On span-$P^{cc}$ and related classes in structural communication complexity

Henning Wunderlich

Ulmer Informatik-Berichte
Nr. 2008-10
Juli 2008
On span-$P^{cc}$ and related classes in structural communication complexity

Henning Wunderlich *

July 8, 2008

Abstract
The complexity classes $\#P$, $\#NP$, $\text{min}-P$, $\text{max}-P$, $\text{opt}-P$ and $\text{span}-P$ are well known in structural complexity. We define analogous classes in structural communication complexity and study some of their properties, e.g. establishing the inclusions $\#P^{cc} \subseteq \text{span}-P^{cc} \subseteq \#NP^{cc}$ and $\text{max}-P^{cc} \subseteq \text{span}-P^{cc}$. Especially, in contrast to the current state of affairs in time complexity, we are able to prove the following separations:

1. $\#P^{cc} \subset \text{span}-P^{cc} \subset \#NP^{cc}$
2. $\text{max}-P^{cc} \subset \#P^{cc}$, $\text{max}-P^{cc} \subset \text{span}-P^{cc}$
3. $\text{min}-P^{cc} \neq \text{max}-P^{cc}$, $\text{min}-P^{cc} \subset \text{span}-P^{cc}$

1 Introduction

In structural complexity theory various natural function classes have been defined by considering certain operators acting over the computation tree of a nondeterministic polynomial time Turing machine (NPTM). Valiant’s classes $\#P$ and $\#NP$ [10] are defined as classes of functions that count the number of accepting paths of the computation tree of an NPTM which in the latter case has access to an $NP$-oracle. Krentel [7] studied optimization problems and defined the class $\text{opt}-P = \text{min}-P \cup \text{max}-P$ containing functions computing the minimum (min-$P$) or maximum (max-$P$), respectively, of the output values occurring at the leaves of computation trees of NPTMs. Motivated by the study of the graph nonisomorphism problem, Köbler, Schöning and Torán [6] introduced the class $\text{span}-P$ of span functions counting the number of different output values of an NPTM. Some of their findings were the inclusions $\#P \subseteq \text{span}-P \subseteq \#NP$, $\text{max}-P \subseteq \text{span}-P$ and several equivalences relating language classes to function classes:

\begin{align*}
\#P = \text{span}-P & \iff \text{NP} = \text{UP} \quad (1) \\
\#P = \text{max}-P & \iff \text{NP} = \text{PP} \quad (2) \\
\#NP = \text{span}-P & \iff \text{NP} = \text{coNP} \quad (3) \\
\text{max}-P = \text{span}-P & \iff \text{NP} = \text{coNP} \quad (4)
\end{align*}

*Universität Ulm, Fakultät für Ingenieurwissenschaften und Informatik, Institut für Theoretische Informatik, Oberer Eselsberg, D-89069 Ulm, e-mail: Henning.Wunderlich@uni-ulm.de
By analogous proof methods we show the \( \Rightarrow \) implications for the corresponding communication complexity classes. As the right hand sides do not hold in the setting of communication complexity, this yields separations for the function classes \( \#P^{cc}, \text{span-} P^{cc}, \#NP^{cc}, \text{opt-} P^{cc}, \text{min-} P^{cc} \) and \( \text{max-} P^{cc} \).

We consider the basic model of communication complexity, introduced by Yao [11]. In this model, there are two players (parties) Alice and Bob, who want to cooperatively compute a function \( f : \mathcal{X} \times \mathcal{Y} \rightarrow \mathcal{Z} \), where \( \mathcal{X}, \mathcal{Y} \) and \( \mathcal{Z} \) are finite sets. Both have complete information about \( f \) and unlimited computational power but receive only parts of the inputs. Alice is given \( x \in \mathcal{X} \), Bob is given \( y \), and they exchange messages in order to compute \( f(x, y) \). The communication is carried out according to a fixed protocol \( \Pi \) (over domain \( \mathcal{X} \times \mathcal{Y} \) with range \( \mathcal{Z} \)), which is a labeled binary tree. An inner node specifies the player who sends a bit of communication next. For a deterministic protocol, this bit solely depends on the player’s input and the bits communicated so far. For a nondeterministic protocol, it can also depend on the player’s guess string. Each leaf \( l \) is labeled with an output value \( z_l \in \mathcal{Z} \). On inputs \( x, y \) we denote the transcript, i.e. the sequence of the bits communicated, by \( \Pi(x, y) \). The output of the protocol is defined as the label associated with the leaf reached by the execution of the protocol. The set \( R_v \) of inputs going through a node \( v \) (including the case of leaves) of a protocol forms a (combinatorial) rectangle, i.e. \( R_v = A \times B, A \subseteq \mathcal{X}, B \subseteq \mathcal{Y} \). A rectangle \( R \) is \( z \)-chromatic for \( f \) if \( f^{-1}(R) = \{ z \} \), and monochromatic, if there exists a \( z \)-value such that \( R \) is \( z \)-chromatic for \( f \). The communication matrix \( M^f \) of \( f \) is the \( \mathcal{X} \times \mathcal{Y} \)-matrix \( (f(x, y))_{x \in \mathcal{X}, y \in \mathcal{Y}} \), i.e. \( f \) written in matrix form. We denote with \( f_{ll} \) the function computed by \( \Pi \). A nondeterministic protocol computing a Boolean function \( f \) induces a cover of \( f^{-1}(1) \) with 1-chromatic rectangles. A deterministic protocol induces a disjoint cover of \( M^f \) with monochromatic rectangles. The (non)-deterministic communication complexity of \( f \) is the minimum number of bits a (non)-deterministic protocol needs to compute \( f \). For a thorough introduction to communication complexity we refer the reader to the book of Kushilevitz and Nisan [8].

Research in the field of structural communication complexity started with the article of Babai, Frankl and Simon [1], where some analogies between Turing machine classes like \( P, NP, PP, \text{PSPACE} \), the polynomial hierarchy \( \text{PH} = \bigcup_k \Sigma^p_k \), etc. and the corresponding communication complexity classes \( P^{cc}, \text{NP}^{cc}, \text{PP}^{cc}, \text{PSPACE}^{cc}, \text{PH}^{cc} = \bigcup_k \Sigma^p_k^{cc} \), etc. were shown. For more ground work, especially on closure properties, the boolean communication hierarchy, or counting communication complexity classes like \( \text{MOD}_m \text{P}^{cc} \), see Halstenberg and Reischuk [4] or Damm et al. [2]. To the best of the author’s knowledge, except for the class \( \#P^{cc} \), none of the communication complexity function classes under consideration in this paper have been defined and studied before.

