数理解析研究所講究録 1301 巻 2003 年 69-78

# Complexity theoretical aspects of applicative theories in connection with boolean circuit computations

Satoru Kuroda
Gunma Prefectural Women's University

黒田 覚
群馬県立女子大

## 1 Introduction

The question concerning applicative theories and computational complexity was posed by S. Feferman [11]. Namely he asked whether a program similar to Buss' bounded arithmetic [4] can be carried through in the context of applicative theories or explicit mathematics.

This question was answered by T. Strahm [16], [17], [18]. He first presented the theory PTO which can be viewed as an applicative analogue of Cook's $PV$ or alternatively Ferreira's $PTCA$ [12]. Then he went on to define another theory $PT$ which has notation induction axiom for NP formulae. Strahm proved that the theory $PT$ is related to type 1 and type 2 poly-time functionals. However it is still open whether provably total terms of $PT$ of arbitrary type can be witnessed by poly-time computations.

In this paper we will deal with applicative theory for the constant depth circuit class $AC^0$. In [13] the author presented an applicative theory $CDD$ whose probably total functions are exactly those in $AC^0$. There we deal only with type 1 functions and here we will prove that provably total type 2 functionals of $CDD$ is witnessed by type 2 $AC^0$ functions in the sense of Clote et.al. [8].

Let us now state the construction of the paper. In the next section we will present basic definitions. In section 3 we will prove that the provably total type 2 functionals of $CDD$ can be realized by type 2 $AC^0$ functions. In section 4 we will present some questions concerning applicative theories and boolean circuit computation.

## 2 Basic definitions

In this section we will give basic definitions of our theories. We will introduce our base theory for binary words. The language $\mathcal{L}_{CDO}$ is a first order language with individual variables $a, b, c, \dots, x, y, z, \dots$ possibly with subscripts. We also have individual constants $\mathsf{K}, \mathsf{S}$ (combinators), $\mathsf{P}, \mathsf{P}_0, \mathsf{P}_1$ (pairing and unpairings), $\mathsf{D}$(definition by cases on binary words), $\epsilon$ (empty word) $0, 1$ (zero,one),

$\mathsf{C}_{\subseteq}$ (initial subword relation), $l_w$ (tally length of binary word), $*, \times$ (word concatenation and word multiplication), LD, MSP (and most significant part and the least digit), CR (concatenation recursion on notation operation). $\mathcal{L}_{CDO}$ has a function symbol $\cdot$ (term application) and relation symbols $\downarrow$ (defined), $W$ (binary words) and $=$ (equality). The language $\mathcal{L}_{CDD}$ is obtained from $\mathcal{L}_{CDO}$ by removing the CRN operator CR.

Terms are built up from variables and constants by using term application, that is, all variables and constants are terms and if $s, t$ are terms then $s \cdot t$ is also a term. Formulae are built up from atomic formulae using logical symbols $\wedge$, $\vee$, $\supset$, $\forall$ and $\exists$. We assume that the negation symbol $\neg$ is not in our language but defined as $\neg A \equiv (A \supset \mathsf{K} = \mathsf{S})$. Quantifiers of the form $\forall x \in W$ and $\exists x \in W$ are called $W$-bounded and their intended meanings are

$$\begin{aligned} \forall x \in W A(x) &\equiv \forall x (x \in W \supset A(x)), \\ \exists x \in W A(x) &\equiv \exists x (x \in W \wedge A(x)) \end{aligned}$$

respectively. Furthermore, quantifiers of the form $\forall x \subseteq t$ and $\exists x \subseteq t$ are called sharply bounded and their intended meanings are

$$\begin{aligned} \forall x \subseteq t A(x) &\equiv \forall x \in W(x \subseteq t \supset A(x)), \\ \exists x \subseteq t A(x) &\equiv \exists x \in W(x \subseteq t \wedge A(x)) \end{aligned}$$

respectively. A formula is called $W$-free if it does not contain the predicate symbol $W$. A formula is called semi-positive if for all subformula of the form $A \subset B$, $A$ is an atomic formula. Some extra symbols which are not included in the language are used for convenience. First we write $s \subseteq t$ instead of $\mathsf{C}_{\subseteq}$. Also $*$ and $\times$ are presented in infix notation as $s * t$ and $s \times t$ and we write $x0$ and $x1$ for $x * 0$ and $x * 1$ respectively.

