Theta_k^p) \neq 0 \), then \( \text{BP} \cdot \Theta_k^p = \Theta_k^p \).
- If \( \mu_p(\text{NP}) \neq 0 \), then \( \text{BP} \cdot \text{NP} \subseteq \text{NP}/\log \).
- If \( \mu_p(\text{NP} \cap \text{coNP}) \neq 0 \), then \( \text{BP} \cdot \text{NP} = \text{NP} \).

As a consequence, if \( \Theta_2^p \) does not have \( p \)-measure 0, then \( \text{AM} \cap \text{coAM} \) is low for \( \Theta_2^p \). Thus, in this case, \( \text{BPP} \) and the graph isomorphism problem are low for \( \Theta_2^p \). By using the Nisan-Wigderson design of a pseudorandom generator we unconditionally show the inclusion \( \text{MA} \subseteq \text{ZPP}^{\text{NP}} \) and that \( \text{MA} \cap \text{coMA} \) is low for \( \text{ZPP}^{\text{NP}} \).

1 Introduction

In recent years, following the development of resource-bounded measure theory, pioneered by Lutz in [Lut92, Lut93], plausible complexity-theoretic assumptions like \( \text{P} \neq \text{NP} \)
have been replaced by the stronger, but arguably plausible measure-theoretic assumption \( \mu_p(\text{NP}) \neq 0 \). With this stronger assumption as hypothesis, a number of interesting complexity-theoretic conclusions have been derived, which are not known to follow from \( P \neq \text{NP} \). Two prominent examples of such results are: there are Turing-complete sets for NP that are not many-one complete [LM94], there are NP problems for which search does not polynomial-time reduce to decision [LM94, BG94].

Recently, Lutz [Lut96] has shown that the hypothesis \( \mu_p(\text{NP}) \neq 0 \) (in fact, the possibly weaker hypothesis \( \mu_p(\Delta^P_2) \neq 0 \)) implies that \( \text{BP} \cdot \Delta^P_k = \Delta^P_k \), \( k \geq 2 \) (in other words, \( \text{BP} \cdot \Delta^P_k \) can be derandomized). This has an improved lowness consequence: it follows that if \( \mu_p(\Delta^P_2) \neq 0 \) then \( \text{AM} \cap \text{coAM} \) is low for \( \Delta^P_2 \) (i.e., any \( \text{AM} \cap \text{coAM} \) language is powerless as oracle to \( \Delta^P_2 \) machines). It also follows from \( \mu_p(\Delta^P_2) \neq 0 \) that if \( \text{NP} \subseteq \text{P/poly} \) then \( \text{PH} = \Delta^P_2 \). Thus the results of Lutz’s paper [Lut96] have opened up a study of derandomization of randomized complexity classes and new lowness properties under assumptions about the resource-bounded measure of different complexity classes.

The results of Lutz in [Lut96] (and also a preceding paper [Lut93]) are intimately related to research on derandomizing randomized algorithms based on the idea of trading hardness for randomness [Sha81, Yao82, NW94]. In particular, Lutz makes essential use of the explicit design of a pseudorandom generator that stretches a short random string to a long pseudorandom string that looks random to deterministic polynomial-size circuits. More precisely, the Nisan-Wigderson generator is built from a set (assumed to exist) that is in \( \text{E} \) and, for some \( \alpha > 0 \), is hard to approximate by circuits of size \( 2^{\alpha n} \). As shown in [NW94], such a pseudorandom generator can be used to derandomize \( \text{BPP} \).

In the present paper we extend the just mentioned result of Nisan and Wigderson to the nondeterministic setting. In section 3 we show that their generator can also be used to derandomize the Arthur-Merlin class \( \text{AM} = \text{BP} \cdot \text{NP} \), provided it is built from a set in \( \text{E} \) that is hard to approximate by nondeterministic circuits of size \( 2^{\alpha n} \) for some \( \alpha > 0 \).

In section 4 we apply this extension to answer some questions left open by Lutz in [Lut96]. We show that for all \( k \geq 2 \), \( \mu_p(\Delta^P_k) \neq 0 \) implies \( \text{BP} \cdot \Sigma^P_k = \Sigma^P_k \) (see Fig. 2). Furthermore, we show under the possibly weaker assumption \( \mu_p(\text{NP}) \neq 0 \) that \( \text{BP} \cdot \text{NP} \) can be derandomized by using a logarithmic number of advice bits (i.e., \( \text{BP} \cdot \text{NP} \subseteq \text{NP/log} \)). Under the stronger hypothesis \( \mu_p(\text{NP} \cap \text{coNP}) \neq 0 \) we also prove that \( \text{BP} \cdot \text{NP} = \text{NP} \) which has some immediate strong implications as, for example, Graph Isomorphism is in \( \text{NP} \cap \text{coNP} \).

Relatively, in section 5 we show that for all \( k \geq 2 \), \( \mu_p(\Theta^P_k) \neq 0 \) implies \( \text{BP} \cdot \Theta^P_k = \Theta^P_k \), answering an open problem stated in [Lut96]. Thus, \( \mu_p(\Theta^P_2) \neq 0 \) has the remarkable consequence that \( \text{AM} \cap \text{coAM} \) (and hence Graph Isomorphism) is low for \( \Theta^P_2 \).

Finally, we show in section 6 that the Arthur-Merlin class \( \text{MA} \) is contained in \( \text{ZPP}^{\text{NP}} \) and that \( \text{MA} \cap \text{coMA} \) is even low for \( \text{ZPP}^{\text{NP}} \). These results follow easily by using the Nisan-Wigderson generator [NW94].
2 Preliminaries

In this section we give formal definitions and describe the results of Nisan and Wigderson [NW94] and of Lutz [Lut96] which we generalize in this paper.

Let the finite alphabet be fixed as $\Sigma = \{0,1\}$. We denote the cardinality of a finite set $X$ by $|X|$ and the length of $x \in \Sigma^*$ by $|x|$. The join of two sets $A$ and $B$, denoted by $A \oplus B$, is defined as $A \oplus B = \{0x \mid x \in A\} \cup \{1x \mid x \in B\}$. The characteristic function of a language $L \subseteq \Sigma^*$ is defined as $L(x) = 1$ if $x \in L$, and $L(x) = 0$ otherwise (by abuse of notation we denote this function also by $L(x)$). The restriction of $L(x)$ to strings of length $n$ can be considered as an $n$-ary boolean function that we denote by $L^n$.

The class $E$ is defined as $\bigcup_{\geq 0} \text{DTIME}(2^n)$, and $\text{EXP} = \bigcup_{\geq 0} \text{DTIME}(2^n)$. For a class $\mathcal{C}$ of sets and a class $\mathcal{F}$ of functions from $\Sigma^*$ to $\Sigma^*$, let $\mathcal{C}/\mathcal{F}$ [KL80] be the class of sets $A$ such that there is a set $B \in \mathcal{C}$ and a function $h \in \mathcal{F}$ such that for all $x \in \Sigma^*$,

$$x \in A \Leftrightarrow (x, h(|x|)) \in B.$$ 

The function $h$ is called an advice function for $A$.

