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

Henning Wunderlich

Ulmer Informatik-Berichte

Nr. 2008-10
Juli 2008

Ulmer Informatik-Berichte
ISSN 0939-5091

Herausgeber:
Universität Ulm
Fakultät für Ingenieurwissenschaften und Informatik
89069 Ulm

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

Henning Wunderlich *

July 8, 2008

Abstract

The complexity classes $\#\mathbf{P}$, $\#\mathbf{NP}$, $\min-\mathbf{P}$, $\max-\mathbf{P}$, opt–$\mathbf{P}$ and span–$\mathbf{P}$ 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 $\#\mathbf{P}^{cc} \subseteq \text{span}–\mathbf{P}^{cc} \subseteq \#\mathbf{NP}^{cc}$ and $\max-\mathbf{P}^{cc} \subseteq \text{span}–\mathbf{P}^{cc}$. Especially, in contrast to the current state of affairs in time complexity, we are able to prove the following separations:

- $\#\mathbf{P}^{cc} \subsetneq \text{span}–\mathbf{P}^{cc} \subsetneq \#\mathbf{NP}^{cc}$
- $\max-\mathbf{P}^{cc} \not\subseteq \#\mathbf{P}^{cc}$, $\max-\mathbf{P}^{cc} \subsetneq \text{span}–\mathbf{P}^{cc}$
- $\min-\mathbf{P}^{cc} \neq \max-\mathbf{P}^{cc}$, $\min-\mathbf{P}^{cc} \not\subseteq \text{span}–\mathbf{P}^{cc}$

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 $\#\mathbf{P}$ and $\#\mathbf{NP}$ [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 $\mathbf{NP}$-oracle. Krentel [7] studied optimization problems and defined the class opt–$\mathbf{P} = \min-\mathbf{P} \cup \max-\mathbf{P}$ containing functions computing the minimum ($\min-\mathbf{P}$) or maximum ($\max-\mathbf{P}$), respectively, of the output values occuring 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 span–$\mathbf{P}$ of span functions counting the number of different output values of an NPTM. Some of their findings were the inclusions $\#\mathbf{P} \subseteq \text{span}–\mathbf{P} \subseteq \#\mathbf{NP}$, $\max-\mathbf{P} \subseteq \text{span}–\mathbf{P}$ and several equivalences relating language classes to function classes:

$$\#\mathbf{P} = \text{span}–\mathbf{P} \quad \Leftrightarrow \quad \mathbf{NP} = \mathbf{UP} \tag{1}$$

$$\#\mathbf{P} = \max-\mathbf{P} \quad \Leftrightarrow \quad \mathbf{NP} = \mathbf{PP} \tag{2}$$

$$\#\mathbf{NP} = \text{span}–\mathbf{P} \quad \Leftrightarrow \quad \mathbf{NP} = \mathbf{coNP} \tag{3}$$

$$\max-\mathbf{P} = \text{span}–\mathbf{P} \quad \Leftrightarrow \quad \mathbf{NP} = \mathbf{coNP} \tag{4}$$

*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 $\#\mathbf{P}^{cc}$, span–$\mathbf{P}^{cc}$, $\#\mathbf{NP}^{cc}$, opt–$\mathbf{P}^{cc}$, $\min$–$\mathbf{P}^{cc}$ and $\max$–$\mathbf{P}^{cc}$.

