Johann Wolfgang Goethe-Universität
Frankfurt am Main

Institut für Informatik
Fachbereich Biologie und Informatik

On Non-Recursive Trade-Offs Between Finite-Turn Pushdown Automata

Andreas Malcher

Nr. 2/04

Frankfurter Informatik-Berichte

Institut für Informatik • Robert-Mayer-Straße 11–15 • 60054 Frankfurt am Main

ISSN 1616-9107

On Non-Recursive Trade-Offs Between Finite-Turn Pushdown Automata

Andreas Malcher
Institut für Informatik, Johann Wolfgang Goethe-Universität
D-60054 Frankfurt am Main, Germany
E-Mail: malcher@psc.informatik.uni-frankfurt.de

Abstract

It is shown that between one-turn pushdown automata (1-turn PDAs) and deterministic finite automata (DFAs) there will be savings concerning the size of description not bounded by any recursive function, so-called non-recursive trade-offs. Considering the number of turns of the stack height as a consumable resource of PDAs, we can show the existence of non-recursive trade-offs between PDAs performing $k+1$ turns and k turns for $k \geq 1$. Furthermore, non-recursive trade-offs are shown between arbitrary PDAs and PDAs which perform only a finite number of turns. Finally, several decidability questions are shown to be undecidable and not semidecidable.

1 Introduction

Descriptional complexity is a field of theoretical computer science where one main question is: How succinctly can a model represent a formal language in comparison with other models? Basic and early results are from Meyer and Fischer [10] from 1971. They investigated regular languages and showed that there are languages being recognized by a nondeterministic finite automaton (NFA) with n states such that every deterministic finite automaton (DFA) recognizing these languages will need 2^n states. Beyond this trade-off bounded by an exponential function, Meyer and Fischer proved that between context-free grammars and DFAs there exists a trade-off which is not bounded by any recursive function, a so-called non-recursive trade-off. Additional non-recursive trade-offs are known to exist between pushdown automata (PDAs) and deterministic PDAs (DPDAs), between DPDAs and unambiguous PDAs (UPDAs), between UPDAs and PDAs and many other models. A survey of results concerning the descriptional complexity of machines with limited resources, including non-recursive trade-offs between various models, may be found in [2]. A thorough discussion of the phenomenon of non-recursive trade-offs may be found in [7].

Restricting a PDA such that the height of its stack is only allowed to increase and then to decrease, thus performing only one turn, leads to the definition of one-turn PDAs [3]. It is known that these PDAs can be grammatically characterized by linear context-free grammars. It is an obvious generalization to consider PDAs which are allowed to perform a finite number of turns, so-called k-turn PDAs [3]. If it is additionally required for a k-turn PDA to empty its stack up to the initial stack symbol before starting the next turn, the resulting model is called strong k-turn PDA [1]. Both models can be grammatically characterized by ultralinear and metalinear context-free grammars, respectively. The definition of the models will be given in the next chapter.

The intention of this paper is to show non-recursive trade-offs between finite-turn PDAs and DFAs, between PDAs performing $k+1$ and k turns, and between arbitrary PDAs and finite-turn PDAs. To this end we are using a generalization of a technique which was first presented by Hartmanis [5]. A combination of this technique with some old results on ultralinear grammars [3] and some new considerations leads to the desired non-recursive trade-offs. Finally, certain decidability questions for finite-turn PDAs are shown to be undecidable and not semidecidable.

2 Preliminaries and Definitions

Let Σ^* denote the set of all words over the finite alphabet Σ, $\Sigma^+ = \Sigma^* \setminus \{\epsilon\}$. Let REG, LCF, CF, RE denote the families of regular, linear context-free, context-free and recursively enumerable languages. We assume that the reader is familiar with the common notions of formal language theory as presented in [6]. Let S be a set of recursively enumerable languages. Then S is said to be a property of the recursively enumerable languages. A set L has the property S, if $L \in S$. Let L_S be the set $\{< M >| T(M) \in S\}$ where $< M >$ is an encoding of a Turing machine M. If L_S is recursive, we say the property S is decidable; if L_S is recursively enumerable, we say the property S is semidecidable.

In the sequel we will use the set of valid computations of a Turing machine. Details are presented in [5] and [6]. The definition of a Turing machine and of an instantaneous description (ID) of a Turing machine may be found in [6].

Definition: Let $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ be a deterministic Turing machine.

$$\begin{aligned}
\text{VALC}[M] &= \{ID_0(x)\#ID_1(x)^R\#ID_2(x)\#ID_3(x)^R\#\ldots\#ID_n(x)\# \mid \\
&\quad x \in \Sigma^*, ID_0(x) \in q_0\Sigma^* \text{ is an initial ID}, \\
&\quad ID_n(x) \in \Gamma^*F\Gamma^* \text{ is an accepting ID}, \\
&\quad ID_{i+1}(x) \in \Gamma^*Q\Gamma^* \text{results from } ID_i(x), \text{i.e., } ID_i(x) \vdash^{M} ID_{i+1}(x)\} \\
\text{INVALC}[M] &= \Lambda^* \setminus \text{VALC}[M] \text{ with respect to a coding alphabet } \Lambda.
\end{aligned}$$

Definition: [4] A context-free grammar $G = (V, \Sigma, S, P)$ is metalinear if all rules of P are of the following forms

$$\begin{array}{ll}
S \to A_1A_2\ldots A_m, & A_i \in V \setminus \{S\}, \\
A \to w_1Bw_2, & A, B \in V \setminus \{S\}, w_1, w_2 \in \Sigma^*, \\
A \to w, & w \in \Sigma^*.
\end{array}$$

The width of G is $\max\{m \mid S \to A_1A_2\dots A_m\}$. L is metalinear of width k if $L = L(G)$ for some metalinear grammar G of width k. By $\mathcal{L}(k\text{-}LCG)$ we denote the set of languages accepted by metalinear grammars of width k. $\mathcal{L}(META\text{-}LCG)$ denotes the set of languages accepted by metalinear grammars.