The BP-operator [Sch89] assigns to each complexity class $\mathcal{C}$ a randomized version $\text{BP} \cdot \mathcal{C}$ as follows. A set $L$ belongs to $\text{BP} \cdot \mathcal{C}$ if there exist a polynomial $p$ and a set $D \in \mathcal{C}$ such
that for all \( x \), \( |x| = n \)

\[
x \in L \Rightarrow \text{Prob}_{r \in_R \{0,1\}^{p(n)}}[\langle x, r \rangle \in D] \geq 3/4,
\]

\[
x \notin L \Rightarrow \text{Prob}_{r \in_R \{0,1\}^{p(n)}}[\langle x, r \rangle \in D] \leq 1/4.
\]

Here, the subscript \( r \in_R \{0,1\}^{p(n)} \) indicates that \( r \) is chosen uniformly at random from \( \{0,1\}^{p(n)} \), and \( \langle \cdot, \cdot \rangle \) is a standard pairing function.

The definitions of complexity classes we consider like \( P \), \( NP \), \( AM \) etc. can be found in standard books [BDG95, BDG90, Pap94]. By \( \log \) we denote the function \( \log x = \max\{1, \lceil \log_2 x \rceil \} \).

We next define boolean functions that are hard-to-approximate and related notions. For a function \( s : \mathcal{N} \to \mathcal{N}^+ \) and an oracle set \( A \subseteq \Sigma^* \), \( \text{CIR}^A(n, s) \) denotes the class of boolean functions \( f : \{0,1\}^n \to \{0,1\} \) that can be computed by some oracle circuit \( c \) of size at most \( s(n) \) having access to \( A \). In case \( A = \emptyset \) we denote this class by \( \text{CIR}(n, s) \).

Furthermore, let \( \text{CIR}(s) = \bigcup_{n \geq 0} \text{CIR}(n, s) \) and \( \text{CIR}^A(s) = \bigcup_{n \geq 0} \text{CIR}^A(n, s) \).

**Definition 1** (cf. [Yao82, NW94])

1. Let \( f : \{0,1\}^n \to \{0,1\} \) be a boolean function, \( C \) be a set of boolean functions, and let \( r \in \mathcal{R}^+ \) be a positive real number. \( f \) is said to be \( r \)-hard for \( C \) if for all \( n \)-ary boolean functions \( g \) in \( C \),

\[
2^n(1 - 1/r) < \| \{ x \in \{0,1\}^n \mid f(x) = g(x) \} \| < 2^n(1 + 1/r).
\]

2. Let \( r : \mathcal{N} \to \mathcal{R}^+ \) be a function. A language \( L \subseteq \Sigma^* \) is said to be \( r \)-hard for \( C \) if for all but finitely many \( n \), \( L^{=n} \), considered as an \( n \)-ary boolean function, is \( r(n) \)-hard for \( C \).

3. A class \( D \) of languages is called \( r \)-hard for \( C \) if some language \( L \in D \) is \( r \)-hard for \( C \).

4. A boolean function \( f \) (a language \( L \), or a language class \( D \)) is called \( \text{CIR}^A(r) \)-hard if \( f \) (resp. \( L, D \)) is \( r \)-hard for \( \text{CIR}^A(r) \).

The already discussed result of Nisan and Wigderson can be stated in relativized form as follows.

**Theorem 2** [NW94] For all \( \alpha > 0 \) and all oracles \( A \), if \( E^A \) is \( \text{CIR}^A(2^{\alpha n}) \)-hard, then \( \text{P}^A = \text{BPP}^A \).

Resource-bounded measure was introduced in [Lut92]. We briefly recall some basic definitions from [Lut92, Lut96] leading to the definition of a language class having p-measure 0. Intuitively, if a class \( C \) of languages has p-measure 0, then \( C \cap E \) forms a negligibly small subclass of \( E \) (see [Lut92, Lut96] for more motivation on this concept).
Definition 3 [Lut92, Lut96]

1. A function \( d : \Sigma^* \rightarrow \mathcal{R}^+ \) is called a supermartingale if for all \( w \in \Sigma^* \),
   \[
d(w) \geq \frac{d(w0) + d(w1)}{2}.
   \]

2. The success set of a supermartingale \( d \) is defined as
   \[
   S^\infty[d] = \{ A \mid \limsup_{l \rightarrow \infty} d(A(s_1) \cdots A(s_l)) = \infty \}
   \]
   where \( s_1 = \lambda, s_2 = 0, s_3 = 1, s_4 = 00, s_5 = 01, \ldots \) is the standard enumeration of \( \Sigma^* \)
   in lexicographic order. The unitary success set of \( d \) is
   \[
   S^1[d] = \bigcup_{d(w) \geq 1} C_w
   \]
   where, for \( w \in \Sigma^* \), \( C_w \) is the class of languages \( A \) such that \( A(s_1) \cdots A(s_{|w|}) = w \).

3. A function \( d : \mathcal{N}^i \times \Sigma^* \rightarrow \mathcal{R} \) is said to be \( p \)-computable if there is a function
   \( f : \mathcal{N}^{i+1} \times \Sigma^* \rightarrow \mathcal{R} \) such that \( f(r, k_1, \ldots, k_i, w) \) is computable in time polynomial in
   \( r + k_1 + \cdots + k_i + |w| \) and \( |f(r, k_1, \ldots, k_i, w) - d(k_1, \ldots, k_i, w)| \leq 2^{-r} \).

4. A class \( X \) of languages has \( p \)-measure 0 (in symbols, \( \mu_p(X) = 0 \)) if there is a \( p \-
   \)computable supermartingale \( d \) such that \( X \subseteq S^\infty[d] \).

In the context of resource-bounded measure, it is interesting to ask for the measure of the class of all sets \( A \) for which \( E^A \) is not \( \text{CIR}^A(2^{|n|}) \)-hard. Building on initial results in [Lut93] it is shown in [AS94] that this class has \( p \)-measure 0.

Lemma 4 [AS94] For all \( 0 < \alpha < 1/3 \), \( \mu_p\{ A \mid E^A \text{ is not } \text{CIR}^A(2^{|n|}) \text{-hard} \} = 0 \).

Lutz strengthened this to the following result that is more useful for many applications.

Lemma 5 [Lut96] For all \( 0 < \alpha < 1/3 \) and all oracles \( B \in E \),
   \[
   \mu_p\{ A \mid E^A \text{ is not } \text{CIR}^{AB}(2^{|n|}) \text{-hard} \} = 0.
   \]

As a consequence of the above lemma, Lutz derives the following theorem.

Theorem 6 [Lut96] For \( k \geq 2 \), if \( \mu_p(\Delta^p_k) \neq 0 \) then \( \text{BP} \cdot \Delta^p_k \subseteq \Delta^p_k \).

It is not hard to see that Theorem 6 can be extended to any complexity class \( C \subseteq \text{EXP} \)
closed under join and polynomial-time Turing reducibility (see also Corollary 22). For example, if \( \oplus P \) does not have \( p \)-measure 0, then \( \text{BP} \cdot \oplus P \subseteq \oplus P \), implying [Tod91] that the polynomial hierarchy is contained in \( \oplus P \).