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 $f\colon \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$, where $\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{Z}$ are finite sets. Both have complete information about f and unlimited computational power but recieve 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* Π (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_Π the function computed by Π. 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 $\mathbf{P}$, $\mathbf{NP}$, $\mathbf{PP}$, $\mathbf{PSPACE}$, the polynomial hierachy $\mathbf{PH} = \bigcup_k \Sigma_k^p$, etc. and the corresponding communication complexity classes $\mathbf{P}^{cc}$, $\mathbf{NP}^{cc}$, $\mathbf{PP}^{cc}$, $\mathbf{PSPACE}^{cc}$, $\mathbf{PH}^{cc} = \bigcup_k \Sigma_k^{cc}$, etc. were shown. For more ground work, especially on closure properties, the boolean communication hierarchy, or counting communication complexity classes like $\mathrm{MOD}_m\mathbf{P}^{cc}$, see Halstenberg and Reischuk [4] or Damm et al. [2]. To the best of the author's knowledge, except for the class $\#\mathbf{P}^{cc}$, 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 is $\overline{x} := 0^{|x|}1x$. In order to encode pairs of strings $x, y \in \mathbb{B}^*$ we use the pairing function $\langle x, y\rangle := \overline{x}y$. $\mathrm{bin}(n)$ 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* χ^L is defined as $\chi^L := (\chi_n^L)$, where $\chi_n^L\colon \mathbb{B}^n \times \mathbb{B}^n \to \mathbb{N}$, $\chi_n^L(x,y) := 1$, if $(x,y) \in L$, and 0 otherwise. The *set of all languages* is denoted by $\mathcal{L}$. A *(communication) complexity class* is a subset $\mathcal{C} \subseteq \mathcal{L}$. We define $\mathbf{poly} := \{f\colon \mathbb{R}^+ \to \mathbb{R}^+ \mid \exists\text{polynomial } p\colon f \leq p\}$, the *set of functions with polynomial growth*. Let $\mathcal{F}_n := \{f \mid f\colon \mathbb{B}^n \times \mathbb{B}^n \to \mathbb{N}\}$, and let $\mathcal{F} := \{f = (f_n)_{n\in\mathbb{N}} \mid f_n \in \mathcal{F}_n\}$. We say that the function family $f := (f_n)_{n\in\mathbb{N}} \in \mathcal{F}$ is *bounded*, if there exists a bound $b \in \mathbf{poly}$ such that for all natural numbers n we have $f_n(x,y) \leq 2^{\lceil b(\log_2 n)\rceil}$ for all inputs $x, y \in \mathbb{B}^n$. A protocol over domain $\mathcal{X}\times\mathcal{Y}$ is an *n-bit* protocol, if $\mathcal{X} = \mathcal{Y} = \mathbb{B}^n$. A deterministic, randomized or nondeterministic protocol Π over $\mathcal{X}$, $\mathcal{Y}$ is an *oracle protocol* with oracle family $O \in \mathcal{F}$, if Π contains *oracle nodes* in its protocol tree. Associated with an oracle node v are two functions $a_v\colon \mathcal{X} \to \mathbb{B}^{m_v}$ and $b_v\colon \mathcal{Y} \to \mathbb{B}^{m_v}$. If Alice and Bob reach an oracle node v during a computation on inputs $x \in X$, $y \in \mathcal{Y}$, they compute by themselves $x' := a_v(x)$ and $y' := b_v(y)$, respectively, and call O on (x', y'). The oracle node v has exactly $|\mathbf{range}(O)|$ many successors. Alice and Bob continue the computation on one of them according to the returned value $O(x', y')$. The communication costs for each oracle call are $\lceil \log_2 |\mathbf{range}(O)|\rceil$. If a language L is used as an oracle family, we write L instead of χ^L.

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* Π *(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* Π *(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* Π *(with oracle family* O*) rejects inputs* (x, y). *We define its* output set *as*

$$\mathrm{out}_\Pi^O(x,y) := \{z \in \mathbb{B}^* \mid \Pi(O) \text{ accepts } (x,y) \text{ with output } z\} \tag{5}$$

In addition, we define

$$\begin{aligned}
\mathrm{span}_\Pi^O(x,y) &:= |\mathrm{out}_\Pi^O(x,y)| && (6)\\
(\mathrm{out}_\Pi^O)_2(x,y) &:= \{(z)_2 \mid z \in \mathrm{out}_\Pi^O(x,y)\} && (7)\\
\min_\Pi^O(x,y) &:= \min(\mathrm{out}_\Pi^O)_2(x,y) && (8)\\
\max_\Pi^O(x,y) &:= \max(\mathrm{out}_\Pi^O)_2(x,y) && (9)
\end{aligned}$$

A transducer Π *is an* acceptor*, if* Π *outputs its transcript in the event of acceptance. For an acceptor* Π*, we define the* number of accepting transcripts *as*

$$\mathrm{acc}_\Pi^O := \mathrm{span}_\Pi^O. \tag{10}$$

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 familes with contraints on the resources used.

Definition 3.2 ($\mathbf{NP}^{cc}$-transducer, $\mathbf{NP}^{cc}$-acceptor)**.** *Let $O = (O_m)_{m\in\mathbb{N}} \in \mathcal{F}$ be an oracle family. An* $\mathbf{NP}^{cc}$-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 $t, g, q, o \in$ **poly** such that when executing Π_n Alice and Bob communicate $\lceil t(\log_2 n)\rceil$ many bits, they use guess strings of length $\lceil g(\log_2 n)\rceil$, they are allowed to use the oracles $O_1, \ldots, O_{2^{\lceil p(\log_2 n)\rceil}}$, and their output has length at most $\lceil o(\log_2 n)\rceil$. Π is a* $\mathbf{P}^{cc}$-transducer, *if each Π_n is a deterministic protocol, and an* $\mathbf{NP}^{cc}$-acceptor, *if each Π_n is an acceptor.*
We also define