It is easily observed that metalinear grammars of width 1 are exactly linear context-free grammars.

Definition: [1, 3] A context-free grammar $G = (V, \Sigma, S, P)$ is *ultralinear* if V is a union of disjoint (possibly empty) subsets $V_0, \dots, V_n$ of V with the following property. For each V_i and each $A \in V_i$, each production with left side A is either of the form

$A \to w_1Bw_2$ with $B \in V_i$ and $w_1, w_2 \in \Sigma^*$, or of the form

$A \to w$ with $w \in (\Sigma \cup V_0 \cup \dots \cup V_{i-1})^*$.

$\{V_0, \dots, V_n\}$ is called an *ultralinear decomposition.* A language is said to be ultralinear if it is generated by some ultralinear grammar. $\mathcal{L}(ULTRA\text{-}LCG)$ denotes the set of languages accepted by ultralinear grammars.

Definition: [1] Let $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ be a pushdown automaton. A sequence of instantaneous descriptions (IDs) on M $(q_1, w_1, \alpha_1)\dots(q_k, w_k, \alpha_k)$ is called *one-turn* if there exists $i \in \{1, \dots, k\}$ such that

$$|\alpha_1| \leq \dots \leq |\alpha_{i-1}| \leq |\alpha_i| > |\alpha_{i+1}| \geq \dots \geq |\alpha_k|$$

A sequence of IDs $S_0, \dots, S_m$ is called *strong k-turn* if there are integers $0 = i_0, \dots, i_l = m$ with $l \leq k$ such that for $j = 0, \dots, l-1$ holds:

(1) $S_{i_j}, \dots, S_{i_{j+1}}$ is one-turn

(2) $S_{i_j} = (q, w, Z_0)$ for some $q \in Q$ and $w \in \Sigma^*$

If only the first condition is fulfilled, then the sequence of IDs is called k-turn. M is a strong k-turn pushdown automaton if every word $w \in T(M)$ is accepted by a sequence of IDs which is strong k-turn. A k-turn pushdown automaton is defined analogously. By $\mathcal{L}(strong\text{-}k\text{-}turn\text{-}PDA)$ and $\mathcal{L}(k\text{-}turn\text{-}PDA)$ we denote the set of languages accepted by strong k-turn PDAs and k-turn PDAs, respectively.

Thus, strong k-turn PDAs are allowed to make a new turn only if the stack is empty up to the initial stack symbol whereas k-turn PDAs can make new turns not depending on the stack height. The following characterization of metalinear languages by strong k-turn PDAs and of ultralinear languages by k-turn PDAs may be found in [1] and [3], respectively.

Theorem 1 *(1)* $\mathcal{L}(k\text{-}LCG) = \mathcal{L}(strong\text{-}k\text{-}turn\text{-}PDA)$

(2) $L \in \mathcal{L}(META\text{-}LCG) \Leftrightarrow \exists$ *strong k-turn PDA M such that* $T(M) = L$.

(3) $L \in \mathcal{L}(ULTRA\text{-}LCG) \Leftrightarrow \exists$ *k-turn PDA M such that* $T(M) = L$.

Theorem 2 *[1] Let A be a k-turn PDA. Then there are homomorphisms h_1, h_2 and a regular language R such that $T(A) = h_1(h_2^{-1}(D_{2,k}) \cap R)$ with $D_{2,k} = D_2 \cap (\{(, [\}^* \{),]\}^*)^k$ and D_2 denotes the Dyck language with 2 types of balanced parentheses.*

Concerning the notations and definitions of descriptional complexity we largely follow the presentation in [2]. A descriptional system D is a recursive set of finite descriptors (e.g. automata or grammars) relating each $A \in D$ to a language $T(A)$. It is additionally required that each descriptor $A \in D$ can be effectively converted to a Turing machine M_A such that $T(M_A) = T(A)$. The language class being described by D is $T(D) = \{T(A) \mid A \in D\}$. For every language L we define $D(L) = \{A \in D \mid T(A) = L\}$. A complexity measure for D is a total, recursive, and finite-to-one function $|\cdot| : D \to \mathbb{N}$ such that the descriptors in D are recursively enumerable in order of increasing complexity. Comparing two descriptional systems D_1 and D_2, we assume that $T(D_1) \cap T(D_2)$ is not finite. We say that a function $f : \mathbb{N} \to \mathbb{N}$, $f(n) \geq n$ is an *upper bound* for the trade-off when changing from a minimal description in D_1 for an arbitrary language to an equivalent minimal description in D_2, if for all $L \in T(D_1) \cap T(D_2)$ the following holds:

$$\min\{|A| \mid A \in D_2(L)\} \leq f(\min\{|A| \mid A \in D_1(L)\}).$$

If no recursive function is an upper bound for the trade-off between two descriptional systems D_1 and D_2, we say the trade-off is non-recursive and write $D_1 \overset{nonrec}{\longrightarrow} D_2$.

3 Non-Recursive Trade-Offs

In [9] the following generalization of Hartmanis' technique to establish non-recursive trade-offs is proven. Additional information on techniques to prove non-recursive trade-offs may be found in [7].