In sections 4 and 5 we address the questions left open by Lutz in [Lut96], namely whether \( \text{BP} \cdot \Sigma^p_k = \Sigma^p_k \) (or \( \text{BP} \cdot \Theta^p_k = \Theta^p_k \)) can be also derived from \( \mu_p(\Delta^p_k) \neq 0 \), and whether stronger consequences can be derived from \( \mu_p(\text{NP}) \neq 0 \) and \( \mu_p(\text{NP} \cap \text{coNP}) \neq 0 \). The general steps of our proofs follow a pattern that is similar to the proofs in [Lut96].
3 Derandomizing AM in relativized worlds

In this section we show that the Nisan-Wigderson generator can also be used to derandomize the Arthur-Merlin class AM = BP-NP [Bab85]. We first define the counterpart of Definition 1 for nondeterministic circuits and the corresponding notion of hard-to-approximate boolean functions. A nondeterministic circuit $c$ has two kinds of input gates: in addition to the actual inputs $x_1, \ldots, x_n$, $c$ has a series of distinguished guess inputs $y_1, \ldots, y_m$. The value computed by $c$ on input $x \in \Sigma^n$ is 1 if there exists a $y \in \Sigma^m$ such that $c(xy) = 1$, and 0 otherwise [SV85].

We now define hardness for nondeterministic circuits. $\text{NCIR}^A(n, s)$ denotes the union $\bigcup_{s \geq 0} \text{NCIR}^A(n, s)$, where $\text{NCIR}^A(n, s)$ contains all boolean functions $f : \{0, 1\}^n \to \{0, 1\}$ that can be computed by some nondeterministic oracle circuit $c$ of size at most $s(n)$, having access to oracle $A$.

**Definition 7** A boolean function $f$ (a language $L$, or a language class $\mathcal{D}$) is called $\text{NCIR}^A(r)$-hard if $f$ (resp. $L$, $\mathcal{D}$) is $r$-hard for $\text{NCIR}^A(r)$.

We continue by recalling some notation from [NW94]. Let $p, l, m, k$ be positive integers. A collection $D = \langle D_1, \ldots, D_p \rangle$ of sets $D_i \subseteq \{1, \ldots, l\}$ is called a $(p, l, m, k)$-design if

- for all $i = 1, \ldots, p$, $|D_i| = m$ and
- for all $i \neq j$, $|D_i \cap D_j| \leq k$.

Using $D$ we get from a boolean function $g : \{0, 1\}^m \to \{0, 1\}$ a sequence of boolean functions $g_i : \{0, 1\}^l \to \{0, 1\}$, $i = 1, \ldots, p$, defined as

$$g_i(s_1, \ldots, s_l) = g(s_{i_1}, \ldots, s_{i_m}) \text{ where } D_i = \{i_1, \ldots, i_m\}.$$ 

By concatenating these function values we get a function $g_D : \{0, 1\}^l \to \{0, 1\}^p$ where $g_D(s) = g_1(s) \ldots g_p(s)$. Now we can state the following lemma.

**Lemma 8** Let $D$ be a $(p, l, m, k)$-design and let $g : \{0, 1\}^m \to \{0, 1\}$ be an $\text{NCIR}^A(p^2 + p2^k)$-hard function. Then the function $g_D$ has the property that for every $p$-input nondeterministic oracle circuit $c$ of size at most $p^2$,

$$\left| \text{Prob}_{y \in \{0, 1\}^p} [c^A(y) = 1] - \text{Prob}_{s \in \{0, 1\}^l} [c^A(g_D(s)) = 1] \right| \leq 1/p.$$ 

**Proof.** The proof follows along similar lines as the one of [NW94, Lemma 2.4]. We show that if there is a nondeterministic oracle circuit $c$ of size at most $p^2$ such that

$$\left| \text{Prob}_{y \in \{0, 1\}^p} [c^A(y) = 1] - \text{Prob}_{s \in \{0, 1\}^l} [c^A(g_D(s)) = 1] \right| > 1/p,$$ 

then
then $g$ is not NCIR$^A(p^2 + p^{2k})$-hard. Let $S_1, \ldots, S_t$ and $Z_1, \ldots, Z_p$ be independently and uniformly distributed random variables over $\{0, 1\}$ and let $S = (S_1, \ldots, S_t)$. Then we can restate the inequality above as follows:

$$\left| \text{Prob}[c^A(Z_1, \ldots, Z_p) = 1] - \text{Prob}[c^A(g_1(S), \ldots, g_p(S)) = 1] \right| > 1/p$$

where $g_i(s)$ denotes the $i$th bit of $g_D(s)$, $i = 1, \ldots, p$. Now consider the random variables

$$X_i = c^A(g_1(S), \ldots, g_{i-1}(S), Z_i, \ldots, Z_p), \quad i = 1, \ldots, p.$$

Since $X_1 = c^A(Z_1, \ldots, Z_p)$ and since $X_{p+1} = c^A(g_1(S), \ldots, g_p(S))$, we can fix an index $j \in \{1, \ldots, p\}$ such that

$$\left| \text{Prob}[X_j = 1] - \text{Prob}[X_{j+1} = 1] \right| > 1/p^2. \quad (1)$$

Consider the boolean function $h : \{0, 1\}^t \times \{0, 1\}^{p-j+1} \to \{0, 1\}$ defined as

$$h(s, z_j, \ldots, z_p) = \begin{cases} 
  z_j, & \text{if } c^A(g_1(s), \ldots, g_{j-1}(s), z_j, \ldots, z_p) = 1, \\
  1 - z_j, & \text{otherwise}.
\end{cases}$$

It is not hard to see that $(1)$ is equivalent to

$$\left| \text{Prob}[h(S, Z_j, \ldots, Z_p) = g_j(S)] - 1/2 \right| \geq 1/p^2. \quad (2)$$

Since $g_j(s_1, \ldots, s_t)$ only depends on the bits $s_i$ with $i \in D_j$, we can apply an averaging argument to find fixed bits $\hat{s}_i; i \not\in D_j$ and fixed bits $\hat{z}_j, \ldots, \hat{z}_p$ such that $(1)$ still holds under the condition that $S_i = \hat{s}_i$ for all $i \not\in D_j$ and $Z_i = \hat{z}_i$ for all $i = j, \ldots, p$. Since $g_j(s_1, \ldots, s_t) = g(s_1, \ldots, s_m)$ (for notational convenience we assume w.l.o.g. that $D_j = \{1, \ldots, m\}$) we thus get

$$\left| \text{Prob}[h(S_1, \ldots, S_m, \hat{s}_{m+1}, \ldots, \hat{s}_i, \hat{z}_j, \ldots, \hat{z}_p) = g(S_1, \ldots, S_m)] - 1/2 \right| \geq 1/p^2.$$