$$\begin{aligned}
\mathrm{out}_\Pi^O &:= (\mathrm{out}_{\Pi_n}^O)_{n\in\mathbb{N}} &(11)\\
\mathrm{span}_\Pi^O &:= (\mathrm{span}_{\Pi_n}^O)_{n\in\mathbb{N}} &(12)\\
(\mathrm{out}_\Pi^O)_2 &:= ((\mathrm{out}_{\Pi_n}^O)_2)_{n\in\mathbb{N}} &(13)\\
\min_\Pi^O &:= (\min_{\Pi_n}^O)_{n\in\mathbb{N}} &(14)\\
\max_\Pi^O &:= (\max_{\Pi_n}^O)_{n\in\mathbb{N}} &(15)
\end{aligned}$$

and in case of an $\mathbf{NP}^{cc}$-acceptor

$$\mathrm{acc}_\Pi^O := \mathrm{span}_\Pi^O. \qquad (16)$$

Definition 3.3. *For a complexity class $\mathcal{C}$ define*

$$\begin{aligned}
\mathbf{F}\mathcal{C} &:= \{f_{\Pi(O)} \mid \Pi \text{ is a } \mathbf{P}^{cc}\text{-transducer with oracle family } O \in \mathcal{C}\} &(17)\\
\#\mathbf{P}^{cc} &:= \{\mathrm{acc}_\Pi \mid \Pi \text{ is an } \mathbf{NP}^{cc}\text{-acceptor}\} &(18)\\
\#\mathcal{C} &:= \{\mathrm{acc}_\Pi^O \mid \Pi \text{ is an } \mathbf{NP}^{cc}\text{-acceptor with oracle family } O \in \mathcal{C}\} &(19)\\
\min\text{-}\mathcal{C} &:= \{\min_\Pi^O \mid \Pi \text{ is an } \mathbf{NP}^{cc}\text{-transducer with oracle family } O \in \mathcal{C}\} &(20)\\
\max\text{-}\mathcal{C} &:= \{\max_\Pi^O \mid \Pi \text{ is an } \mathbf{NP}^{cc}\text{-transducer with oracle family } O \in \mathcal{C}\} &(21)\\
\mathrm{span}\text{-}\mathcal{C} &:= \{\mathrm{span}_\Pi^O \mid \Pi \text{ is an } \mathbf{NP}^{cc}\text{-transducer with oracle family } O \in \mathcal{C}\} &(22)
\end{aligned}$$

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

Fact 3.4. *The following statements hold:*

1. *A language L is in $\mathbf{P}^{cc}$ iff there exists a $\mathbf{P}^{cc}$-acceptor for L.*
2. *A language L is in $\mathbf{NP}^{cc}$ iff there exists an $\mathbf{NP}^{cc}$-acceptor for L.*
3. *A language L is in $\mathbf{NP}^{cc}$ iff there exists a language L' in $\mathbf{P}^{cc}$ and a $p \in$ **poly** such that for all $(x, y) \in \mathbb{B}^{**}$, $n := |x| = |y|$,*

$$\begin{aligned}
(x, y) \in L \quad \Leftrightarrow \quad &\exists (w_A, w_B)\colon |w_A|, |w_B| = \lceil p(\log_2 n)\rceil,\\
&(\langle x, w_A\rangle, \langle y, w_B\rangle) \in L'. \qquad (23)
\end{aligned}$$

Lemma 3.5. *Let $f = \mathrm{acc}_\Pi^O$ for an $\mathbf{NP}^{cc}$-acceptor $\Pi = (\Pi_n)_{n\in\mathbb{N}}$ and an oracle O in $\mathbf{NP}^{cc}$, then $f = \mathrm{acc}_{\Pi'}^{O'}$ for an oracle O' in $\mathbf{NP}^{cc}$ and an $\mathbf{NP}^{cc}$-acceptor Π' that for every input and every possible transcript asks at most one question to the oracle.*