2 Notation and basic definitions

We only work with the binary alphabet \( \mathbb{B} := \{0, 1\} \). The length of a string \( x \in \mathbb{B}^* \) is denoted by \( |x| \). A prefix-free encoding of \( x \in \mathbb{B}^* \) is \( \overline{x} := 0^{|x|}1x \). In order to encode pairs of strings \( x, y \in \mathbb{B}^* \) we use the pairing function \( (x, y) := \overline{x}y \). bin(\( n \)) is the binary representation of \( n \), and \((\cdot)_2 \) is its inverse. The set of pairs of strings of equal length is denoted by \( \mathbb{B}^* := \{(x, y) \mid x, y \in \mathbb{B}^*, |x| = |y|\} \). A language \( L \) is a subset of \( \mathbb{B}^* \), its characteristic function \( \chi^L \) is defined as \( \chi^L := (\chi^L_x) \).
In addition, we define

$$\chi^L_v: \mathbb{B}^n \times \mathbb{B}^n \to \mathbb{N}, \chi^L_v(x, y) := 1, \text{ if } (x, y) \in L, \text{ and } 0 \text{ otherwise. The set of all languages is denoted by } \mathcal{L}. \text{ A (communication) complexity class is a subset } \mathcal{C} \subseteq \mathcal{L}. \text{ We define } \text{poly} := \{ f: \mathbb{R}^+ \to \mathbb{R}^+ | \exists \text{polynomial } p: f \leq p \}, \text{ the set of functions with polynomial growth. Let } \mathcal{F}_n := \{ f: \mathbb{B}^n \times \mathbb{B}^n \to \mathbb{N} \}. \text{ and let } \mathcal{F} := \{ f = (f_n)_{n \in \mathbb{N}} | f_n \in \mathcal{F}_n \}. \text{ We say that the function family } f := (f_n)_{n \in \mathbb{N}} \in \mathcal{F} \text{ is bounded, if there exists a bound } b \in \text{poly} \text{ such that for all natural numbers } n \text{ we have } f_n(x, y) \leq 2^{[b(\log_2 n)]} \text{ for all inputs } x, y \in \mathbb{B}^n. \text{ A protocol over domain } \mathcal{X} \times \mathcal{Y} \text{ is an } n\text{-bit protocol, if } \mathcal{X} = \mathcal{Y} = \mathbb{B}^n. \text{ A deterministic, randomized or nondeterministic protocol } \Pi \text{ over } \mathcal{X}, \mathcal{Y} \text{ is an oracle protocol with oracle family } O \in \mathcal{F}, \text{ if } \Pi \text{ contains oracle nodes in its protocol tree. Associated with an oracle node } v \text{ are two functions } a_v: \mathcal{X} \to \mathbb{B}^{m_v} \text{ and } b_v: \mathcal{Y} \to \mathbb{B}^{m_v}. \text{ If Alice and Bob reach an oracle node } v \text{ during a computation on inputs } x \in \mathcal{X}, y \in \mathcal{Y}, \text{ they compute by themselves } x':= a_v(x) \text{ and } y':= b_v(y), \text{ respectively, and call } O \text{ on } (x', y'). \text{ The oracle node } v \text{ has exactly } |\text{range}(O)| \text{ many successors. Alice and Bob continue the computation on one of them according to the returned value } O(x', y'). \text{ The communication costs for each oracle call are } |\log_2 |\text{range}(O)||. \text{ If a language } L \text{ is used as an oracle family, we write } L \text{ instead of } \chi^L_n.\\\\3 Counting classes

Executing an NPTM on a specific input one can count the number of accepting computations or the number of different output values, etc. The same can be done for communication protocols:

**Definition 3.1** (Transducer, Acceptor). A nondeterministic protocol \( \Pi \) (with oracle family \( O \)) is a transducer, if its output nodes are marked with elements \((b, z), b \in \mathbb{B}, z \in \mathbb{B}^* \). We say that \( \Pi \) (with oracle family \( O \)) accepts inputs \((x, y), x \in \mathcal{X}, y \in \mathcal{Y} \) with output \( z \in \mathbb{B}^* \), if there exists a guess \((g_A, g_B)\) such that Alice and Bob arrive at an output node labeled with \((1, z)\) when executing \( \Pi(O) \) on inputs \((x, y)\) and guess strings \( g_A, g_B \). Otherwise, we say that \( \Pi \) (with oracle family \( O \)) rejects inputs \((x, y)\). We define its output set as

$$\text{out}^O_{\Pi}(x, y) := \{ z \in \mathbb{B}^* | \Pi(O) \text{ accepts } (x, y) \text{ with output } z \}$$

(5)

In addition, we define

$$\text{span}^O_{\Pi}(x, y) := |\text{out}^O_{\Pi}(x, y)|$$

(6)

$$\text{(out}^O_{\Pi})_2(x, y) := \{ (z)_2 | z \in \text{out}^O_{\Pi}(x, y) \}$$

(7)

$$\text{min}^O_{\Pi}(x, y) := \min(\text{out}^O_{\Pi})_2(x, y)$$

(8)

$$\text{max}^O_{\Pi}(x, y) := \max(\text{out}^O_{\Pi})_2(x, y)$$

(9)

A transducer \( \Pi \) is an acceptor, if \( \Pi \) outputs its transcript in the event of acceptance. For an acceptor \( \Pi \), we define the number of accepting transcripts as

$$\text{acc}^O_{\Pi} := \text{span}^O_{\Pi}.$$ 

(10)

Considering a single protocol does not make sense in structural communication complexity, because we are interested in the asymptotic behaviour as input size increases. Accordingly, we have to define special classes of protocol families with contraints on the resources used.
Definition 3.2 (\(\text{NP}^{cc}\)-transducer, \(\text{NP}^{cc}\)-acceptor). Let \(O = (O_m)_{m \in \mathbb{N}} \in \mathcal{F}\) be an oracle family. An \(\text{NP}^{cc}\)-transducer (with oracle family \(O\)) is a family 
\[ \Pi = (\Pi_n)_{n \in \mathbb{N}} \]
of \(n\)-bit transducers together with transcript, guess, query and output bounds \(t, g, q, o \in \text{poly}\) such that when executing \(\Pi_n\), Alice and Bob communicate \([t(\log_2 n)]\) many bits, they use guess strings of length \([g(\log_2 n)]\), they are allowed to use the oracles \(O_1, \ldots, O_{2^{g(\log_2 n)}}\), and their output has length at most \([o(\log_2 n)]\). \(\Pi\) is a \(\text{P}^{cc}\)-transducer, if each \(\Pi_n\) is a deterministic protocol, and an \(\text{NP}^{cc}\)-acceptor, if each \(\Pi_n\) is an acceptor.