Now consider the nondeterministic oracle circuit $c'$ with inputs $s_1, \ldots, s_m$ that first computes $g_1(s_1, \ldots, s_m, \hat{s}_{m+1}, \ldots, \hat{s}_i), \ldots, g_{j-1}(s_1, \ldots, s_m, \hat{s}_{m+1}, \ldots, \hat{s}_i)$ and then simulates $c^A$ to compute $c^A(g_1(s_1, \ldots, s_m, \hat{s}_{m+1}, \ldots, \hat{s}_i), \ldots, g_{j-1}(s_1, \ldots, s_m, \hat{s}_{m+1}, \ldots, \hat{s}_i), \hat{z}_j, \ldots, \hat{z}_p)$. Then either $c'^A(s_1, \ldots, s_m)$ computes the function $h(s_1, \ldots, s_m, \hat{s}_{m+1}, \ldots, \hat{s}_i, \hat{z}_j, \ldots, \hat{z}_p)$ or the negation of this function (depending on whether $\hat{z}_j = 1$ or $\hat{z}_j = 0$) and hence it follows that

$$\left| \text{Prob}[c'^A(S_1, \ldots, S_m) = g(S_1, \ldots, S_m)] - 1/2 \right| \geq 1/p^2.$$

Crucially, observe that each of $g_1(s_1, \ldots, s_m, \hat{s}_{m+1}, \ldots, \hat{s}_i), \ldots, g_{j-1}(s_1, \ldots, s_m, \hat{s}_{m+1}, \ldots, \hat{s}_i)$ only depends on at most $k$ input bits. Hence, these values can be computed by a deterministic subcircuit of size at most $2^k$ (namely, the brute-force circuit that evaluates that particular $k$-ary boolean function). This means that the size of $c'$ is at most $p^2 + p^{2k}$, implying that $g$ is not NCIR$^A(p^2 + p^{2k})$-hard.

For our extension of Theorem 2 we also need the following lemma.
Lemma 9 [NW94] Let \( c \) be a positive integer and let the integer valued functions \( l, m, k \) defined as \( l(p) = 2c^3 \log p, m(p) = c \log p, \) and \( k(p) = \log p \). Then there is a polynomial-time algorithm that on input \( 1^p \) computes a \( (p, l(p), m(p), k(p)) \)-design.

Theorem 10 Let \( A \) and \( B \) be oracles and let \( \alpha > 0 \). If \( E^A \) is NCIR\( B(2^{\alpha n}) \)-hard, then \( BP \cdot NP^B \subseteq NP^B / FP^A \). In particular, if \( E^A \) is NCIR\( A(2^{\alpha n}) \)-hard, then \( BP \cdot NP^A = NP^A \).

Proof. Let \( L \in BP \cdot NP^B \). Then there exist a polynomial \( p \) and a set \( D \in NP^B \) such that for all \( x, |x| = n \)

\[
\begin{align*}
x \in L & \Rightarrow \text{Prob}_{r \in \{0,1\}^{p(n)}}[(x, r) \in D] \geq 3/4, \\
x \notin L & \Rightarrow \text{Prob}_{r \in \{0,1\}^{p(n)}}[(x, r) \in D] \leq 1/4.
\end{align*}
\]

For a fixed input \( x \), the decision procedure for \( D \) on input \( x, r \) can be simulated by some nondeterministic oracle circuit \( c_x \) with inputs \( r_1, \ldots, r_{p(n)} \), implying that

\[
\begin{align*}
x \in L & \Rightarrow \text{Prob}_{r \in \{0,1\}^{p(n)}}[c_x^B(r) = 1] \geq 3/4, \\
x \notin L & \Rightarrow \text{Prob}_{r \in \{0,1\}^{p(n)}}[c_x^B(r) = 1] \leq 1/4
\end{align*}
\]

where w.l.o.g. we can assume that the size of \( c_x \) is bounded by \( p^3(|x|) \).

Let \( \alpha > 0 \) and let \( C \in E^A \) be an NCIR\( B(2^{\alpha n}) \)-hard language. Then for almost all \( n \), the boolean function \( C^{m(n)} : \{0,1\}^n \to \{0,1\} \) is NCIR\( B(2^{\alpha n}) \)-hard. Thus, letting \( c = \lceil \alpha^{-1} \rceil \) and \( m(n) = 3c \log p(n) \), it follows that for almost all \( n \), \( C^{m(n)} \) is NCIR\( B(p(n)^3) \)-hard.

Now let \( l(n) = 18c^2 \log p(n) \) and \( k(n) = \log p(n) \). Then we can apply Lemmas 9 and 8 to get for almost all \( n \) a \( (p(n), l(n), m(n), k(n)) \)-design \( D \) such that the function \( C^{m(n)}_D : \{0,1\}^{l(n)} \to \{0,1\}^{p(n)} \) has for every \( p(n) \)-input nondeterministic oracle circuit \( c \) of size at most \( p(n)^2 \) the property that

\[
\left| \text{Prob}_{y \in \{0,1\}^{l(n)}}[c^B(y) = 1] - \text{Prob}_{s \in \{0,1\}^{l(n)}}[c^B(C^{m(n)}_D(s)) = 1] \right| \leq 1/p(n).
\]

Notice that since \( m(n) = O(\log n) \) and since \( C \in E^A \), it is possible to compute the advice function \( h(1^n) = C(0^{m(n)}) \cdots C(1^{m(n)}) \) in \( FP^A \). Hence, the following procedure witnesses \( B \in NP^B / FP^A \):

\textbf{input} \( x, |x| = n \), and the sequence \( h(1^n) = C(0^{m(n)}) \cdots C(1^{m(n)}) \); 
compute a \( (p(n), l(n), m(n), k(n)) \)-design \( D \) and let \( r_1, \ldots, r_{2^{l(n)}} \) be the pseudorandom strings produced by \( C^{m(n)}_D \) on all seeds from \( \{0,1\}^{l(n)} \); 
if the number of \( r_i \) for which \( c^B_x(r_i) = 1 \) is at least \( 2^{l(n)-1} \) then accept else reject.
4 Derandomizing $BP \cdot \Sigma^P_k$ if $\Delta^P_k$ is not small

In this section we apply the relativized derandomization of the previous section to extend Lutz’s Theorem 6 to the $\Sigma^P_k$ levels of the polynomial hierarchy. A crucial result used in the proof of Lutz’s Lemma 5 is the fact that there are many $n$-ary boolean functions that are CIR($2^{an}$)-hard.

**Lemma 11** [Lut93] For each $\alpha$ such that $0 < \alpha < 1/3$, there is a constant $m_0$ such that for all $m \geq m_0$ the number of boolean functions $f : \{0,1\}^m \rightarrow \{0,1\}$ that are not CIR($2^{an}$)-hard is at most $2^{2m} \cdot e^{-2m/4}$.

In the next lemma we establish the same bound on the number of $m$-ary boolean functions that are not NCIR($2^{am}$)-hard.

**Lemma 12** For each $\alpha$ such that $0 < \alpha < 1/3$, there is a constant $m_0$ such that for all $m \geq m_0$ the number of $m$-ary boolean functions that are not NCIR($2^{am}$)-hard is at most $2^{2m} \cdot e^{-2m/4}$.

**Proof.** The proof follows an essentially similar counting argument as in the deterministic case (see [Lut93]). We sketch the counting argument below to point out how nondeterministic circuits are also handled in the same way.

In the sequel, let $q = 2^{am}$ and let NCIR$_j(m,q)$ denote the class of $m$-ary boolean functions computed by nondeterministic circuits of size $q$ with exactly $j$ guess inputs. Notice that NCIR$(m,q) = \bigcup_{j=0}^m$ NCIR$_j(m,q)$, implying that $\|\text{NCIR}(m,q)\| \leq \sum_{j=0}^m \|\text{NCIR}_j(m,q)\|$. It is shown in [Sch86] by a standard counting argument that $\|\text{CIR}(m,q)\| \leq q[(16e(m+q)^2)/q]^q$.