Proof. Let $f = \mathrm{acc}_{\Pi}^{O}$ with O in $\mathbf{NP}^{cc}$. There is a language Q in $\mathbf{P}^{cc}$ and a bound $p_O \in \mathbf{poly}$ such that for all $(x, y) \in \mathbb{B}^{**}$, $(x, y) \in O$ iff $\exists(w_A, w_B)$, $|w_A|, |w_B| = \lceil p_O(\log_2 |x|)\rceil$ and $(\langle x, w_A\rangle, \langle y, w_B\rangle) \in Q$. Let $g, t, q \in \mathbf{poly}$ be the guess, transcript and query bounds of Π. Consider the following $\mathbf{NP}^{cc}$-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 Π_n on $(\langle x, g_A\rangle, \langle y, g_B\rangle)$. Each time they reach an oracle node v_i, $i \in [m]$, $m \le \lceil t(\log_2 n)\rceil$, instead of calling an oracle, Alice guesses the answer $z_i \in \mathbb{B}$ and sends it to Bob. If Π_n rejects, they reject. If Π_n accepts, for each i with $z_i = 1$ Alice and Bob privately guess w_A^i, w_B^i of length $\lceil p_O(\log_2 n)\rceil$ and check $(\langle x, w_A^i\rangle, \langle y, w_B^i\rangle) \in Q$. If one of the checks fails, they reject. Otherwise, they call oracle O' on input $(\langle\langle a_{v_i}(x), w_A^i \mid z_i = 1\rangle, \langle a_{v_i}(x) \mid z_i = 0\rangle\rangle, \langle\langle b_{v_i}(y), w_B^i \mid z_i = 1\rangle, \langle b_{v_i}(y) \mid z_i = 0\rangle\rangle)$. Alice and Bob accept iff O' rejects.
The $\mathbf{NP}^{cc}$-language O' contains all pairs $(\langle\langle p_A^i, w_A^i \mid i \in [m_1]\rangle, \langle q_A^i \mid i \in [m_2]\rangle\rangle$, $\langle\langle p_A^i, w_A^i \mid i \in [m_1]\rangle, \langle q_A^i \mid i \in [m_2]\rangle\rangle)$ such that there exists an n with $m_1 + m_2 \le \lceil t(\log_2 n)\rceil$, $|p_A^i|, |p_B^i|, |q_A^i|, |q_B^i| \le 2^{\lceil q(\log_2 n)\rceil}$, $|p_A^i| = |p_B^i|$, $|q_A^i| = |q_B^i|$, $|w_A^i|, |w_B^i| = \lceil p_O(\log_2 |p_A^i|)\rceil$, and $((\exists i \in [m_2]\colon (q_A^i, q_B^i) \in O)$ or $(\exists i \in [m_1]\colon \exists r_A, r_B\colon |r_A| = |r_B| = |w_A^i|, (r_A r_B)_2 < (w_A^i w_B^i)_2$ and $(\langle p_A^i, r_A\rangle, \langle p_B^i, r_B\rangle) \in Q))$. □

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

$$\mathrm{Graph}_{\le}(f) := \{(\langle x, \mathrm{bin}(z)\rangle, \langle y, \mathrm{bin}(z)\rangle) \mid x, y \in \mathbb{B}^n, z \le f(x, y)\} \quad (24)$$
$$\mathrm{Graph}_{\ge}(f) := \{(\langle x, \mathrm{bin}(z)\rangle, \langle y, \mathrm{bin}(z)\rangle) \mid x, y \in \mathbb{B}^n, z \ge f(x, y)\} \quad (25)$$

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

$$\mathrm{Graph}_{\le}(f) := (\mathrm{Graph}_{\le}(f_n))_{n\in\mathbb{N}} \quad (26)$$
$$\mathrm{Graph}_{\ge}(f) := (\mathrm{Graph}_{\ge}(f_n))_{n\in\mathbb{N}} \quad (27)$$

Corollary 3.7. *Let $f \in \mathcal{F}$ be bounded. Then $f \in \mathbf{FP}^{cc}(\mathrm{Graph}_{\le}(f))$.*

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

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

1. $\mathrm{Graph}_{\le}(f) \in \mathbf{NP}^{cc} \iff \mathrm{Graph}_{\ge}(f) \in \mathbf{co}{-}\mathbf{NP}^{cc}$,
2. $f \in \max{-}\mathbf{P}^{cc} \iff \mathrm{Graph}_{\le}(f) \in \mathbf{NP}^{cc}$,
3. $f \in \min{-}\mathbf{P}^{cc} \iff \mathrm{Graph}_{\ge}(f) \in \mathbf{NP}^{cc}$.

Proof.