We also define
\[
\begin{align*}
\text{out}_{\Pi_n}^O & := (\text{out}_{\Pi_n}^O)_{n \in \mathbb{N}} \quad (11) \\
\text{span}_{\Pi_n}^O & := (\text{span}_{\Pi_n}^O)_{n \in \mathbb{N}} \quad (12) \\
(\text{out}_{\Pi_n}^O)_2 & := ((\text{out}_{\Pi_n}^O)_2)_{n \in \mathbb{N}} \quad (13) \\
\text{min}_{\Pi_n}^O & := (\text{min}_{\Pi_n}^O)_{n \in \mathbb{N}} \quad (14) \\
\text{max}_{\Pi_n}^O & := (\text{max}_{\Pi_n}^O)_{n \in \mathbb{N}} \quad (15)
\end{align*}
\]
and in case of an \(\text{NP}^{cc}\)-acceptor
\[
\text{acc}_{\Pi_n}^O := \text{span}_{\Pi_n}^O. \quad (16)
\]

Definition 3.3. For a complexity class \(\mathcal{C}\) define
\[
\begin{align*}
\mathcal{FC} & := \{ f_{\Pi(O)} \mid \Pi \text{ is a } \text{P}^{cc}\text{-transducer with oracle family } O \in \mathcal{C} \} \quad (17) \\
\#\text{P}^{cc} & := \{ \text{acc}_{\Pi} \mid \Pi \text{ is an } \text{NP}^{cc}\text{-acceptor} \} \quad (18) \\
\#\mathcal{C} & := \{ \text{acc}_{\Pi}^O \mid \Pi \text{ is an } \text{NP}^{cc}\text{-acceptor with oracle family } O \in \mathcal{C} \} \quad (19) \\
\text{min-}\mathcal{C} & := \{ \text{min}_{\Pi}^O \mid \Pi \text{ is an } \text{NP}^{cc}\text{-transducer with oracle family } O \in \mathcal{C} \} \quad (20) \\
\text{max-}\mathcal{C} & := \{ \text{max}_{\Pi}^O \mid \Pi \text{ is an } \text{NP}^{cc}\text{-transducer with oracle family } O \in \mathcal{C} \} \quad (21) \\
\text{span-}\mathcal{C} & := \{ \text{span}_{\Pi}^O \mid \Pi \text{ is an } \text{NP}^{cc}\text{-transducer with oracle family } O \in \mathcal{C} \} \quad (22)
\end{align*}
\]

For the time complexity class \(\text{NP}\) one can give a characterization via witnesses and polynomial time predicates for its languages. An analogous statement also holds for \(\text{NP}^{cc}\).

Fact 3.4. The following statements hold:

1. A language \(L\) is in \(\text{P}^{cc}\) iff there exists a \(\text{P}^{cc}\)-acceptor for \(L\).
2. A language \(L\) is in \(\text{NP}^{cc}\) iff there exists an \(\text{NP}^{cc}\)-acceptor for \(L\).
3. A language \(L\) is in \(\text{NP}^{cc}\) iff there exists a language \(L'\) in \(\text{P}^{cc}\) and a \(p \in \text{poly}\) such that for all \((x, y) \in \mathcal{B}^*\), \(n := |x| = |y|\),
\[
(x, y) \in L \iff \exists (w_A, w_B) : |w_A|, |w_B| = [p(\log_2 n)], \quad (x, w_A), (y, w_B) \in L'.
\]

Lemma 3.5. Let \(f = \text{acc}_{\Pi}^O\) for an \(\text{NP}^{cc}\)-acceptor \(\Pi = (\Pi_n)_{n \in \mathbb{N}}\) and an oracle \(O\) in \(\text{NP}^{cc}\), then \(f = \text{acc}_{\Pi'}^O\) for an oracle \(O'\) in \(\text{NP}^{cc}\) and an \(\text{NP}^{cc}\)-acceptor \(\Pi'\) that for every input and every possible transcript asks at most one question to the oracle.
Proof. Let $f = \text{acc}_O^\Pi$ with $O$ in $\text{NP}^{cc}$. There is a language $Q$ in $\text{P}^{cc}$ and a bound $p_Q \in \text{poly}$ such that for all $(x, y) \in \mathbb{B}^*$, $(x, y) \in O$ iff $\exists (w_A, w_B)$, $|w_A|, |w_B| = \lceil p_Q(\log_2 |x|) \rceil$ and $(x, w_A), (y, w_B) \in Q$. Let $g, f, q \in \text{poly}$ be the guess, transcript and query bounds of $\Pi$. Consider the following $\text{NP}^{cc}$-acceptor $\Pi' = (\Pi'_n)_{n \in \mathbb{N}}$, where each protocol on $n$-bit inputs $(x, y)$ does the following: Alice and Bob privately guess $\lceil g(\log_2 n) \rceil$ bits $g_A$ and $g_B$, respectively, and simulate $\Pi_n$ on $(x, g_A, (y, g_B))$. Each time they reach an oracle node $v_i, i \in [m]$, $m \leq |f(\log_2 n)|$, instead of calling an oracle, Alice guesses the answer $z_i \in \mathbb{B}$ and sends it to Bob. If $\Pi_n$ rejects, they reject. If $\Pi_n$ accepts, for each with $z_i = 1$ Alice and Bob privately guess $w_A^i, w_B^i$ of length $|p_O(\log_2 n)|$ and check $((x, w_A^i), (y, w_B^i)) \in Q$. If one of the checks fails, they reject. Otherwise, they call oracle $O'$ on input $(((a_n(x), w_A^i | z_i = 1), (a_n(x) | z_i = 0)), ((b_n(y), w_B^i | z_i = 1), (b_n(y) | z_i = 0))$. Alice and Bob accept iff $O'$ rejects. The $\text{NP}^{cc}$-language $O'$ contains all pairs $((p_A^i, w_A^i | i \in [m_1]), (q_A^i | i \in [m_2]))$, $((p_B^i, w_B^i | i \in [m_1]), (q_B^i | i \in [m_2]))$ such that there exists an $n$ with $m_1 + m_2 \leq \lceil f(\log_2 n) \rceil$, $|p_A^i|, |p_B^i|, |q_A^i|, |q_B^i| \leq 2^{|f(\log_2 n)|}$, $|p_A^i| = |p_B^i|$, $|q_A^i| = |q_B^i|$, $|w_A^i|, |w_B^i| = \lceil p_O(\log_2 p_A^i) \rceil$, and $((\exists i \in [m_1] : (q_A^i, q_B^i) \in O) \lor (\exists i \in [m_2] : (q_A^i, q_B^i) \in O)) \Rightarrow A. r_B : r_A = |r_A| = |w_A^i|, (r_A r_B) < (w_A^i w_B^i)_{2} < (p_A^i, r_A, r_B, p_B^i) \in Q)$. □