Since each function in NCIR$_j(m,q)$ is uniquely determined by a deterministic circuit of size $q$ with $m+j$ inputs, it follows that $\|\text{NCIR}_j(m,q)\| \leq \|\text{CIR}(m+j,q)\| \leq q[(16e(m+j+q)^2)/q]^q \leq q[(16e(3q)^2)/q]^q = q(144eq)^q$. Thus it follows that $\|\text{NCIR}(m,q)\| \leq \sum_{j=0}^m q(144eq)^q = q(q+1)(144eq)^q$.

We now place a bound on the number of $m$-ary boolean functions that are not NCIR$(q)$-hard. Let NCIR$(m,q) = \{ g \mid g \in \text{NCIR}(m,q) \}$. Furthermore, as defined in [Lut96], let $\text{Delta}(m,q) = \{ D \subseteq \Sigma^m \mid ||D|| \leq 2^{m-1}(1 - 1/q) \}$. Now, by applying standard Chernoff bounds, as shown in [Lut93], it can be seen that $||\text{Delta}(m,q)|| \leq 2^{2m}2^{-c(1-2\alpha)m}$, where $c > 0$ is a small constant.

From the notion of NCIR$(q)$-hard functions (Definition 7) it is easy to see that there are at most $||\text{NCIR}(m,q) \cup \text{NCIR}(m,q)|| \cdot ||\text{Delta}(m,q)|| \leq 2\alpha q(144eq)^q \cdot 2^{2m}2^{-c(1-2\alpha)m}$. 

9
distinct m-ary boolean functions that are not NCIR(q)-hard. Hence, using the fact that 0 < α < 1/3 we can easily find a constant m₀ such that for m ≥ m₀ the above number is bounded above by 2²ᵐᵉ⁻ᵐ/₄ as required. ■

We further need the important Borel-Cantelli-Lutz Lemma [Lut92]. A series ∑ₖ₌₀∞ aₖ of nonnegative reals is said to be p-convergent if there is a polynomial q such that for all r ∈ N, ∑ₖ₌ₐ(r) aₖ ≤ 2⁻ʳ.

**Theorem 13** [Lut92] Assume that d : N × Σ⁺ → R⁺ is a function with the following properties

1. d is p-computable.
2. For each k ∈ N, the function dₖ, defined by dₖ(w) = d(k, w) is a supermartingale.
3. The series ∑ₖ₌₀∞ dₖ(λ) is p-convergent.

Then μₚ(∩ₖ₌₁∞ Uₖ₌₁∞ S₁[dₖ]) = 0.

Now we are ready to extend Lutz’s Lemma 5 to the case of nondeterministic circuits.

**Lemma 14** For all 0 < α < 1/3 and all oracles B ∈ E,

μₚ{A | E^A is not NCIR^{A\#B}(2^αm)-hard} = 0.

**Proof.** Let 0 < α < 1/3 and B ∈ E. For each language A define the test language

C(A) = {x | x1₀²|ₓ| ∈ A},

and let X = {A | C(A) is not NCIR^{A\#B}(2^αm)-hard}. Notice that since C(A) ∈ E^A, the theorem follows from the following claim.

**Claim.** μₚ(X) = 0.

**Proof of Claim.** The proof follows exactly the same lines as in [Lut96, Theorem 3.2] except for minor changes to take care of the fact that we are dealing with nondeterministic circuits. Let q = 2^αm and recall from the proof of Lemma 12 that for all m > m₀,

||NCIR(m, q)|| · ||Δ(m, q)|| ≤ 2²ᵐₑ⁻²ᵐ/₄.

Let 2ᵐ = k and 2ᵐ₀ = k₀. For each nondeterministic m-ary circuit γ of size q and each D ∈ Δ(m, q), define the class

Y_{γ, D} = \{A | L(γ^{A\#B})ΔD = C(A)^≡m\}.

¹This test language was originally defined by [AS94] and later used in [Lut96].
For each \( k > 0 \), let
\[
X_k = \begin{cases} 
\bigcup_{\gamma,D} \mathcal{Y}_{\gamma,D}, & \text{if } k = 2^m \text{ for some } m, \\
\emptyset, & \text{otherwise}
\end{cases}
\]
where the union is taken over all nondeterministic circuits \( \gamma \) of size \( q \) and \( D \in \text{Delta}(m, q) \). It follows immediately that
\[
\mathcal{X} = \bigcap_{j \geq 0} \bigcup_{k \geq j} X_k.
\]
We will show that \( \mu_p(\mathcal{X}) = 0 \) by applying the Borel-Cantelli-Lutz Lemma (Theorem 13). In order to apply it we define (exactly as in [Lut96]) \( d : \mathcal{N} \times \Sigma^* \to \mathcal{R}^+ \) as follows:

1. If \( k < k_0 \) or \( k \) is not a power of 2 then \( d_k(w) = 0 \).
2. If \( k = 2^m \geq k_0 \) and \( |w| < 2^{k+1} \) then \( d_k(w) = e^{-k^{1/4}} \).
3. If \( k = 2^n > k_0 \) and \( |w| \geq 2^{k+1} \) then \( d_k(w) = \Sigma_{\gamma,D} \text{Pr}[\mathcal{Y}_{\gamma,D}[C_w]] \).

Now, it can be proved exactly as in [Lut96] that:

1. \( d \) is \( p \)-computable;
2. For each \( k > 0 \), \( d_k \) is a supermartingale with \( d_k(\lambda) \leq e^{-k^{1/4}} \);
3. For all \( k \geq k_0 \), \( X_k \subseteq S^1[d_k] \);
4. \( \mathcal{X} \subseteq \bigcup_{j \geq 0} \bigcap_{k \geq j} S^1[d_k] \).

The only point where a different argument is required is in showing that \( d \) is \( p \)-computable because the circuits \( \gamma \) used to define \( \mathcal{Y}_{\gamma,D} \) are nondeterministic. Nevertheless, notice that the only nontrivial case to be handled in the definition of \( d_k \) is when \( k = 2^m > k_0 \) and \( |w| \geq 2^{k+1} \). In this case, the size of the considered circuit \( \gamma \) is bounded by \( 2^{\alpha m} \leq k \). Therefore, in time polynomial in \( |w| \) the nondeterministic circuit \( \gamma \) for each length \( m \) input can be evaluated by exhaustive search.

It is now easy to derandomize \( \text{BP} \cdot \Sigma_k^p \) under the assumption that \( \Delta_k^p \) has non-zero \( p \)-measure.

**Theorem 15** For all \( k \geq 2 \), if \( \mu_p(\Delta_k^p) \neq 0 \), then \( \text{BP} \cdot \Sigma_k^p = \Sigma_k^p \).