1. Clearly, $z \le f(x, y) \iff \neg(z - 1 \ge f(x, y))$.
2. ($\Rightarrow$) Let $f := \max_{\Pi}$ for an $\mathbf{NP}^{cc}$-transducer $\Pi = (\Pi_n)_{n\in\mathbb{N}}$. Then for each n we have $\mathrm{Graph}_{\le}(f_n) = \{(\langle x, \mathrm{bin}(z)\rangle, \langle y, \mathrm{bin}(z)\rangle) \mid x, y \in \mathbb{B}^n,$ there exists an accepting transcript $\Pi_n(x, y)$ with output $u \ge z\}$.
This implies $\mathrm{Graph}_{\le}(f) \in \mathbf{NP}^{cc}$.
($\Leftarrow$) Let $b \in \mathbf{poly}$ be a bound for f. We construct an $\mathbf{NP}^{cc}$-transducer $\Pi = (\Pi_n)_{n\in\mathbb{N}}$ such that $f = \max_{\Pi}$: In Π_n on inputs $x, y \in \mathbb{B}^n$ Alice guesses a number $z \le 2^{\lceil b(\log_2 n)\rceil}$ and sends it to Bob. They output z, if the verification of $(\langle x, \mathrm{bin}(z)\rangle, \langle y, \mathrm{bin}(z)\rangle) \in \mathrm{Graph}_{\le}(f_n)$ succeeds.

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 $\Delta_k^{cc} := \mathbf{P}^{cc}(\Sigma_k^{cc})$ are related to the polynomial hierarchy.

Theorem 3.9. $\min\text{–}\mathbf{P}^{cc}, \max\text{–}\mathbf{P}^{cc} \subseteq \mathbf{F}\Delta_2^{cc}$.

Proof. Follows from Lemma 3.8 and Corollary 3.7. □

Definition 3.10. *For a pair of n-bit transducers Π, Π' define the function* $\mathrm{span}_{\Pi-\Pi'}$ *such that* $\mathrm{span}_{\Pi-\Pi'}(x,y)$ *is the number of different outputs that* Π *on inputs (x,y) can produce that cannot be produced by* Π'.
For a pair of $\mathbf{NP}^{cc}$*-transducers* $\Pi = (\Pi_n)_{n\in\mathbb{N}}$, $\Pi' = (\Pi'_n)_{n\in\mathbb{N}}$ *define*

$$\mathrm{span}_{\Pi-\Pi'} := (\mathrm{span}_{\Pi_n-\Pi'_n})_{n\in\mathbb{N}}. \tag{28}$$

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

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 $f = \mathrm{acc}_\Pi^O$, and for every n and every transcript the transducer Π_n asks at most one question to the oracle. Let Q be a language in $\mathbf{P}^{cc}$ and let $q \in \mathbf{poly}$ such that for all (x,y), $|x| = |y| = n$ it holds: $(x,y) \in O$ iff $\exists (w_A, w_B)\colon |w_A|, |w_B| = \lceil q(\log_2 n)\rceil$ and $(\langle x, w_A\rangle, \langle y, w_B\rangle) \in Q$. Consider the following $\mathbf{NP}^{cc}$-transducers $\Pi' = (\Pi'_n)_{n\in\mathbb{N}}$ and $\Pi'' = (\Pi''_n)_{n\in\mathbb{N}}$, respectively:

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

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

It follows that for all (x,y), $|x| = |y| = n$, $f(x,y) = \mathrm{span}_{\Pi'_n-\Pi''_n}(x,y)$, since Π'_n on inputs (x,y) will output a different value for every accepting transcript of Π_n, and every output value in Π'_n corresponding to a simulation of Π_n in which the oracle question was wrongly guessed, will also be in the span of Π''_n and therefore, it will not be counted.
For the backward inclusion, let $f = \mathrm{span}_{\Pi^1-\Pi^0}$ 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 $(\langle x, b, z\rangle, \langle y, b, 0^{|z|}\rangle)$ such that $|x| = |y| = n$ and z is an output value of Π^b_n. Clearly, $O \in \mathbf{NP}^{cc}$. Let $r_b \in \mathbf{poly}$ be the output bound of Π^b, $b \in \{0,1\}$. Consider the following $\mathbf{NP}^{cc}$-acceptor $\Pi = (\Pi_n)_{n\in\mathbb{N}}$: When executing Π_n on n bit inputs (x,y), Alice guesses a string z of length $\le \lceil \max\{r_0(\log_2 n), r_1(\log_2 n)\}\rceil$ and sends $|z|$ to Bob. They accept iff $(\langle x, 1, z\rangle, \langle x, 1, 0^{|z|}\rangle) \in O$ and $(\langle x, 0, z\rangle, \langle x, 0, 0^{|z|}\rangle) \notin O$. □

Corollary 3.12. *If* $f \in \#\mathbf{NP}^{cc}$*, then there exist two functions* $g_1, g_2 \in \mathrm{span}\text{–}\mathbf{P}^{cc}$ *such that for every* $(x,y) \in \mathbb{B}^{**}$, $f(x,y) = g_1(x,y) - g_2(x,y)$.