Definition 3.6. Let $f \in \mathcal{F}_n$ be a function. We define the $\leq$ and $\geq$-graph of $f$ as

\[
\begin{align*}
\text{Graph}_{\leq}(f) & := \{(x, \text{bin}(z)), (y, \text{bin}(z)) \mid x, y \in \mathbb{B}^n, z \leq f(x, y)\} \\
\text{Graph}_{\geq}(f) & := \{(x, \text{bin}(z)), (y, \text{bin}(z)) \mid x, y \in \mathbb{B}^n, z \geq f(x, y)\}
\end{align*}
\]

Let $f = (f_n)_{n \in \mathbb{N}} \in \mathcal{F}$ be a function family.

\[
\begin{align*}
\text{Graph}_{\leq}(f) & := (\text{Graph}_{\leq}(f_n))_{n \in \mathbb{N}} \\
\text{Graph}_{\geq}(f) & := (\text{Graph}_{\geq}(f_n))_{n \in \mathbb{N}}
\end{align*}
\]

Corollary 3.7. Let $f \in \mathcal{F}$ be bounded. Then $f \in \text{FP}^{cc}(\text{Graph}_{\leq}(f))$.

Proof. As there exists a bound $b \in \text{poly}$ with $f(x, y) \leq 2^{|b(\log_2 n)|}$ for every $x, y \in \mathbb{B}^n$, Alice and Bob can determine the value $f(x, y)$ simply by binary search using $|b(\log_2 n)|$ many oracle calls. □

Lemma 3.8. For bounded $f \in \mathcal{F}$ it holds

1. $\text{Graph}_{\leq}(f) \in \text{NP}^{cc} \iff \text{Graph}_{\geq}(f) \in \text{co-NP}^{cc}$,
2. $f \in \text{max-P}^{cc} \iff \text{Graph}_{\leq}(f) \in \text{NP}^{cc}$,
3. $f \in \text{min-P}^{cc} \iff \text{Graph}_{\geq}(f) \in \text{NP}^{cc}$.

Proof. 1. Clearly, $z \leq f(x, y) \iff \lnot(z - 1 \geq f(x, y))$.

2. (⇒) Let $f := \text{max}_f$ for an $\text{NP}^{cc}$-transducer $\Pi = (\Pi_n)_{n \in \mathbb{N}}$. Then for each $n$ we have $\text{Graph}_{\leq}(f_n) = \{(x, \text{bin}(z)), (y, \text{bin}(z)) \mid x, y \in \mathbb{B}^n, \text{there exists an accepting transcript } \Pi_n(x, y) \text{ with output } u \geq z\}$. This implies $\text{Graph}_{\leq}(f) \in \text{NP}^{cc}$.

(⇐) Let $b \in \text{poly}$ be a bound for $f$. We construct an $\text{NP}^{cc}$-transducer $\Pi = (\Pi_n)_{n \in \mathbb{N}}$ such that $f = \text{max}_f$. In $\Pi_n$ on inputs $x, y \in \mathbb{B}^n$ Alice guesses a number $z \leq 2^{|b(\log_2 n)|}$ and sends it to Bob. They output $z$, if the verification of $(x, \text{bin}(z)), (y, \text{bin}(z)) \in \text{Graph}_{\leq}(f_n)$ succeeds.

5
3. Immediate consequence of (i) and (ii).

Now, we have the tools at hand to show an analog of a theorem of Krentel proved in [7]. The classes $\Delta^c_n := P^{cc}(\Sigma^c_n)$ are related to the polynomial hierarchy.

**Theorem 3.9.** $\min-\mathbf{P}^{cc}, \max-\mathbf{P}^{cc} \subseteq F\Delta^c_n$.

**Proof.** Follows from Lemma 3.8 and Corollary 3.7.

**Definition 3.10.** For a pair of $n$-bit transducers $\Pi, \Pi'$ define the function $\text{span}_{\Pi, \Pi'}$ such that $\text{span}_{\Pi, \Pi'}(x, y)$ is the number of different outputs that $\Pi$ on inputs $(x, y)$ can produce that cannot be produced by $\Pi'$.

For a pair of $\mathbf{NP}^{cc}$-transducers $\Pi = (\Pi_n)_{n \in \mathbb{N}}, \Pi' = (\Pi'_n)_{n \in \mathbb{N}}$ define

$$\text{span}_{\Pi, \Pi'} := (\text{span}_{\Pi_n, \Pi'_n})_{n \in \mathbb{N}}.$$  \hfill (28)

**Proposition 3.11 (\#NP^{cc}-characterization).** $\#\mathbf{NP}^{cc} = \{ f \mid f = \text{span}_{\Pi, \Pi'} \text{ for some pair of } \mathbf{NP}^{cc}\text{-transducers } \Pi, \Pi' \}$.

**Proof.** For the forward inclusion let $f \in \#\mathbf{NP}^{cc}$. By Lemma 3.5 there is an $\mathbf{NP}^{cc}$-transducer $\Pi = (\Pi_n)_{n \in \mathbb{N}}$ and an oracle $O$ in $\mathbf{NP}^{cc}$ such that $f = \text{acc}^O_\Pi$, and for every $n$ and every transcript the transducer $\Pi_n$ asks at most one question to the oracle. Let $Q$ be a language in $\mathbf{P}^{cc}$ and let $q \in \text{poly}$ such that for all $(x, y)$, $|x| = |y| = n$ it holds: $(x, y) \in O$ if $\exists(w_A, w_B): |w_A|, |w_B| = [q(\log_2 n)]$ and $((x, w_A), (y, w_B)) \in Q$. Consider the following $\mathbf{NP}^{cc}$-transducers $\Pi' = (\Pi'_n)_{n \in \mathbb{N}}$ and $\Pi'' = (\Pi''_n)_{n \in \mathbb{N}}$, respectively:

- $\Pi'_n$: On $n$ bit inputs $(x, y)$ Alice and Bob simulate $\Pi_n$. If $\Pi_n$ asks the question $(q_A, q_B) \in \mathbb{B}^*$ to the oracle, then Alice and Bob guess $(w_A, w_B)$, $|w_A|, |w_B| = [q(\log_2 |q_A|)]$. If $((q_A, w_A), (q_B, w_B)) \in Q$, they continue with answer 1, else with answer 0. If $\Pi_n$ accepts $(x, y)$ with transcript $t$, Alice and Bob accept with output $t$ and reject, otherwise.

- $\Pi''_n$: The protocol begins exactly as $\Pi'_n$ until Alice and Bob get the oracle answer. If $((q_A, w_A), (q_B, w_B)) \in Q$, they continue with answer 1, else reject. If $\Pi_n$ rejects $(x, y)$ with transcript $t$, Alice and Bob accept with output $t$ and reject, otherwise.