Theorem 3 *Let D_1 and D_2 be two descriptional systems. If for every Turing machine M a language $L_M \in T(D_1)$ and a descriptor $A_M \in D_1$ for L_M can be effectively constructed such that $L_M \in T(D_2) \Leftrightarrow T(M)$ is finite, then the trade-off between D_1 and D_2 is non-recursive.*

Let $L = \text{INVALC}[M] \subseteq \Lambda^*$ and $\{a, b, c\} \cap \Lambda = \emptyset$. Then we define

$$\tilde{L} = \{a^n LcLb^n \mid n \geq 1\} \quad .$$

Lemma 1 *Let M be a Turing machine and $k \geq 0$. Then the following pushdown automata can be effectively constructed:*

(1) A strong $(k+1)$-turn PDA A_{k+1} accepting $(Lc)^{k+1}$.

(2) A strong infinite-turn PDA A_+ accepting $(Lc)^+$.

(3) A 2-turn PDA $\tilde{A}$ accepting $\tilde{L}$.

Proof: It is shown in [6] that INVALC$[M]$ is a context-free language. Taking a close look at the construction we can show that INVALC$[M]$ is the union of languages which are accepted by finite automata or 1-turn PDAs. Since the linear context-free languages are effectively closed under union, we can construct a 1-turn PDA A_1 such that $T(A_1) = \text{INVALC}[M]c = (Lc)^1$. For $k \geq 1$ the language $(Lc)^{k+1}$ can be represented as the marked concatenation of languages which are accepted by one-turn PDAs. Thus, it is easy to construct a strong $(k+1)$-turn PDA A_{k+1} accepting $(Lc)^{k+1}$. Analogously, a strong PDA A_+ making infinite turns can be constructed accepting $(Lc)^+$. The language LcL is accepted by a 2-turn PDA. Thus, a 2-turn PDA $\tilde{A}$ accepting $\tilde{L}$ can be easily constructed. □

Theorem 4 (Ginsburg, Spanier [3]) *Let Σ be a finite alphabet, and let $c \notin \Sigma$. Let $S \subseteq \Sigma^*$. Then $(Sc)^+ \in \mathcal{L}(ULTRA\text{-}LCG) \Leftrightarrow S$ is regular.*

Lemma 2 *Let M be a Turing machine and $k \geq 1$. Then*

(1) $Lc \in REG \Leftrightarrow T(M)$ is finite

(2) $(Lc)^{k+1} \in \mathcal{L}(k\text{-turn } PDA) \Leftrightarrow T(M)$ is finite

(3) $(Lc)^+ \in \mathcal{L}(\text{finite-turn } PDA) \Leftrightarrow T(M)$ is finite

(4) $\tilde{L} \in \mathcal{L}(\text{strong infinite-turn } PDA) \Leftrightarrow T(M)$ is finite

Proof:

(1) If $T(M)$ is finite, then VALC$[M]$ is a finite set. This implies that the complement $L = \text{INVALC}[M]$ and thus Lc are regular. In [6] it is proven that VALC$[M] \in$ CF $\Leftrightarrow T(M)$ is finite. Then, the first claim is easy to show.

(2) If $T(M)$ is finite, then $(Lc)^{k+1}$ is a regular language and thus can be accepted by k-turn PDA. We next show that $(Lc)^{k+1} \notin \mathcal{L}(k\text{-turn } PDA)$ provided that $T(M)$ is infinite. If $T(M)$ is infinite, then INVALC$[M] \in \mathcal{L}(LCG) \setminus$ REG. By the definition of the rank r of a ultralinear language [3] we obtain that $r(\text{INVALC}[M]) = 1$. Applying Corollary 1 from [3] results in $r((Lc)^{k+1}) = k+1$. We now assume that $(Lc)^{k+1} \in \mathcal{L}(k\text{-turn } PDA)$. Thus there is a k-turn PDA A such that $T(A) = (Lc)^{k+1}$. Due to Theorem 2 $(Lc)^{k+1}$ then has a representation as $h_1(h_2^{-1}(D_{2,k}) \cap R)$ with homomorphisms h_1, h_2 and a regular set R. It can be easily shown that $r(D_{2,k}) \leq k$. Thus $r((Lc)^{k+1}) \leq k$, since the operations homomorphism, inverse homomorphism and intersection with regular languages do not increase the rank of a language due to Theorem 4.2 in [3]. This is a contradiction to the above fact that $r((Lc)^{k+1}) = k+1$.

(3) This claim follows easily from (1) and Theorem 4.

(4) If $T(M)$ is finite, then $\tilde{L}$ is a linear language and a strong infinite-turn PDA accepting $\tilde{L}$ can be easily constructed. We next show that the fact that $T(M)$ is infinite implies that $\tilde{L} \notin \mathcal{L}(\text{strong infinite-turn } PDA)$. We first assume that $\tilde{L} \in$