**Proof.** Assume the hypothesis and let \( B \) be a fixed \( \Sigma_k^{p-1} \)-complete set. We know from Lemma 14 that for \( \alpha = 1/4 \), \( \mu_p(\{A \mid E^A \text{ is not NCIR}^{A\oplus B}(2^{2n})\text{-hard}\}) = 0 \). On the other hand, \( \mu_p(\Delta_k^p) \neq 0 \). Hence, there is a set \( A \in \Delta_k^p \) such that \( E^A \) (and thus also \( E^{A\oplus B} \)) is NCIR\(^{A\oplus B}(2^{2n})\text{-hard} \). Applying Theorem 10 we get
\[
\Sigma_k^p = \text{NP}^{A\oplus B} = \text{BP} \cdot \text{NP}^{A\oplus B} = \text{BP} \cdot \Sigma_k^p,
\]
which completes the proof.

Furthermore, we obtain the following two interesting consequences.

**Corollary 16** If \( \mu_p(NP \cap coNP) \neq 0 \), then \( BP \cdot NP = NP \).

**Proof.** Assuming that \( \mu_p(NP \cap coNP) \neq 0 \), similar to the proof of Theorem 15 it follows that there is a set \( A \in NP \cap coNP \) such that \( NP^A = BP \cdot NP^A \). From the fact that \( NP^{NP \cap coNP} = NP \), we immediately get that \( NP = BP \cdot NP \). ◼

**Corollary 17** If \( \mu_p(NP) \neq 0 \), then \( BP \cdot NP \subseteq NP/\log \).

**Proof.** If \( \mu_p(NP) \neq 0 \), then from Theorems 10 and 14 it follows that there is a set \( A \in NP \) such that \( BP \cdot NP \subseteq NP/FP^A \). Actually, from the proof of Lemma 14 we know something stronger. Namely, we know that the test language

\[
C(A) = \{ x \mid x10^{2|I|} \in A \}
\]

is in \( E^A \) and is \( NCIR(2^n) \)-hard. Hence, we can assume that \( A \) is sparse and therefore we get \( BP \cdot NP \subseteq NP/\log \), by using a census argument [Kad89] (see also [KT94]). ◼

## 5 Derandomizing \( BP \cdot \Theta_k^P \) if \( \Theta_k^P \) is not small

In [Lut96] it was an open question whether \( BP \cdot \Theta_2^P = \Theta_2^P \) can be proven as a consequence of \( \mu_p(NP) \neq 0 \). We answer this question by proving the same consequence from a possibly weaker assumption. For a complexity class \( K \in \{ P, BPP, E \} \) and oracle \( A \), let \( K^A \) denote the respective relativized class where the machine for \( K \) makes only parallel queries to \( A \).

**Definition 18** Let \( A \subseteq \Sigma^* \) be an oracle set. Let \( \text{CIR}^A(n, s) \) denote the class of boolean functions \( f : \{0,1\}^n \rightarrow \{0,1\} \) that can be computed by some oracle circuit \( c \) of size at most \( s(n) \) that makes only parallel queries to oracle \( A \). Furthermore, let \( \text{CIR}^A(s) = \bigcup_{n \geq 0} \text{CIR}^A(n, s) \).

It is not hard to verify that Nisan and Wigderson's result (Theorem 2) also holds in the parallel setting.

**Theorem 19** For all \( \alpha > 0 \) and all oracles \( A \), if \( E^A \) is \( \text{CIR}^A(2^n) \)-hard, then \( P^A = \text{BPP}^A \).

**Corollary 20** For all \( k \geq 2 \), if \( \mu_p(\Theta_k^P) \neq 0 \), then \( BP \cdot \Theta_k^P = \Theta_k^P \).
Proof. Assume the hypothesis and let $B$ be a fixed $\Sigma^p_{k-1}$-complete set. Observe that if $\mu_p(\Theta^p_k) \neq 0$, it follows from the proof of Lemma 5 (as given in [Lut96]) that for $\alpha = 1/4$ there is a set $A \in \Theta^p_k$ such that $C(A)$ is $\text{CIR}^{A \oplus B}(2^{2n})$-hard. Since $C(A) \in E^A_{||} \subseteq E^{A \oplus B}_{||}$ and since $\text{CIR}^{A \oplus B}_{||}(2^{2n}) \subseteq \text{CIR}^{A \oplus B}(2^{2n})$, it follows that $E^A_{||}$ is $\text{CIR}^{A \oplus B}_{||}(2^{2n})$-hard, implying that

$$\Theta^p_k = P^{A \oplus B}_{||} = \text{BPP}^{A \oplus B}_{||} = \text{BP} \cdot \Theta^p_k,$$

where the second equality follows by Theorem 19.

Corollary 20 has the following immediate lowness consequence.

Corollary 21 If $\mu_p(\Theta^p_k) \neq 0$ then $\text{AM} \cap \text{coAM}$ (and hence the graph isomorphism problem) is low for $\Theta^p_k$.

Corollary 20 can easily be extended to further complexity classes.

Corollary 22 For any complexity class $C \subseteq \text{EXP}$ closed under join and polynomial-time truth-table reducibility, $\mu_p(C) \neq 0$ implies that $\text{BP} \cdot C \subseteq C$.

Proof. Assume the hypothesis and let $L$ be a set in $\text{BP} \cdot C$, witnessed by some set $B \in C$. Since $C$ is closed under many-one reducibility we can define a suitably padded version $\hat{B}$ of $B$ in $C \cap E$ such that $L$ belongs to $\text{BP} \cdot \{\hat{B}\}$. Now, exactly as in the proof of Corollary 20 we can argue that there is a set $A \in C$ with the property that $E^{A \oplus \hat{B}}_{||}$ is $\text{CIR}^{A \oplus \hat{B}}_{||}(2^{2n})$-hard. Hence, by Theorem 19 it follows that

$$L \in \text{BP} \cdot \{\hat{B}\} \subseteq \text{BPP}^{A \oplus \hat{B}}_{||} = P^{A \oplus \hat{B}}_{||} \subseteq C.$$

For example, using the fact that $\text{PP}$ is closed under polynomial-time truth-table reducibility [FR96], it follows that if $\mu_p(\text{PP}) \neq 0$, then $\text{BP} \cdot \text{PP} = \text{PP}$.

6 MA is contained in $\text{ZPP}^{\text{NP}}$

In this section we apply the Nisan-Wigderson generator to show that $\text{MA}$ is contained in $\text{ZPP}^{\text{NP}}$ and, as a consequence, that $\text{MA} \cap \text{coMA}$ is low for $\text{ZPP}^{\text{NP}}$. This improves on a result of [ZF87] where a quantifier simulation technique is used to show that $\text{NP}^{\text{BPP}}$ (a subclass of $\text{MA}$) is contained in $\text{ZPP}^{\text{NP}}$. The proof of the next theorem also makes use of the fact that there are many $n$-ary boolean functions that are $\text{CIR}(2^{2n})$-hard (Lemma 11).

Theorem 23 $\text{MA}$ is contained in $\text{ZPP}^{\text{NP}}$. 