It follows that for all $(x, y)$, $|x| = |y| = n$, $f(x, y) = \text{span}_{\Pi_n, \Pi''_n}(x, y)$, since $\Pi'_n$ on inputs $(x, y)$ will output a different value for every accepting transcript of $\Pi_n$, and every output value in $\Pi''_n$ corresponding to a simulation of $\Pi_n$ in which the oracle question was wrongly guessed, will also be in the span of $\Pi''_n$ and therefore, it will not be counted.

For the backward inclusion, let $f = \text{span}_{\Pi, \Pi'}$ with $\mathbf{NP}^{cc}$-transducers $\Pi^0 = (\Pi^0_n)_{n \in \mathbb{N}}, \Pi^1 = (\Pi^1_n)_{n \in \mathbb{N}}$. The language $O$ contains all pairs $((x, b, z), (y, b, 0|z))$ such that $|x| = |y| = n$ and $z$ is an output value of $\Pi^b_n$. Clearly, $O \in \mathbf{NP}^{cc}$. Let $r_b \in \text{poly}$ be the output bound of $\Pi^b_n, b \in \{0, 1\}$. Consider the following $\mathbf{NP}^{cc}$-acceptor $\Pi = (\Pi_n)_{n \in \mathbb{N}}$: When executing $\Pi_n$ on $n$ bit inputs $(x, y)$, Alice guesses a string $z$ of length $\leq \max\{r_0(\log_2 n), r_1(\log_2 n)\}$ and sends $|z|$ to Bob. They accept iff $(x, 1, z), (x, 0, 0|z)) \in O$ and $(x, 0, z), (x, 0, 0|z)) \notin O$.

**Corollary 3.12.** If $f \in \#\mathbf{NP}^{cc}$, then there exist two functions $g_1, g_2 \in \min-\mathbf{P}^{cc}$ such that for every $(x, y) \in \mathbb{B}^*$, $f(x, y) = g_1(x, y) - g_2(x, y)$.  \hfill $\square$
Proof. Let \( f \in \#\text{NP}^{cc} \). By Proposition 3.11, there is a pair of \( \text{NP}^{cc} \)-transducers \( \Pi = (\Pi_n)_{n \in \mathbb{N}} \), \( \Pi' = (\Pi'_n)_{n \in \mathbb{N}} \) such that \( f = \text{span}_{\Pi - \Pi'} \). Define the \( \text{NP}^{cc} \)-transducer \( \Pi'' = (\Pi''_n)_{n \in \mathbb{N}} \) as follows: When executing \( \Pi''_n \) on inputs \((x, y)\), Alice and Bob simulate \( \Pi_n \) and \( \Pi'_n \) on \((x, y)\). They accept with output \( z \) if both \( \Pi_n \) and \( \Pi'_n \) accept with output \( z \). Define \( g_1 := \text{span}_\Pi \) and \( g_2 := \text{span}_{\Pi'} \). It follows that \( f = g_1 - g_2 \). \( \square \)

Theorem 3.13 (Inclusions). It holds:

1. \( \text{FP}^{cc} \subseteq \min -\text{P}^{cc}, \max -\text{P}^{cc}, \text{opt} -\text{P}^{cc}, \#\text{P}^{cc} \subseteq \#\text{NP}^{cc} \)
2. \( \#\text{P}^{cc} \subseteq \text{span} -\text{P}^{cc} \subseteq \#\text{NP}^{cc} \)
3. \( \text{max} -\text{P}^{cc} \subseteq \text{span} -\text{P}^{cc} \)

Proof. 1. The inclusion \( \text{FP}^{cc} \subseteq \min -\text{P}^{cc}, \max -\text{P}^{cc}, \text{opt} -\text{P}^{cc}, \#\text{P}^{cc} \) is an immediate consequence of the definitions in Def. 3.3. As \( \text{FP}^{cc} \subseteq \#\text{P}^{cc} \) relativizes, \( \text{F}\Delta_2^{cc} \subseteq \#\text{NP}^{cc} \) follows. The inclusion \( \min -\text{P}^{cc}, \max -\text{P}^{cc} \subseteq \text{F}\Delta_2^{cc} \) was shown in Theorem 3.9.

2. Again, the first inequality follows directly from the definitions, as every \( \text{NP}^{cc} \)-acceptor is an \( \text{NP}^{cc} \)-transducer. For the second inequality, let \( f = \text{span}_\Pi \) for some \( \text{NP}^{cc} \)-transducer \( \Pi \). Obviously, \( f = \text{span}_{\Pi - \Pi'} \), where \( \Pi' \) is an \( \text{NP}^{cc} \)-transducer rejecting every input.

3. Let \( f = \text{max}_\Pi \) for some \( \text{NP}^{cc} \)-transducer \( \Pi = (\Pi_n)_{n \in \mathbb{N}} \). Let \( o \in \text{poly} \) be the output bound of \( \Pi \). We can construct a new transducer \( \Pi' = (\Pi'_n)_{n \in \mathbb{N}} \) such that for \( \Pi'_n \) Alice and Bob on inputs \((x, y)\) simulate \( \Pi_n \) and for every output \( z \) Alice guesses a positive integer \( z' \leq z \), sends it to Bob using \([o(n)]\) many bits, and both accept with output \( z' \). \( \Pi'_n \) will have as many different output values as the maximum of the output values of \( \Pi_n \). This proves \( f = \text{span}_{\Pi'} \). \( \square \)

We need to define the analog of the unambiguous nondeterministic polynomial time class UP defined by Valiant [9].

Definition 3.14. A UP\(^{cc}\)-acceptor is an \( \text{NP}^{cc} \)-acceptor \( \Pi \) with \( \text{acc}_\Pi \leq 1 \). UP\(^{cc}\) is the class of all languages recognized by UP\(^{cc}\)-acceptors.

While the separation of the time class \text{P} and UP is equivalent to the existence of certain kinds of one-way functions [3, 5], separating the communication classes P\(^{cc}\) from UP\(^{cc}\) would disprove the famous log rank conjecture (see [8, Open Problem 2.20, p.26]). Thus, separating P\(^{cc}\) from UP\(^{cc}\) seems to be hard. In contrast, separating UP\(^{cc}\) from NP\(^{cc}\) is easy. Luckily, only the latter is needed in the sequel.

For a function \( f : \mathbb{B}^n \times \mathbb{B}^n \rightarrow \mathbb{B} \) the measure \( C^{D,1}(f) \) denotes the number of 1-chromatic rectangles needed to partition \( f^{-1}(1) \) (the protocol partition number for the ones in the communication matrix \( M^f \)). With \( D(f) \) we denote the deterministic communication complexity of \( f \).