Proof. Let $f \in \#\mathbf{NP}^{cc}$. By Proposition 3.11, there is a pair of $\mathbf{NP}^{cc}$-transducers $\Pi = (\Pi_n)_{n\in\mathbb{N}}$, $\Pi' = (\Pi'_n)_{n\in\mathbb{N}}$ such that $f = \mathrm{span}_{\Pi-\Pi'}$. Define the $\mathbf{NP}^{cc}$-transducer $\Pi'' = (\Pi''_n)_{n\in\mathbb{N}}$ as follows: When executing Π''_n on inputs (x, y), Alice and Bob simulate Π_n and Π'_n on (x, y). They accept with output z iff both Π_n and Π'_n accept with output z. Define $g_1 := \mathrm{span}_{\Pi}$ and $g_2 := \mathrm{span}_{\Pi'}$. It follows that $f = g_1 - g_2$. □

Theorem 3.13 (Inclusions)**.** *It holds:*

1. $\mathbf{FP}^{cc} \subseteq \min\text{–}\mathbf{P}^{cc}, \max\text{–}\mathbf{P}^{cc}, \mathrm{opt}\text{–}\mathbf{P}^{cc}, \#\mathbf{P}^{cc} \subseteq \#\mathbf{NP}^{cc}$
2. $\#\mathbf{P}^{cc} \subseteq \mathrm{span}\text{–}\mathbf{P}^{cc} \subseteq \#\mathbf{NP}^{cc}$
3. $\max\text{–}\mathbf{P}^{cc} \subseteq \mathrm{span}\text{–}\mathbf{P}^{cc}$

Proof.

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

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

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

□

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

Definition 3.14. *A* $\mathbf{UP}^{cc}$*-acceptor is an* $\mathbf{NP}^{cc}$*-acceptor* Π *with* $\mathrm{acc}_{\Pi} \leq 1$. $\mathbf{UP}^{cc}$ *is the class of all languages recognized by* $\mathbf{UP}^{cc}$*-acceptors.*

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

For a function $f\colon \mathbb{B}^n \times \mathbb{B}^n \to \mathbb{B}$ 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 $\mathrm{D}(f)$ we denote the *deterministic communication complexity* of f.

Proposition 3.15. *It holds:*

1. $\mathbf{UP}^{cc} = \{L \in \mathcal{L} \mid \exists p \in \mathbf{poly}\colon\ \log_2 C^{D,1}(\chi_n^L) \leq \lceil p(\log_2 n) \rceil\}$

2. $\mathbf{UP}^{cc} \subsetneq \mathbf{NP}^{cc}$

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

Proof. 1. Let Π be a nondeterministic protocol for a function f such that $\mathrm{acc}_\Pi \leq 1$. 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 $\mathbf{UP}^{cc}$-complexity of a function f is just $\log_2 C^{D,1}(f)$.

2. The inclusion is trivial; the separation is witnessed by the nonequality function $\mathrm{NE} = (\mathrm{NE}_n)_{n\in\mathbb{N}}$ (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 ($\operatorname{rank} M^f \leq C^{D,1}(f)$, [8, Lemma 1.28, p.13; Example 1.29, p.14]) yields a linear lower bound for $\log_2 C^{D,1}(\mathrm{NE}_n)$.

3. If the log rank conjecture holds, then for every function family $f = (f_n)_{n\in\mathbb{N}}$ of Boolean functions $f_n \colon \mathbb{B}^n \times \mathbb{B}^n \to \mathbb{B}$ there exists a bound $q \in \mathbf{poly}$ such that $\mathrm{D}(f_n) \leq \lceil q(\log_2 \operatorname{rank} M^{f_n})\rceil$. In addition, $\operatorname{rank} M^{f_n} \leq C^{D,1}(f_n)$. Thus, if $\log_2 C^{D,1}(f_n)$ is polynomially bounded, then $\mathrm{D}(f_n)$ is, too.