$\mathcal{L}$(strong finite-turn PDA). Then $\tilde{L}$ can be generated by a metalinear grammar of width k. Thus, each $w \in \tilde{L}$ has a derivation $S \Rightarrow A_1 A_2 \ldots A_m \Rightarrow^* w$ with $m \leq k$ where each A_i $(1 \leq i \leq m)$ generates a linear language. There exists at least one non-terminal A_i from which words containing infinitely many a's can be derived. This A_i generates a linear language. Let n be the constant number resulting from Ogden's lemma for $L(A_i)$ where all a's are marked. We now choose a word $w \in \tilde{L}$ such that w contains a subword $w' \in L(A_i)$ with $|w'|_a \geq n$. Applying Ogden's lemma to $L(A_i)$ we obtain that either a's and b's or a's and no b's are pumped. This leads, in the latter case, to words in $\tilde{L}$ with different numbers of a's and b's which is a contradiction. If a's and b's are pumped, then $L(A_i)$ generates a linear language which is a subset of $\{a\}^* LcL \{b\}^*$. We now consider the set N of all non-terminals A_i from which words containing infinitely many a's can be derived. By the preceding considerations we obtain that each $A \in N$ generates a linear subset of $\{a\}^* LcL \{b\}^*$. Thus, $\bigcup_{A \in N} L(A) = M_1 LcL M_2$ with $M_1 \subseteq \{a\}^*$ and $M_2 \subseteq \{b\}^*$. Since the set of linear languages is closed under union, left and right quotient with regular languages and concatenation with regular languages, we obtain that $LcLc$ is a linear language and thus accepted by a 1-turn PDA. Applying (2) we have that $T(M)$ is finite which is a contradiction.

We next show that $\tilde{L}$ is not accepted by a strong infinite-turn PDA. If $\tilde{L}$ is accepted by a strong infinite-turn PDA A, we can conclude that the number of turns needed to accept an input increases with the length of the input. Otherwise, the number of turns could be bounded by a fixed number and thus $\tilde{L}$ would be accepted by a strong finite-turn PDA which is a contradiction. Let n be an arbitrary natural number. If we choose a word $w \in LcL$ large enough with $a^n w b^n \in \tilde{L}$, then a combination of some state q and the initial stack symbol is attained during A's course of computation at least two times. The subword v read between these two occurrences then can be repeated arbitrarily often without affecting the acceptance of the input. If v contains a's, b's or both a's and b's, then A accepts inputs with a different number of a's and b's or inputs with the wrong format which is a contradiction. If v contains no a's and b's, then v also contains no c and w.l.o.g. it can be assumed that v is located in the first L of LcL. We are now using an incompressibility argument. More general information on Kolmogorov complexity and the incompressibility method may be found in [8]. Let $a^n w'$ be the subword read until the combination of the state q and the initial stack symbol occurs for the first time. Then $a^n w' \#\# c b^n \in \tilde{L}$. But this implies that n can be described by a program simulating A starting in state q with the initial stack symbol and reading the input $\#\# c b^*$ until an accepting state in A is attained. Thus, n is the number of b's read until the input is accepted. The Kolmogorov complexity $C(n)$ of n, i.e. the minimal size of a program describing n, is then bounded by the description sizes of A, q and the above program. Obviously, these sizes are bounded by a constant number c not depending on n. Thus, $C(n) \leq c$. Due to [8] there exist natural numbers such that $C(n) \geq \log n$. If we choose such a number and consider a word $a^n w b^n \in \tilde{L}$ being large enough, we get a contradiction.

□

Combining the results of Theorem 3, Lemma 1 and Lemma 2 we get the following non-recursive trade-offs which are pictorially summarized in Fig. 1.

Theorem 5 *Let $k \geq 1$:*

- *(strong) 1-turn PDA* $\overset{nonrec}{\longrightarrow}$ *NFA using* $L_M = Lc$
- *(strong)* $(k+1)$*-turn PDA* $\overset{nonrec}{\longrightarrow}$ *(strong) k-turn PDA using* $L_M = (Lc)^{k+1}$
- $(k+1)$*-turn PDA* $\overset{nonrec}{\longrightarrow}$ *strong* $(k+1)$*-turn PDA using* $L_M = \tilde{L}$
- *(strong) finite-turn PDA* $\overset{nonrec}{\longrightarrow}$ *(strong) k-turn PDA using* $L_M = (Lc)^{k+1}$
- *finite-turn PDA* $\overset{nonrec}{\longrightarrow}$ *strong finite-turn PDA using* $L_M = \tilde{L}$
- *(strong) infinite-turn PDA* $\overset{nonrec}{\longrightarrow}$ *(strong) finite-turn PDA using* $L_M = (Lc)^+$
- *infinite-turn PDA* $\overset{nonrec}{\longrightarrow}$ *strong infinite-turn PDA using* $L_M = \tilde{L}$

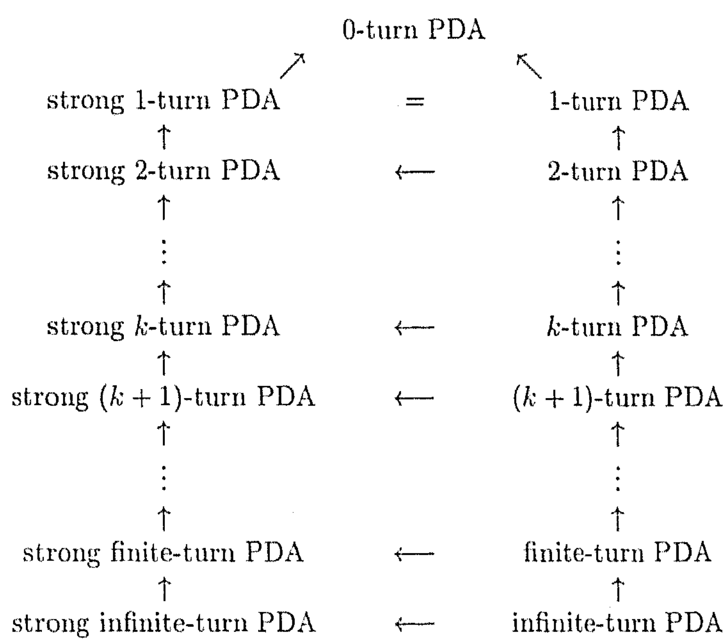

Figure 1: Non-recursive trade-offs between PDAs with different numbers of turns allowed

Remark: It should be noted that the non-recursive trade-offs between strong $(k+1)$-turn PDAs and strong k-turn PDAs could have been shown using a result from [4] which states that $(Lc)^{k+1} \in \mathcal{L}(k\text{-}LCG)$ if and only if $L \in \text{REG}$. The approach presented in this paper extends the non-recursive trade-offs to arbitrary k-turn PDAs.