13
Proof. Let \( L \) be a set in MA. Then there exist a polynomial \( p \) and a set \( B \in \mathcal{P} \) such that for all \( x \), \( |x| = n \),

\[
\begin{align*}
x \in A & \Rightarrow \exists y, |y| = p(n) : \text{Prob}_{r \in \{0,1\}^{p(n)}}[(x, y, r) \in B] \geq 3/4, \\
x \notin A & \Rightarrow \forall y, |y| = p(n) : \text{Prob}_{r \in \{0,1\}^{p(n)}}[(x, y, r) \in B] \leq 1/4.
\end{align*}
\]

For fixed strings \( x, y \), the decision procedure for \( B \) on input \( x, y, r \) can be simulated by some circuit \( c_{x,y} \) with inputs \( r_1, \ldots, r_{p(n)} \), implying that

\[
\begin{align*}
x \in A & \Rightarrow \exists y, |y| = p(n) : \text{Prob}_{r \in \{0,1\}^{p(n)}}[c_{x,y}(r) = 1] \geq 3/4, \\
x \notin A & \Rightarrow \forall y, |y| = p(n) : \text{Prob}_{r \in \{0,1\}^{p(n)}}[c_{x,y}(r) = 1] \leq 1/4
\end{align*}
\]

where w.l.o.g. we can assume that the size of \( c_{x,y} \) is bounded by \( p^2(|x|) \). It follows by the deterministic version of Lemma 8 that for any \( (p, l, m, k) \)-design \( D \) and any CIR(\( p^2 + p^{2k} \))-hard boolean function \( g : \{0,1\}^m \rightarrow \{0,1\} \),

\[
|\text{Prob}_{y \in \{0,1\}^{p}}[c(y) = 1] - \text{Prob}_{s \in \{0,1\}^{l}}[c(g_D(s)) = 1]| \leq 1/p
\]

holds for every \( p \)-input circuit \( c \) of size at most \( p^2 \). Now let \( m(n) = 12 \log p(n) \), \( l(n) = 2 \cdot 12^2 \log p(n) \), and \( k(n) = \log p(n) \). Furthermore, by Lemma 11 we know that for all sufficiently large \( n \), a randomly chosen boolean function \( g : \{0,1\}^{m(n)} \rightarrow \{0,1\} \) is CIR(\( 2^{m(n)/4} \))-hard (and thus CIR(\( p^2 + p^{2k(n)} \))-hard) with probability at least \( 1 - e^{-2^{m(n)/4}} \). Hence, the following algorithm together with the NP oracle set

\[
B = \{ (x, r_1, \ldots, r_k) \mid \exists y \in \Sigma^p \langle x \rangle : \| \{1 \leq i \leq k \mid c_{x,y}(r_i) = 1 \} \| \geq k/2 \}
\]

witnesses \( L \subseteq \mathcal{ZPP}^\mathcal{NP} \):

\[
\begin{align*}
\text{input } x, |x| = n; \\
\text{compute a } (p(n), l(n), m(n), k(n))\text{-design } D; \\
\text{choose randomly } g : \{0,1\}^{m(n)} \rightarrow \{0,1\}; \\
\text{if } g \text{ is CIR}(2^{m(n)/4})\text{-hard then } \{\text{this can be found out by asking an NP oracle}\} \\
\text{let } r_1, \ldots, r_{2l(n)} \text{ be the pseudorandom strings produced by } g_D \text{ on all seeds}; \\
\text{if } (x, r_1, \ldots, r_{2l(n)}) \in B \text{ then accept else reject}
\end{align*}
\]

We note that Theorem 23 cannot be further improved to \( \mathcal{AM} \subseteq \mathcal{ZPP}^\mathcal{NP} \) by relativizing techniques since there is an oracle relative to which \( \mathcal{AM} \) is not contained in \( \Sigma^p_2 \) [San89].

From the closure properties of MA (namely that MA is closed under conjunctive truth-table reductions) it easily follows that \( \mathcal{NP}^\mathcal{MA} = \mathcal{MA} \). From Theorem 23 we have \( \mathcal{MA} \subseteq \mathcal{ZPP}^\mathcal{NP} \). Hence, \( \mathcal{NP}^\mathcal{MA} \subseteq \mathcal{ZPP}^\mathcal{NP} \), implying that \( \mathcal{ZPP}^\mathcal{NP} = \mathcal{ZPP}^\mathcal{NP} \). We have proved the following corollary.

Corollary 24 MA \( \cap \) coMA is low for \( \mathcal{ZPP}^\mathcal{NP} \) and, consequently, BPP is low for \( \mathcal{ZPP}^\mathcal{NP} \).
Acknowledgement

The second author would like to thank Lance Fortnow for interesting discussions on the topic of this paper.

References


Liste der bisher erschienenen Ulmer Informatik-Berichte
Einige davon sind per FTP von ftp.informatik.uni-ulm.de erhältlich
Die mit * markierten Berichte sind vergriffen

List of technical reports published by the University of Ulm
Some of them are available by FTP from ftp.informatik.uni-ulm.de
Reports marked with * are out of print

91-01  Ker-I Ko, P. Orponen, U. Schöning, O. Watanabe
        Instance Complexity

91-02* K. Gnaditz, H. Fassbender, H. Vogler
        Compiler-Based Implementation of Syntax-Directed Functional Programming

91-03* Alfons Geser
        Relative Termination

91-04* J. Köbler, U. Schöning, J. Toran
        Graph Isomorphism is low for PP

91-05 Johannes Köbler, Thomas Thierauf
        Complexity Restricted Advice Functions

91-06* Uwe Schöning
        Recent Highlights in Structural Complexity Theory

91-07* F. Green, J. Köbler, J. Toran
        The Power of Middle Bit

91-08* V. Arvind, Y. Han, L. Hamachandra, J. Köbler, A. Lozano, M. Mundhenk, A. Ogiwara, U.
        Schöning, R. Silvestri, T. Thierauf
        Reductions for Sets of Low Information Content

92-01* Vikraman Arvind, Johannes Köbler, Martin Mundhenk
        On Bounded Truth-Table and Conjunctive Reductions to Sparse and Tally Sets

92-02* Thomas Noll, Heiko Vogler
        Top-down Parsing with Simultaneous Evaluation of Noncircular Attribute Grammars

92-03  Fakultät für Informatik
        17. Workshop über Komplexitätstheorie, effiziente Algorithmen und Datenstrukturen

92-04* V. Arvind, J. Köbler, M. Mundhenk
        Lowness and the Complexity of Sparse and Tally Descriptions

92-05* Johannes Köbler
        Locating P/poly Optimally in the Extended Low Hierarchy

92-06* Armin Kühnemann, Heiko Vogler
        Synthesized and inherited functions - a new computational model for syntax-directed semantics

92-07* Heinz Fassbender, Heiko Vogler
        A Universal Unification Algorithm Based on Unification-Driven Leftmost Outermost Narrowing
92-08* Uwe Schöning
On Random Reductions from Sparse Sets to Tally Sets

92-09* Hermann von Hasseln, Laura Martignon
Consistency in Stochastic Network

92-10 Michael Schmitt
A Slightly Improved Upper Bound on the Size of Weights Sufficient to Represent Any Linearly Separable Boolean Function

92-11 Johannes Köbler, Seinosuke Toda
On the Power of Generalized MOD-Casses

92-12 V. Arvind, J. Köbler, M. Mundhenk
Reliable Reductions, High Sets and Low Sets

92-13 Alfons Geser
On a monotonic semantic path ordering

92-14* Joost Engelfriet, Heiko Vogler
The Translation Power of Top-Down Tree-To-Graph Transducers