□

Theorem 3.16 (Separations). *It holds:*

1. $\#\mathbf{P}^{cc} \subsetneq \text{span–}\mathbf{P}^{cc}$

2. $\text{span–}\mathbf{P}^{cc} \subsetneq \#\mathbf{NP}^{cc}$

3. $\min\text{–}\mathbf{P}^{cc} \neq \max\text{–}\mathbf{P}^{cc}$

4. $\min\text{–}\mathbf{P}^{cc} \not\subseteq \text{span–}\mathbf{P}^{cc}$

5. $\min\text{–}\mathbf{P}^{cc}, \max\text{–}\mathbf{P}^{cc} \not\subseteq \#\mathbf{P}^{cc}$

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

2. We show that $\text{span–}\mathbf{P}^{cc} = \#\mathbf{NP}^{cc}$ implies $\mathbf{NP}^{cc} = \mathbf{co}\text{–}\mathbf{NP}^{cc}$, a contradiction: Let L be a language in $\mathbf{NP}^{cc}$ and $\Pi = (\Pi_n)_{n\in\mathbb{N}}$ an $\mathbf{NP}^{cc}$-acceptor for L. Let $\Pi^1 = (\Pi^1_n)_{n\in\mathbb{N}}$ be an $\mathbf{NP}^{cc}$-transducer such that Π^1_n outputs 1 on every input, and $\Pi^2 = (\Pi^2_n)_{n\in\mathbb{N}}$ an $\mathbf{NP}^{cc}$-transducer such that Π^2_n simulates Π_n and outputs 1 in every accepting transcript of Π_n. Define the function $f := (f_n)_{n\in\mathbb{N}} := \mathrm{span}_{\Pi^1 - \Pi^2}$. It follows that $f_n(x,y) > 0$ iff $(x,y) \notin L$. By Proposition 3.11 we have $f \in \#\mathbf{NP}^{cc}$. By the hypothesis, f is the span function of some $\mathbf{NP}^{cc}$-transducer $\Pi^3 = (\Pi^3_n)_{n\in\mathbb{N}}$. Consider the $\mathbf{NP}^{cc}$-acceptor $\Pi^4 = (\Pi^4_n)_{n\in\mathbb{N}}$ such that Π^4_n accepts iff Π^3_n accepts (with some output). Π^4 witnesses that $\overline{L} \in \mathbf{NP}$, and it follows $\mathbf{NP}^{cc} = \mathbf{co}\text{–}\mathbf{NP}^{cc}$.

3. Assume $\min-\mathbf{P}^{cc} = \max-\mathbf{P}$. For every language L in $\mathbf{co}-\mathbf{NP}^{cc}$ we have $\chi^L \in \min-\mathbf{P}^{cc}$ as $\mathrm{Graph}_{\leq}(\chi^L)$ is in $\mathbf{co}-\mathbf{NP}^{cc}$. But $\chi^L \in \max-\mathbf{P}$ implies $L \in \mathbf{NP}^{cc}$, a contradiction.

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

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

$\square$

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

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*

$$f_n(A, B) := |A \operatorname{op} B| \tag{29}$$

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

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

$$g_n(A, B) := |A \operatorname{op} B| \tag{30}$$

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

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 $\#\mathbf{P}$, span–$\mathbf{P}$, $\#\mathbf{NP}$ via the levels Σ_k^p, Π_k^p of the polynomial hierarchy are considered, i.e. $\#\Sigma_k^p$, span–Σ_k^p, $\#\Pi_k^p$. The corresponding function communication classes are $\#\Sigma_k^{cc}$, span–Σ_k^{cc}, $\#\Pi_k^{cc}$. 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

[1] L. Babai, P. Frankl, and J. Simon. Complexity classes in communication complexity theory (preliminary version). In *27th Annual Symposium on Foundations of Computer Science, 27-29 October 1986, Toronto, Ontario, Canada*, pages 337–347, 1986.

[2] C. Damm, M. Krause, C. Meinel, and S. Waack. On relations between counting communication complexity classes. *J. Comput. Syst. Sci.*, 69(2):259–280, 2004.

[3] J. Grollmann and A. L. Selman. Complexity measures for public-key cryptosystems. *SIAM J. Comput.*, 17(2):309–335, 1988.

[4] B. Halstenberg and R. Reischuk. Relations between communication complexity classes. *J. Comput. Syst. Sci.*, 41(3):402–429, 1990.

[5] K.-I. Ko. On some natural complete operators. *Theor. Comput. Sci.*, 37:1–30, 1985.

[6] J. Köbler, U. Schöning, and J. Torán. On counting and approximation. *Acta Inf.*, 26(4):363–379, 1989.

[7] M. W. Krentel. The complexity of optimization problems. *J. Comput. Syst. Sci.*, 36(3):490–509, 1988.

[8] E. Kushilevitz and N. Nisan. *Communication Complexity.* Cambridge University Press, 1997.

[9] L. G. Valiant. Relative complexity of checking and evaluating. *Inf. Process. Lett.*, 5(1):20–23, 1976.

[10] L. G. Valiant. The complexity of computing the permanent. *Theor. Comput. Sci.*, 8:189–201, 1979.