4 Decidability Questions

The fact that the set of invalid computations can be recognized by a 1-turn PDA allows us to simply prove that certain decidability questions for strong k-turn PDAs are not decidable and not even semidecidable. The results obviously hold for k-turn PDAs and arbitrary PDAs as well.

Lemma 3 *Let M be a Turing machine. It is not semidecidable whether $T(M) = \emptyset$ or $T(M)$ is finite.*

Proof: The lemma can be easily seen using Rice's theorem for recursively enumerable index sets [6]. □

Theorem 6 *Let $k, k' \geq 1$ be two integers. It is not semidecidable for arbitrary strong k-turn PDAs A and strong k'-turn PDAs A' whether*

(1) $T(A) = \Sigma^*$

(2) $T(A) = T(A')$, $T(A) \subseteq T(A')$

(3) $T(A) \in REG$

(4) $T(A) \in \mathcal{L}($*strong* $(k-1)$-*turn PDA*$)$

Proof: Let M be an arbitrary Turing machine. By Lemma 1, we can construct a 1-turn PDA A accepting INVALC$[M]$. Suppose that the first question is semidecidable. Then we can semidecide whether INVALC$[M] = \Sigma^*$, or equivalently, whether VALC$[M] = \emptyset$. Thus, we can semidecide whether an arbitrary Turing machine accepts the empty language which is a contradiction to the above lemma. The questions of (2) can be easily reduced to the first question. If we could semidecide question (3), we could semidecide whether M accepts a finite language due to Lemma 2(1). This again contradicts the above lemma. The non-semidecidability of (4) is shown similarly considering Lemma 2(2). □

It can be learned from the proofs of (3) and (4) that the existence of non-recursive trade-offs implies that it is not semidecidable for a PDA with a certain number of turns allowed whether its language accepted could be accepted by any other PDA with a smaller number of turns. For example, it is not semidecidable whether a language described by an infinite-turn PDA can be accepted by a finite-turn PDA. Thus, the minimal number of turns needed to accept a context-free language cannot be determined algorithmically.

References

[1] L. Balke, K.H. Böhling: "Einführung in die Automatentheorie und Theorie formaler Sprachen," BI Wissenschaftsverlag, Mannheim, 1993

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

[3] S. Ginsburg, E.H. Spanier: "Finite-turn pushdown automata," SIAM Journal on Control, 4(3): 429–453, 1966

[4] S.A. Greibach: "The unsolvability of the recognition of linear context-free languages," Journal of the ACM, 13(4): 582–587, 1966

[5] J. Hartmanis: "On the succinctness of different representations of languages," SIAM Journal on Computing, 9(1): 114–120, 1980

[6] J.E. Hopcroft, J.D. Ullman: "Introduction to Automata Theory, Languages and Computation," Addison-Wesley, Reading MA, 1979

[7] M. Kutrib: "The phenomenon of non-recursive trade-offs," In L. Ilie, D. Wotschke (Eds.): "Sixth International Workshop on Descriptional Complexity of Formal Systems (DCFS 2004)," Report No. 619, Department of Computer Science, The University of Western Ontario, London, Ontario, Canada, 83–97, 2004

[8] M. Li, P. Vitányi: "An Introduction to Kolmogorov Complexity and Its Applications," Springer-Verlag, New York, 1993

[9] A. Malcher: "Descriptional complexity of cellular automata and decidability questions", Journal of Automata, Languages and Combinatorics, 7(4): 549–560, 2002

[10] A.R. Meyer, M.J. Fischer: "Economy of descriptions by automata, grammars, and formal systems," IEEE Symposium on Foundations of Computer Science, 188–191, 1971

Interne Berichte am Fachbereich Informatik
Johann Wolfgang Goethe-Universität Frankfurt

1/1987 Risse, Thomas:
On the number of multiplications needed to evaluate the reliability of k-out-of-n systems

2/1987 Roll, Georg [u.a.]:
Ein Assoziativprozessor auf der Basis eines modularen vollparallelen Assoziativspeicherfeldes

3/1987 Waldschmidt, Klaus ; Roll, Georg:
Entwicklung von modularen Betriebssystemkernen für das ASSKO-Multi-Mikroprozessorsystem

4/1987 Workshop über Komplexitätstheorie, effiziente Algorithmen und Datenstrukturen:
3.2.1987, Universität Frankfurt/Main

5/1987 Seidl, Helmut:
Parameter-reduction of higher level grammars

6/1987 Kemp, Rainer:
On systems of additive weights of trees

7/1987 Kemp, Rainer:
Further results on leftist trees

8/1987 Seidl, Helmut:
The construction of minimal models

9/1987 Weber, Andreas ; Seidl, Helmut:
On finitely generated monoids of matrices with entries in N

10/1987 Seidl, Helmut:
Ambiguity for finite tree automata

1/1988 Weber, Andreas:
A decomposition theorem for finite-valued transducers and an application to the equivalence problem