Proposition 3.15. It holds:

1. \( \text{UP}^{cc} = \{ L \in \mathcal{L} \mid \exists p \in \text{poly}: \log_2 C^{D,1}(\chi^L_m) \leq \lceil p(\log_2 n) \rceil \} \)
2. UP$^{cc} \subset \text{NP}^{cc}$

3. The log rank conjecture implies P$^{cc} = \text{UP}^{cc}$.

Proof. 1. Let $\Pi$ be a nondeterministic protocol for a function $f$ such that acc$_{NP} \leq 1$. Then the protocol induced cover of $f^{-1}(1)$ with 1-chromatic rectangles is actually a disjoint cover, i.e. a partition of $f^{-1}(1)$. Thus, the UP$^{cc}$-complexity of a function $f$ is just log$_2 C_{D,1}^{D,1}(f)$.

2. The inclusion is trivial; the separation is witnessed by the nonequality function $\text{NE} = (\text{NE}_n)_{n \in \mathbb{N}}$ (see [8, Example 2.17, p.24]). On the one hand, the nondeterministic communication complexity of NE is logarithmic [8, Example 2.5, p.19]. On the other hand, the rank lower bound (rank $M^f \leq C_{D,1}^{D,1}(f)$, [8, Lemma 1.28, p.13; Example 1.29, p.14]) yields a linear lower bound for log$_2 C_{D,1}^{D,1}(\text{NE}_n)$.

3. If the log rank conjecture holds, then for every function family $f = (f_n)_{n \in \mathbb{N}}$ of Boolean functions $f_n : \mathbb{B}^n \times \mathbb{B}^n \rightarrow \mathbb{B}$ there exists a bound $q \in \text{poly}$ such that $D(f_n) \leq q(\log_2 \text{rank } M^f_n)$. In addition, $\text{rank } M^f_n \leq C_{D,1}^{D,1}(f_n)$. Thus, if log$_2 C_{D,1}^{D,1}(f_n)$ is polynomially bounded, then $D(f_n)$ is too. □

Theorem 3.16 (Separations). It holds:

1. $\#P^{cc} \subset \text{span-P}^{cc}$
2. $\text{span-P}^{cc} \subset \#\text{NP}^{cc}$
3. min-$P^{cc} \neq \text{max-P}^{cc}$
4. min-$P^{cc}$ is in contradiction to Proposition 3.15: Suppose $\#P^{cc} = \text{span-P}^{cc}$, and let $L$ be a language in $\text{NP}^{cc}$: Let $\Pi = (\Pi_n)_{n \in \mathbb{N}}$ be an $\text{NP}^{cc}$-acceptor for $L$. Define an $\text{NP}^{cc}$-transducer $\Pi' = (\Pi'_n)_{n \in \mathbb{N}}$ such that $\Pi'_n$ outputs 1 if $\Pi_n$ accepts, and nothing otherwise. Then span$_{\Pi'}$ is the characteristic function of $L$, which is in $\#P^{cc}$ by the assumption. That is, there exists an $\text{NP}^{cc}$-acceptor $\Pi''$ with acc$_{\Pi''} = \text{span}_{\Pi'} \leq 1$. Thus, $L \in \text{UP}^{cc}$.

2. We show that span-$P^{cc} = \#\text{NP}^{cc}$ implies $\text{NP}^{cc} = \text{co-NP}^{cc}$, a contradiction: Let $L$ be a language in $\text{NP}^{cc}$ and $\Pi = (\Pi_n)_{n \in \mathbb{N}}$ an $\text{NP}^{cc}$-acceptor for $L$. Let $\Pi_1 = (\Pi'_n)_{n \in \mathbb{N}}$ be an $\text{NP}^{cc}$-transducer such that $\Pi'_1$ outputs 1 on every input, and $\Pi_2 = (\Pi'_n)_{n \in \mathbb{N}}$ an $\text{NP}^{cc}$-transducer such that $\Pi'_2$ simulates $\Pi_1$ and outputs 1 in every accepting transcript of $\Pi$. Define the function $f := (f_n)_{n \in \mathbb{N}} := \text{span}_{\Pi_1 \cdots \Pi_2}$. It follows that $f_n(x, y) > 0$ iff $(x, y) \notin L$. By Proposition 3.11 we have $f \in \#\text{NP}^{cc}$. By the hypothesis, $f$ is the span function of some $\text{NP}^{cc}$-transducer $\Pi^3 = (\Pi'_n)_{n \in \mathbb{N}}$. Consider the $\text{NP}^{cc}$-acceptor $\Pi^4 = (\Pi''_n)_{n \in \mathbb{N}}$ such that $\Pi''_n$ accepts iff $\Pi'_n$ accepts (with some output). $\Pi^4$ witnesses that $\overline{Z} \in \text{NP}$, and it follows $\text{NP}^{cc} = \text{co-NP}^{cc}$.
3. Assume $\min - P^{cc} = \max - P$. For every language $L$ in $\co - \NP^{cc}$ we have $\chi^L \in \min - P^{cc}$ as $\text{Graph}_L(\chi^L)$ is in $\co - \NP^{cc}$. But $\chi^L \in \max - P$ implies $L \in \NP^{cc}$, a contradiction.

4. Assume $\min - P^{cc} \subseteq \text{span} - P^{cc}$. For every language $L$ in $\co - \NP^{cc}$ we have $\chi^L \in \min - P^{cc}$. But $\chi^L \in \text{span} - P$ implies $L \in \NP^{cc}$, a contradiction.

5. This follows from $L \in \UP^{cc} \Leftrightarrow \chi^L \in \#P^{cc}$.

We close this section with some examples of natural function classes contained in $\#P^{cc}$ and $\text{span} - P^{cc}$.

**Example 3.17.** Consider the following families of functions.

1. $f := (f_n)_{n \in \mathbb{N}} \in \mathcal{F}$, where $f_n$ is defined as

   $$f_n(A, B) := |A \text{ op } B|$$

   for subsets $A, B \subseteq [n]$ and a set operation like $\text{op} \in \{\cup, \cap, -\}$. It is easy to see that $f$ is in $\#P^{cc}$.

2. $g := (g_n)_{n \in \mathbb{N}} \in \mathcal{F}$, where $g_n$ is defined as

   $$g_n(A, B) := |A \text{ op } B|$$

   for subsets $A, B \subseteq \mathbb{Z}_n$ and an arithmetic operation $\text{op} \in \{+, *, -\}$, where $A \text{ op } B := \{a \text{ op } b | a \in A, b \in B\}$. Clearly, $f$ is in $\text{span} - P^{cc}$.