[11] A. C.-C. Yao. Some complexity questions related to distributive computing (preliminary report). In *Conference Record of the Eleventh Annual ACM Symposium on Theory of Computing, 30 April-2 May, 1979, Atlanta, Georgia, USA*, pages 209–213. ACM, 1979.

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

93-02 *M.H. Scholl, C. Laasch, C. Rich, H.-J. Schek, M. Tresch*
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

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

94-09 *F.W. von Henke, A. Dold, H. Rueß, D. Schwier, M. Strecker*
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 *Helmuth Partsch*
Again on Recognition and Parsing of Context-Free Grammars:
Two Exercises in Transformational Programming

94-18 *Helmuth 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 Indepentent 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*
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, Helmuth 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*
$ADEPT_{flex}$ - 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

98-12 *Gerhard Schellhorn*
Proving Properties of Directed Graphs: A Problem Set for Automated Theorem Provers

98-13 *Gerhard Schellhorn, Wolfgang Reif*
Theorems from Compiler Verification: A Problem Set for Automated Theorem Provers

98-14 *Mohammad Ali Livani*
SHARE: A Transparent Mechanism for Reliable Broadcast Delivery in CAN

98-15 *Mohammad Ali Livani, Jörg Kaiser*
Predictable Atomic Multicast in the Controller Area Network (CAN)

99-01 *Susanne Boll, Wolfgang Klas, Utz Westermann*
A Comparison of Multimedia Document Models Concerning Advanced Requirements

99-02 *Thomas Bauer, Peter Dadam*
Verteilungsmodelle für Workflow-Management-Systeme - Klassifikation und Simulation

99-03 *Uwe Schöning*
On the Complexity of Constraint Satisfaction

99-04 *Ercument Canver*
Model-Checking zur Analyse von Message Sequence Charts über Statecharts

99-05 *Johannes Köbler, Wolfgang Lindner, Rainer Schuler*
Derandomizing RP if Boolean Circuits are not Learnable

99-06 *Utz Westermann, Wolfgang Klas*
Architecture of a DataBlade Module for the Integrated Management of Multimedia Assets

99-07 *Peter Dadam, Manfred Reichert*
Enterprise-wide and Cross-enterprise Workflow Management: Concepts, Systems, Applications. Paderborn, Germany, October 6, 1999, GI–Workshop Proceedings, Informatik '99

99-08 *Vikraman Arvind, Johannes Köbler*
Graph Isomorphism is Low for ZPP^{NP} and other Lowness results

99-09 *Thomas Bauer, Peter Dadam*
Efficient Distributed Workflow Management Based on Variable Server Assignments

2000-02 *Thomas Bauer, Peter Dadam*
Variable Serverzuordnungen und komplexe Bearbeiterzuordnungen im Workflow-Management-System ADEPT

2000-03 *Gregory Baratoff, Christian Toepfer, Heiko Neumann*
Combined space-variant maps for optical flow based navigation

2000-04 *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.)*
FM-Tools 2000: The 4th Workshop on Tools for System Design and Verification

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

2008-02 *Manfred Reichert, Peter Dadam, Martin Jurisch,l Ulrich Kreher, Kevin Göser, Markus Lauer*
Architectural Design of Flexible Process Management Technology

2008-03 *Frank Raiser*
Semi-Automatic Generation of CHR Solvers from Global Constraint Automata

2008-04 *Ramin Tavakoli Kolagari, Alexander Raschke, Matthias Schneiderhan, Ian Alexander*
Entscheidungsdokumentation bei der Entwicklung innovativer Systeme für produktlinien-basierte Entwicklungsprozesse

2008-05 *Markus Kalb, Claudia Dittrich, Peter Dadam*
Support of Relationships Among Moving Objects on Networks

2008-06 *Matthias Frank, Frank Kargl, Burkhard Stiller (Hg.)*
WMAN 2008 – KuVS Fachgespräch über Mobile Ad-hoc Netzwerke

2008-07 *M. Maucher, U. Schöning, H.A. Kestler*
An empirical assessment of local and population based search methods with different degrees of pseudorandomness

2008-08 *Henning Wunderlich*
Covers have structure

2008-09 *Karl-Heinz Niggl, Henning Wunderlich*
Implicit characterization of FPTIME and NC revisited

2008-10 *Henning Wunderlich*
On span-P^{cc} and related classes in structural communication complexity

Ulmer Informatik-Berichte
ISSN 0939-5091

Herausgeber:
Universität Ulm
Fakultät für Ingenieurwissenschaften und Informatik
89069 Ulm