2/1988 Roth, Peter:
A note on word chains and regular languages

3/1988 Kemp, Rainer:
Binary search trees for d-dimensional keys

4/1988 Dal Cin, Mario:
On explicit fault-tolerant, parallel programming

5/1988 Mayr, Ernst W.:
Parallel approximation algorithms

6/1988 Mayr, Ernst W.:
Membership in polynomial ideals over Q is expotential space complete

1/1989 Lutz, Joachim [u.a.]:
Parallelisierungskonzepte für ATTEMPO-2

2/1989 Lutz, Joachim [u.a.]:
Die Erweiterung der ATTEMPO-2 Laufzeitbibliothek

3/1989 Kemp, Rainer:
A One-to-one Correspondence between Two Classes of Ordered Trees

4/1989 Mayr, Ernst W. ; Plaxton, C. Greg:
Pipelined Parallel Prefix Computations, and Sorting on a Pipelined Hypercube

5/1989 Brause, Rüdiger:
Performance and Storage Requirements of Topology-conserving Maps for Robot Manipulator Control

6/1989 Roth, Peter:
Every Binary Pattern of Length Six is Avoidable on the Two-Letter Alphabet

7/1989 Mayr, Ernst W.:
Basic Parallel Algorithms in Graph Theory

8/1989 Brauer, Johannes:
A Memory Device for Sorting

1/1990 Vollmer, Heribert:
Subpolynomial Degrees in P and Minimal Pairs for L

2/1990 Lenz, Katja:
The Complexity of Boolean Functions in Bound Depth Circuits over Basis $\{\wedge, \oplus\}$

3/1990 Becker, Bernd ; Hahn R. ; Krieger, R. ; Sparmann, U.:
Structure Based Methods for Parallel Pattern Fault Simulation in Combinational Circuits

4/1990 Goldstine, J. ; Kintala, C.M.R. ; Wotschke D.:
On Measuring Nondeterminism in Regular Languages

5/1990 Goldstein, J. ; Leung, H. ; Wotschke, D.:
On the Relation between Ambiguity and Nondeterminism in Finite Automata

1/1991 Brause, Rüdiger:
Approximator Networks and the Principles of Optimal Information Distribution

2/1991 Brauer, Johannes ; Stuchly, Jürgen:
HyperEDIF: Ein Hypertext-System für VLSI Entwurfsdaten

3/1991 Brauer, Johannes:
Repräsentation von Entwurfsdaten als symbolische Ausdrücke

4/1991 Trier, Uwe:
Additive Weights of a Special Class of Nonuniformly Distributed Backtrack Trees

5/1991 Dömel, P. [u.a.]:
Concepts for the Reuse of Communication Software

6/1991 Heistermann, Jochen:
Zur Theorie genetischer Algorithmen

7/1991 Wang, Alexander [u.a.]:
Embedding complete binary trees in faulty hypercubes

1/1992 Brause, Rüdiger:
The Minimum Entropy Network

2/1992 Trier, Uwe:
Additive Weights Under the Balanced Probability Model

3/1992 Trier, Uwe:
(Un)expected path lengths of asymetric binary search trees

4/1992 Coen Alberto ; Lavazza, Luigi ; Zicari, Roberto:
Assuring type-safety of object oriented languages

5/1992 Coen, Alberto ; Lavazza, Luigi ; Zicari, Roberto:
Static type checking of an object-oriented database schema

6/1992 Coen, Alberto ; Lavazza, Luigi ; Zicari, Roberto:
Overview and progress report of the ESSE project : Supporting object-oriented database schema analysis and evolution

7/1992 Schmidt-Schauß, Manfred:
Some results for unification in distributive equational theories

8/1992 Mayr, Ernst W. ; Werchner, Ralph:
Divide-and-conquer algorithms on the hypercube

1/1993 Becker, Bernd ; Drechsler, Rolf ; Hengster, Harry:
Local circuit transformations preserving robust path-delay-fault testability

2/1993 Krieger, Rolf ; Becker, Bernd ; Sinković, Robert:
A BDD-based algorithmen for computation of exact fault detection probabilities

3/1993 Mayr, Ernst W. ; Werchner, Ralph:
Optimal routing of parentheses on the hypercube

4/1993 Drechsler, Rolf ; Becker, Bernd:
Rapid prototyping of fully testable multi-level AND/EXOR networks

5/1993 Becker, Bernd ; Drechsler, Rolf:
On the computational power of functional decision diagrams

6/1993 Berghoff, P. ; Dömel, P. ; Drobnik, O. [u.a.]:
Development and management of communication software systems

7/1993 Krieger, Rolf ; Hahn, Ralf ; Becker Bernd:
test_circ : Ein abstrakter Datentyp zur Repräsentation von hierarchischen Schaltkreisen (Benutzeranleitung)

8/1993 Krieger, Rolf ; Becker, Bernd ; Hengster, Harry:
lgc++ : Ein Werkzeug zur Implementierung von Logiken als abstrakte Datentypen in C++ (Benutzeranleitung)

9/1993 Becker, Bernd ; Drechsler, Rolf ; Meinel, Christoph:
On the testability of circuits derived from binary decision diagrams

10/1993 Liu, Ling ; Zicari, Roberto ; Liebherr, Karl ; Hürsch, Walter:
Polymorphic reuse mechanism for object-oriented database specifications

11/1993 Ferrandina, Fabrizio ; Zicari, Roberto:
Object-oriented database schema evolution: are lazy updates always equivalent to immediate updates ?