93-01 Alfred Lupper, Konrad Froitzheim
AppleTalk Link Access Protocol basierend auf dem Abstract Personal Communications Manager

The COCOON Object Model

93-03 Thomas Thierauf, Seinosuke Toda, Osamu Watanabe
On Sets Bounded Truth-Table Reducible to P-selective Sets

93-04 Jin-Yi Cai, Frederic Green, Thomas Thierauf
On the Correlation of Symmetric Functions

93-05 K. Kuhn, M. Reichert, M. Nathe, T. Beuter, C. Heinlein, P. Dadam
A Conceptual Approach to an Open Hospital Information System

93-06 Klaus Gaßner
Rechnerunterstützung für die konzeptuelle Modellierung

93-07 Ulrich Keßler, Peter Dadam
Towards Customizable, Flexible Storage Structures for Complex Objects

94-01 Michael Schmitt
On the Complexity of Consistency Problems for Neurons with Binary Weights

94-02 Armin Kühnemann, Heiko Vogler
A Pumping Lemma for Output Languages of Attributed Tree Transducers

94-03 Harry Buhrman, Jim Kadin, Thomas Thierauf
On Functions Computable with Nonadaptive Queries to NP

94-04 Heinz Faßbender, Heiko Vogler, Andrea Wedel
Implementation of a Deterministic Partial E-Unification Algorithm for Macro Tree Transducers
94-05  V. Arvind, J. Köbler, R. Schuler
On Helping and Interactive Proof Systems

94-06  Christian Kalus, Peter Dadam
Incorporating record subtyping into a relational data model

94-07  Markus Tresch, Marc H. Scholl
A Classification of Multi-Database Languages

94-08  Friedrich von Henke, Harald Rueß
Arbeitstreffen Typtheorie: Zusammenfassung der Beiträge

Construction and Deduction Methods for the Formal Development of Software

94-10  Axel Dold
Formalisierung schematischer Algorithmen

94-11  Johannes Köbler, Osamu Watanabe
New Collapse Consequences of NP Having Small Circuits

94-12  Rainer Schuler
On Average Polynomial Time

94-13  Rainer Schuler, Osamu Watanabe
Towards Average-Case Complexity Analysis of NP Optimization Problems

94-14  Wolfram Schulte, Ton Vullinghs
Linking Reactive Software to the X-Window System

94-15  Alfred Lupper
Namensverwaltung und Adressierung in Distributed Shared Memory-Systemen

94-16  Robert Regn
Verteilte Unix-Betriebssysteme

94-17  Helmut Partsch
Again on Recognition and Parsing of Context-Free Grammars:
Two Exercises in Transformational Programming

94-18  Helmut Partsch
Transformational Development of Data-Parallel Algorithms: an Example

95-01  Oleg Verbitsky
On the Largest Common Subgraph Problem

95-02  Uwe Schöning
Complexity of Presburger Arithmetic with Fixed Quantifier Dimension

95-03  Harry Buhrman, Thomas Thierauf
The Complexity of Generating and Checking Proofs of Membership

95-04  Rainer Schuler, Tomoyuki Yamakami
Structural Average Case Complexity
95-05  Klaus Achatz, Wolfram Schulte
       Architecture Independent Massive Parallelization of Divide-and-Conquer Algorithms

95-06  Christoph Karg, Rainer Schuler
       Structure in Average Case Complexity

95-07  P. Dadam, K. Kuhn, M. Reichert, T. Beuter, M. Nathe
       ADEPT: Ein integrierender Ansatz zur Entwicklung flexibler, zuverlässiger kooperierender
       Assistentensysteme in klinischen Anwendungsumgebungen

95-08  Jürgen Kehrer, Peter Schultess
       Aufbereitung von gescannten Röntgenbildern zur filmlosen Diagnostik

95-09  Hans-Jörg Bartschick, Wolfgang Lindner
       On Sets Turing Reducible to P-Selective Sets

95-10  Boris Hartmann
       Berücksichtigung lokaler Randbedingung bei globaler Zieloptimierung mit neuronalen Netzen
       am Beispiel Truck Backer-Upper

95-12  Klaus Achatz, Wolfram Schulte
       Massive Parallelization of Divide-and-Conquer Algorithms over Powerlists

95-13  Andrea Mößle, Heiko Vogler
       Efficient Call-by-value Evaluation Strategy of Primitive Recursive Program Schemes

95-14  Axel Dold, Friedrich W. von Henke, Holger Pfeifer, Harald Rueß
       A Generic Specification for Verifying Peephole Optimizations

96-01  Erçüment Canver, Jan-Tecker Gayen, Adam Moik
       Formale Entwicklung der Steuerungssoftware für eine elektrisch ortsbediente Weiche mit VSE

96-02  Bernhard Nebel
       Solving Hard Qualitative Temporal Reasoning Problems: Evaluating the Efficiency of Using
       the ORD-Horn Class

96-03  Ton Vullinghs, Wolfram Schulte, Thilo Schwinn
       An Introduction to TkGofe

96-04  Thomas Beuter, Peter Dadam
       Anwendungsspezifische Anforderungen an Workflow-Manegement-Systeme am Beispiel der Domäne
       Concurrent-Engineering

96-05  Gerhard Schellhorn, Wolfgang Ahrendt
       Verification of a Prolog Compiler - First Steps with KIV

96-06  Manindra Agrawal, Thomas Thierau
       Satisfiability Problems

96-07  Vikraman Arvind, Jacobo Torán
       A nonadaptive NC Checker for Permutation Group Intersection

96-08  David Cyrluk, Oliver Möller, Harald Rueß
       An Efficient Decision Procedure for a Theory of Fix-Sized Bitvectors with Composition and
       Extraction
96-09  Bernd Biechele, Dietmar Ernst, Frank Houdek, Joachim Schmid, Wolfram Schulte
Erfahrungen bei der Modellierung eingebetteter Systeme mit verschiedenen SA/RT-Ansätzen

96-10  Falk Bartels, Axel Dold, Friedrich W. von Henke, Holger Pfeifer, Harald Rueß
Formalizing Fixed-Point Theory in PVS

96-11  Axel Dold, Friedrich W. von Henke, Holger Pfeifer, Harald Rueß
Mechanized Semantics of Simple Imperative Programming Constructs

96-12  Axel Dold, Friedrich W. von Henke, Holger Pfeifer, Harald Rueß
Generic Compilation Schemes for Simple Programming Constructs

96-13  Klaus Achatz, Helmut Partsch
From Descriptive Specifications to Operational ones: A Powerful Transformation Rule, its
Applications and Variants

97-01  Jochen Messner
Pattern Matching in Trace Monoids

97-02  Wolfgang Lindner, Rainer Schuler
A Small Span Theorem within P

97-03  Thomas Bauer, Peter Dadam
A Distributed Execution Environment for Large-Scale Workflow Management Systems with
Subnets and Server Migration

97-04  Christian Heinlein, Peter Dadam
Interaction Expressions - A Powerful Formalism for Describing Inter-Workflow Dependencies

97-05  Vikraman Arvind, Johannes Köbler
On Pseudorandomness and Resource-Bounded Measure