4 Conclusion and open problems

We defined various function classes in communication complexity motivated by existing well known classes in structural complexity theory and established several separation results. In structural complexity theory also higher level versions of $\#P$, $\text{span} - P$, $\#NP$ via the levels $\Sigma^P_k$, $\Pi^P_k$ of the polynomial hierarchy are considered, i.e. $\#\Sigma^P_k$, $\text{span} - \Sigma^P_k$, $\#\Pi^P_k$. The corresponding function communication classes are $\#\Sigma^{cc}_k$, $\text{span} - \Sigma^{cc}_k$, $\#\Pi^{cc}_k$. It would be interesting to prove separations for these classes, if possible, and to present natural function families demonstrating their computational strength.

Acknowledgement

I would like to express my deep gratitude to Martin Dietzfelbinger, Jacobo Torán and Uwe Schöning for careful reading, fruitful discussions, and their support.

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. Gladitz, 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 Simulataneous 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-Classes

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 Ullrich 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
V. Arvind, J. Köbler, R. Schuler
On Helping and Interactive Proof Systems

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

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

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

Construction and Deduction Methods for the Formal Development of Software

Axel Dold
Formalisierung schematischer Algorithmen

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

Rainer Schuler
On Average Polynomial Time

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

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

Alfred Lupper
Namensverwaltung und Adressierung in Distributed Shared Memory-Systemen

Robert Regn
Verteilte Unix-Betriebssysteme

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

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

Oleg Verbitsky
On the Largest Common Subgraph Problem

Uwe Schöning
Complexity of Presburger Arithmetic with Fixed Quantifier Dimension

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

Rainer Schuler, Tomoyuki Yamakami
Structural Average Case Complexity

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 Assistenzsysteme in klinischen Anwendungsumgebungen

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

95-09  Hans-Jörg Burtschick, 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  Ercüment 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 TkGofer

96-04  Thomas Beuter, Peter Dadam
Anwendungsspezifische Anforderungen an Workflow-Mangement-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 Thierauf
Satisifiability 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

97-06  Gerhard Partsch
Punkt-zu-Punkt- und Mehrpunkt-basierende LAN-Integrationsstrategien für den
digitalen Mobilfunkstandard DECT

97-07  Manfred Reichert, Peter Dadam
ADEPTflex* - Supporting Dynamic Changes of Workflows Without Loosing Control

97-08  Hans Braxmeier, Dietmar Ernst, Andrea Mößle, Heiko Vogler
The Project NoName - A functional programming language with its development
environment

97-09  Christian Heinlein
Grundlagen von Interaktionsausdrücken

97-10  Christian Heinlein
Graphische Repräsentation von Interaktionsausdrücken

97-11  Christian Heinlein
Sprachtheoretische Semantik von Interaktionsausdrücken

97-12  Gerhard Schellhorn, Wolfgang Reif
Proving Properties of Finite Enumerations: A Problem Set for Automated Theorem
Provers
97-13 Dietmar Ernst, Frank Houdek, Wolfram Schulte, Thilo Schwinn
Experimenteller Vergleich statischer und dynamischer Softwareprüfung für eingebettete Systeme

97-14 Wolfgang Reif, Gerhard Schellhorn
Theorem Proving in Large Theories

97-15 Thomas Wennekers
Asymptotik rekurrenter neuronaler Netze mit zufälligen Kopplungen

97-16 Peter Dadam, Klaus Kuhn, Manfred Reichert
Clinical Workflows - The Killer Application for Process-oriented Information Systems?

97-17 Mohammad Ali Livani, Jörg Kaiser
EDF Consensus on CAN Bus Access in Dynamic Real-Time Applications

97-18 Johannes Köbler, Rainer Schuler
Using Efficient Average-Case Algorithms to Collapse Worst-Case Complexity Classes

98-01 Daniela Damm, Lutz Claes, Friedrich W. von Henke, Alexander Seitz, Adelinde Uhrmacher, Steffen Wolf
Ein fallbasiertes System für die Interpretation von Literatur zur Knochenheilung

98-02 Thomas Bauer, Peter Dadam
Architekturen für skalierbare Workflow-Management-Systeme - Klassifikation und Analyse

98-03 Marko Luther, Martin Strecker
A guided tour through Typelab

98-04 Heiko Neumann, Luiz Pessoa
Visual Filling-in and Surface Property Reconstruction

98-05 Ercüment Canver
Formal Verification of a Coordinated Atomic Action Based Design

98-06 Andreas Küchler
On the Correspondence between Neural Folding Architectures and Tree Automata

98-07 Heiko Neumann, Thorsten Hansen, Luiz Pessoa
Interaction of ON and OFF Pathways for Visual Contrast Measurement

98-08 Thomas Wennekers
Synfire Graphs: From Spike Patterns to Automata of Spiking Neurons

98-09 Thomas Bauer, Peter Dadam
Variable Migration von Workflows in ADEPT

98-10 Heiko Neumann, Wolfgang Sepp
Recurrent V1 – V2 Interaction in Early Visual Boundary Processing

98-11 Frank Houdek, Dietmar Ernst, Thilo Schwinn
Prüfen von C–Code und Statmate/Matlab–Spezifikationen: Ein Experiment
Gerhard Schellhorn
Proving Properties of Directed Graphs: A Problem Set for Automated Theorem Provers

Gerhard Schellhorn, Wolfgang Reif
Theorems from Compiler Verification: A Problem Set for Automated Theorem Provers

Mohammad Ali Livani
SHARE: A Transparent Mechanism for Reliable Broadcast Delivery in CAN

Mohammad Ali Livani, Jörg Kaiser
Predictable Atomic Multicast in the Controller Area Network (CAN)

Susanne Boll, Wolfgang Klas, Utz Westermann
A Comparison of Multimedia Document Models Concerning Advanced Requirements

Thomas Bauer, Peter Dadam
Verteilungsmodelle für Workflow-Management-Systeme - Klassifikation und Simulation

Uwe Schöning
On the Complexity of Constraint Satisfaction

Ercument Canver
Model-Checking zur Analyse von Message Sequence Charts über Statecharts

Johannes Köbler, Wolfgang Lindner, Rainer Schuler
Derandomizing RP if Boolean Circuits are not Learnable

Utz Westermann, Wolfgang Klas
Architecture of a DataBlade Module for the Integrated Management of Multimedia Assets

Peter Dadam, Manfred Reichert

Vikraman Arvind, Johannes Köbler
Graph Isomorphism is Low for ZPP<sup>NP</sup> and other Lowness results

Thomas Bauer, Peter Dadam
Efficient Distributed Workflow Management Based on Variable Server Assignments

Thomas Bauer, Peter Dadam
Variable Serverzuordnungen und komplexe Bearbeiterzuordnungen im Workflow-Management-System ADEPT

Gregory Baratoff, Christian Toepfer, Heiko Neumann
Combined space-variant maps for optical flow based navigation

Wolfgang Gehring
Ein Rahmenwerk zur Einführung von Leistungspunktsystemen
2000-05  Susanne Boll, Christian Heinlein, Wolfgang Klas, Jochen Wandel  
Intelligent Prefetching and Buffering for Interactive Streaming of MPEG Videos

2000-06  Wolfgang Reif, Gerhard Schellhorn, Andreas Thums  
Fehlersuche in Formalen Spezifikationen

2000-07  Gerhard Schellhorn, Wolfgang Reif (eds.)  