12/1993 Becker, Bernd ; Drechsler, Rolf ; Werchner, Ralph:
On the Relation Between BDDs and FDDs

13/1993 Becker, Bernd ; Drechsler, Rolf:
Testability of circuits derived from functional decision diagrams

14/1993 Drechsler, R. ; Sarabi, A. ; Theobald, M. ; Becker, B. ; Perkowski, M.A.:
Efficient repersentation and manipulation of switching functions based on ordered Kronecker functional decision diagrams

15/1993 Drechsler, Rolf ; Theobald, Michael ; Becker, Bernd:
Fast FDD based Minimization of Generalized Reed-Muller Forms

1/1994 Ferrandina, Fabrizio ; Meyer, Thorsten ; Zicari, Roberto:
Implementing lazy database updates for an object database system

2/1994 Liu, Ling ; Zicari, Roberto ; Hürsch, Walter ; Liebherr, Karl:
The Role of Polymorhic Reuse mechanism in Schema Evolution in an Object-oriented Database System

3/1994 Becker, Bernd ; Drechsler, Rolf ; Theobald, Michael:
Minimization of 2-level AND/XOR Expressions using Ordered Kronecker Functional Decision Diagrams

4/1994 Drechsler, R. ; Becker, B. ; Theobald, M. ; Sarabi, A. ; Perkowski, M.A.:
On the computational power of Ordered Kronecker Functional Decision Diagrams

5/1994 Even, Susan ; Sakkinen, Marku:
The safe use of polymorphism in the O2C database language

6/1994 GI/ITG-Workshop:
Anwendungen formaler Methoden im Systementwurf : 21. und 22. März 1994

7/1994 Zimmermann, M. ; Mönch, Ch. [u.a.]:
Die Telematik-Klassenbibliothek zur Programmierung verteilter Anwendungen in C++

8/1994 Zimmermann, M. ; Krause, G.:
Eine konstruktive Beschreibungsmethodik für verteilte Anwendungen

9/1994 Becker, Bernd ; Drechsler, Rolf:
How many Decomposition Types do we need ?

10/1994 Becker, Bernd ; Drechsler, Rolf:
Sympathy: Fast Exact Minimization of Fixed Polarity Reed-Muller Expression for Symmetric Functions

11/1994 Drechsler, Rolf ; Becker, Bernd ; Jahnke, Andrea:
On Variable Ordering and Decompostion Type Choice in OKFDDs

12/1994 Schmidt-Schauß:
Unification of Stratified Second-Order Terms

13/1994 Schmidt-Schauß:
An Algorithmen for Distributive Unification

14/1994 Becker, Bernd ; Drechsler, Rolf:
Synthesis for Testability: Circuit Derived from ordered Kronecker Functional Decision Diagrams

15/1994 Bär, Brigitte:
Konformität von Objekten in offenen verteilten Systemen

16/1994 Seidel, T. ; Puder, A. ; Geihs, K. ; Gründer, H.:
Global object space: Modell and Implementation

17/1994 Drechsler, Rolf ; Esbensen, Henrik ; Becker, Bernd:
Genetic algorithms in computer aided design of integrated circuits

1/1995 Schütz, Marko:
The $G^{\#}$-Machine: efficient strictness analysis in Haskell

2/1995 Henning, Susanne ; Becker, Bernd:
GAFAP: A Linear Time Scheduling Approach for High-Level-Synthesis

3/1995 Drechsler, Rolf ; Becker, Bernd ; Göckel, Nicole:
A Genetic Algorithm for variable Ordering of OBDDs

4/1995 Nebel, Markus E.:
Exchange Trees, eine Klasse Binärer Suchbäume mit Worst Case Höhe von $\log(n)$

5/1995 Drechsler, Rolf ; Becker, Bernd:
Dynamic Minimization of OKFDDs

6/1995 Breché, Philippe ; Ferrandina, Fabrizio ; Kuklok, Martin:
Simulation of Schema and Database Modification using Views

7/1995 Breché, Philippe ; Wörner, Martin:
Schema Update Primitives for ODB Design

8/1995 Schmidt-Schauß, Manfred:
On the Sematics and Interpretation of Rule Based Programs with Static Global Variables

9/1995 Rußmann, Arnd:
Adding Dynamic Actions to LL(k) Parsers

10/1995 Rußmann, Arnd:
Dynamic LL(k) Parsing

11/1995 Leyendecker, Thomas ; Oehler, Peter ; Waldschmidt, Klaus:
Spezifikation hybrider Systeme

12/1995 Cerone, Antonio ; Maggiolo-Schettini, Andrea:
Time-based Expressivity of Times Petri Nets

1/1996 Schütz, Marko ; Schmidt-Schauß, Manfred:
A Constructive Calculus Using Abstract Reduction for Context Analysis (nicht erschienen)

2/1996 Schmidt-Schauß, Manfred:
CPE: A Calculus for Proving Equivalence of Expressions in a Nonstrict Functional Language

1/1997 Kemp, Rainer:
On the Expected Number of Nodes at Level k in 0-balanced Trees

2/1997 Nebel, Markus:
New Results on the Stack Ramification of Binary Trees

3/1997 Nebel, Markus:
On the Average Complexity of the Membership Problem for a Generalized Dyck Language

4/1997 Liebehenschel, Jens:
Ranking and Unranking of Lexicographically Ordered Words: An Average-Case Analysis

5/1997 Kappes, Martin:
On the Generative Capacity of Bracketed Contextual Grammars

1/1998 Arlt, B. ; Brause, R.:
The Principal Independent Components of Images. *Elektronisch publiziert unter URL http://www.informatik.uni-frankfurt.de/fbreports/fbreport1-98.ps.gz*