2000-08  Thomas Bauer, Manfred Reichert, Peter Dadam  
Effiziente Durchführung von Prozessmigrationen in verteilten Workflow-Management-Systemen

2000-09  Thomas Bauer, Peter Dadam  
Vermeidung von Überlastsituationen durch Replikation von Workflow-Servern in ADEPT

2000-10  Thomas Bauer, Manfred Reichert, Peter Dadam  
Adaptives und verteiltes Workflow-Management

2000-11  Christian Heinlein  
Workflow and Process Synchronization with Interaction Expressions and Graphs

2001-01  Hubert Hug, Rainer Schuler  
DNA-based parallel computation of simple arithmetic

2001-02  Friedhelm Schwenker, Hans A. Kestler, Günther Palm  
3-D Visual Object Classification with Hierarchical Radial Basis Function Networks

2001-03  Hans A. Kestler, Friedhelm Schwenker, Günther Palm  
RBF network classification of ECGs as a potential marker for sudden cardiac death

2001-04  Christian Dietrich, Friedhelm Schwenker, Klaus Riede, Günther Palm  
Classification of Bioacoustic Time Series Utilizing Pulse Detection, Time and Frequency Features and Data Fusion

2002-01  Stefanie Rinderle, Manfred Reichert, Peter Dadam  
Effiziente Verträglichkeitsprüfung und automatische Migration von Workflow-Instanzen bei der Evolution von Workflow-Schemata

2002-02  Walter Guttmann  
Deriving an Applicative Heapsort Algorithm

2002-03  Axel Dold, Friedrich W. von Henke, Vincent Vialard, Wolfgang Goerigk  
A Mechanically Verified Compiling Specification for a Realistic Compiler

2003-01  Manfred Reichert, Stefanie Rinderle, Peter Dadam  
A Formal Framework for Workflow Type and Instance Changes Under Correctness Checks

2003-02  Stefanie Rinderle, Manfred Reichert, Peter Dadam  
Supporting Workflow Schema Evolution By Efficient Compliance Checks

2003-03  Christian Heinlein  
Safely Extending Procedure Types to Allow Nested Procedures as Values
2003-04  Stefanie Rinderle, Manfred Reichert, Peter Dadam
On Dealing With Semantically Conflicting Business Process Changes.

2003-05  Christian Heinlein
Dynamic Class Methods in Java

2003-06  Christian Heinlein
Vertical, Horizontal, and Behavioural Extensibility of Software Systems

2003-07  Christian Heinlein
Safely Extending Procedure Types to Allow Nested Procedures as Values
(Corrected Version)

2003-08  Changling Liu, Jörg Kaiser
Survey of Mobile Ad Hoc Network Routing Protocols

2004-01  Thom Frühwirth, Marc Meister (eds.)
First Workshop on Constraint Handling Rules

2004-02  Christian Heinlein
Concept and Implementation of C++, an Extension of C++ to Support User-Defined Operator Symbols and Control Structures

2004-03  Susanne Biundo, Thom Frühwirth, Günther Palm (eds.)
Poster Proceedings of the 27th Annual German Conference on Artificial Intelligence

2005-01  Armin Wolf, Thom Frühwirth, Marc Meister (eds.)
19th Workshop on (Constraint) Logic Programming

2005-02  Wolfgang Lindner (Hg.), Universität Ulm, Christopher Wolf (Hg.) KU Leuven
2. Krypto-Tag – Workshop über Kryptographie, Universität Ulm

2005-03  Walter Guttmann, Markus Maucher
Constrained Ordering

2006-01  Stefan Sarstedt
Model-Driven Development with ACTIVECHARTS, Tutorial

2006-02  Alexander Raschke, Ramin Tavakoli Kolagari
Ein experimenteller Vergleich zwischen einer plan-getriebenen und einer leichtgewichtigen Entwicklungsmethode zur Spezifikation von eingebetteten Systemen

2006-03  Jens Kohlmeyer, Alexander Raschke, Ramin Tavakoli Kolagari
Eine qualitative Untersuchung zur Produktlinien-Integration über Organisationsgrenzen hinweg

2006-04  Thorsten Liebig
Reasoning with OWL - System Support and Insights –

2008-01  H.A. Kestler, J. Messner, A. Müller, R. Schuler
On the complexity of intersecting multiple circles for graphical display
<table>
<thead>
<tr>
<th>Year</th>
<th>Author(s)</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>2008-03</td>
<td>Frank Raiser</td>
<td>Semi-Automatic Generation of CHR Solvers from Global Constraint Automata</td>
</tr>
<tr>
<td>2008-04</td>
<td>Ramin Tavakoli Kolagari, Alexander Raschke, Matthias Schneiderhan, Ian Alexander</td>
<td>Entscheidungsdokumentation bei der Entwicklung innovativer Systeme für produktlinien-basierte Entwicklungsprozesse</td>
</tr>
<tr>
<td>2008-05</td>
<td>Markus Kalb, Claudia Dittrich, Peter Dadam</td>
<td>Support of Relationships Among Moving Objects on Networks</td>
</tr>
<tr>
<td>2008-06</td>
<td>Matthias Frank, Frank Kargl, Burkhard Stiller (Hg.)</td>
<td>WMAN 2008 – KuVS Fachgespräch über Mobile Ad-hoc Netzwerke</td>
</tr>
<tr>
<td>2008-07</td>
<td>M. Maucher, U. Schöning, H.A. Kestler</td>
<td>An empirical assessment of local and population based search methods with different degrees of pseudorandomness</td>
</tr>
<tr>
<td>2008-08</td>
<td>Henning Wunderlich</td>
<td>Covers have structure</td>
</tr>
<tr>
<td>2008-09</td>
<td>Karl-Heinz Niggl, Henning Wunderlich</td>
<td>Implicit characterization of FPTIME and NC revisited</td>
</tr>
<tr>
<td>2008-10</td>
<td>Henning Wunderlich</td>
<td>On span-P\textsuperscript{ec} and related classes in structural communication complexity</td>
</tr>
</tbody>
</table>