2/1998 Miltrup, Matthias ; Schnitger, Georg:
Large Deviation Results for Quadratic Forms

3/1998 Miltrup, Matthias ; Schnitger, Georg:
Neural Networks and Efficient Associative Memory

4/1998 Kappes, Martin:
Multi-Bracketed Contextual Grammars

5/1998 Liebehenschel, Jens:
Lexicographical Generation of a Generalized Dyck Language

6/1998 Kemp, Rainer:
On the Joint Distribution of the Nodes in Uniform Multidimensional Binary Trees

7/1998 Liebehenschel, Jens:
Ranking and Unranking of a Generalized Dyck Language

8/1998 Grimm, Christoph ; Waldschmidt, Klaus:
Hybride Datenflußgraphen

9/1998 Kappes, Martin:
Multi-Bracketed Contextual Rewriting Grammars

1/1999 Kemp, Rainer:
On Leftist Simply Generated Trees

2/1999 Kemp, Rainer:
A One-to-one Correspondence Between a Class of Leftist Trees and Binary Trees

3/1999 Kappes, Martin:
Combining Contextual Grammars and Tree Adjoining Grammars

4/1999 Kappes, Martin:
Descriptional Complexity of Deterministic Finite Automata with Multiple Initial States

5/1999 Nebel, Markus E.:
New Knowledge on AVL-Trees

6/1999 Manfred Schmidt-Schauß, Marko Schütz (editors):
13^{th} International Workshop on Unification

7/1999 Brause, R.; Langsdorf, T.; Hepp, M.:
Credit Card Fraud Detection by Adaptive Neural Data Mining. *Elektronisch publiziert unter URL http://www.informatik.uni-frankfurt.de/fbreports/fbreport7-99.ps.gz*

8/1999 Kappes, Martin:
External Multi-Bracketed Contextual Grammars

9/1999 Priese, Claus P.:
A Flexible Type-Extensible Object-Relational DataBase Wrapper-Architecture

10/1999 Liebehenschel, Jens:
The Connection between Lexicographical Generation and Ranking

11/1999 Brause, R.; Arlt, B.; Tratar, E.:
A Scale-Invariant Object Recognition System for Content-based Queries in Image Databases. *Elektronisch publiziert unter URL http://www.informatik.uni-frankfurt.de/fbreports/fbreport11-99.ps.gz*

12/1999 Kappes, M.; Klemm, R. P.; Kintala, C. M. R.:
Determining Component-based Software System Reliability is Inherently Impossible

13/1999 Kappes, Martin:
Multi-Bracketed Contextual Rewriting Grammars With Obligatory Rewriting

14/1999 Kemp, Rainer:
On the Expected Number of Leftist Nodes in Simply Generated Trees

1/2000 Kemp, Rainer:
On the Average Shape of Dynamically Growing Trees

2/2000 Arlt, B.; Brause, R.; Tratar, E.:
MASCOT: A Mechanism for Attention-based Scale-invariant Object Recognition in Images. *Elektronisch publiziert unter URL http://www.cs.uni-frankfurt.de/fbreports/fbreport2-00.pdf*

3/2000 Heuschen, Frank; Waldschmidt, Klaus:
Bewertung analoger und digitaler Schaltungen der Signalverarbeitung

4/2000 Hamker, Fred H.; Paetz, Jürgen; Thöne, Sven; Brause, Rüdiger; Hanisch, Ernst:
Erkennung kritischer Zustände von Patienten mit der Diagnose „Septischer Schock" mit einem RBF-Netz. *Elektronisch publiziert unter URL http://www.cs.uni-frankfurt.de/fbreports/fbreport04-00.pdf*

1/2001 Nebel, Markus E.:
A Unified Approach to the Analysis of Horton-Strahler Parameters of Binary Tree Structures

2/2001 Nebel, Markus E.:
Combinatorial Properties of RNA Secondary Structures

3/2001 Nebel, Markus E.:
Investigation of the Bernoulli-Model for RNA Secondary Structures

4/2001 Malcher, Andreas:
Descriptional Complexity of Cellular Automata and Decidability Questions

1/2002 Paetz, Jürgen:
Durchschnittsbasierte Generalisierungsregeln; Teil I: Grundlagen

2/2002 Paetz, Jürgen; Brause, Rüdiger:
Durchschnittsbasierte Generalisierungsregeln Teil II: Analyse von Daten septischer Schock-Patienten

3/2002 Nießner, Frank:
Decomposition of Deterministic ω - regular Liveness Properties and Reduction of Corresponding Automata

4/2002 Kim, Pok-Son:
Das $\mathcal{RSV}$-Problem ist $\mathcal{NP}$-vollständig

5/2002 Nebel, Markus E.:
On a Statistical Filter for RNA Secondary Structures

6/2002 Malcher, Andreas:
Minimizing Finite Automata is Computationally Hard

1/2003 Malcher, Andreas:
On One-Way Cellular Automata with a Fixed Number of Cells

2/2003 Malcher, Andreas:
On Two-Way Communication in Cellular Automata with a Fixed Number of Cells

3/2003 Malcher, Andreas:
On the Descriptional Complexity of Iterative Arrays

4/2003 Kemp, Rainer:
On the Expected Number of Leftist Nodes in Dynamically Growing Trees

5/2003 Nebel, Markus E.:
Identifying Good Predictions of RNA Secondary Structure

1/2004 Meise, Christian:
Zwischenbericht zum Projekt BeCom

2/2004 Malcher, Andreas:
On Non-Recursive Trade-Offs Between Finite-Turn Pushdown Automata