Since our theories are concerned with partial term application, a term is not necessarily be totally defined. So we introduce the partial equality relation as

$$s \simeq t \equiv (s \downarrow \vee t \downarrow) \supset (s = t).$$

Furthermore, we will use the following abbreviations in the sequel:

$$\begin{aligned} \vec{s} \in W &\equiv W(s_1) \vee \cdots \vee W(s_n), \\ t : W \to W &\equiv (\forall x \in W)(tx \in W), \\ t : W^{n+1} \to W &\equiv (\forall x \in W)(tx : W^n \to W). \end{aligned}$$

The theory $BOW$ (Base Theory for binary Words) is a $\mathcal{L}_{CDO}$ theory which consists of the following axioms:

**I** Partial combinatory algebra

**(1)** $\mathsf{K}xy \simeq x$

**(2)** $\mathsf{S}xy \downarrow \wedge \mathsf{S}xyz \simeq xz(yz)$

**II** Pairing and unpairings

(3) $\mathsf{P}_0\mathsf{P}xy = x \wedge \mathsf{P}_1\mathsf{P}xy = y$

**III** Binary word manipulation on $W$

**(4)** $\varepsilon \in W \wedge 0 \in W \wedge 1 \in W$

**(5)** $* : W^2 \to W$

**(6)** $x \in W \supset x * \varepsilon = x$

**(7)** $x \in W \wedge y \in W \supset x * (y * i) = (x * y) * i \quad (i = 0, 1)$

**(8)** $x \in W \wedge y \in W \supset x * 0 \neq y * 1 \wedge x * 0 \neq \varepsilon \wedge x * 1 \neq \varepsilon$

**(9)** $x \in W \wedge y \in W \wedge x * 0 = y * 0 \supset x = y$

**(10)** $x \in W \wedge y \in W \wedge x * 1 = y * 1 \supset x = y$

**IV** Word multiplication on $W$

**(11)** $\times : W^2 \to W$

**(12)** $x \in W \supset x \times \varepsilon = \varepsilon$

**(13)** $x \in W \wedge y \in W \supset x \times (y * 0) = (x \times y) * x \wedge x \times (y * 1) = (x \times y) * x$

**V** Predecessor on $W$

**(14)** $\mathsf{PR} : W \to W$

**(15)** $\mathsf{PR}\varepsilon = \varepsilon$

**(16)** $x \in W \supset \mathsf{PR}(x * 0) = \mathsf{PR}(x * 1) = x$

**(17)** $x \in W \wedge x \neq \varepsilon \supset (\mathsf{PR}x) * 0 = x \vee (\mathsf{PR}x) * 1 = x$

**VI** Initial subword relation on $W$

**(18)** $x \in W \wedge y \in W \supset x \subseteq y \vee x \not\subseteq y$

**(19)** $x \in W \supset (x \subseteq \varepsilon \leftrightarrow x = \varepsilon)$

**(20)** $x \in W \wedge y \in W \wedge y \neq \varepsilon \supset (x \subseteq y \leftrightarrow x \subseteq \mathsf{PR}y \vee x = y)$

**VII** Tally length of binary words

**(21)** $\mathsf{l}_W : W \to W \wedge \mathsf{l}_W\varepsilon = \varepsilon$

**(22)** $x \in W \to \mathsf{l}_W(x * 0) = (\mathsf{l}_W x) * 1 \wedge \mathsf{l}_W(x * 1) = (\mathsf{l}_W x) * 1$

**VIII** Most significant part and least digit

**(23)** $x \in W \supset \mathsf{MSP}x\varepsilon = x$

**(24)** $x \in Wy \in W \supset \mathsf{MSP}x(y0) = \mathsf{PR}(\mathsf{MSP}xy) \wedge \mathsf{MSP}x(y1) = \mathsf{PR}(\mathsf{MSP}xy)$

**(25)** $\mathsf{LD}\varepsilon = \varepsilon$

**(26)** $x \in W \supset \mathsf{LD}x0 = 0 \wedge \mathsf{LD}x1 = 1$

**IX** Definition by cases on $W$

(27) $a \in W \wedge b \in W \wedge a = b \supset \mathrm{D}xyab = x$

(28) $a \in W \wedge b \in W \wedge a \neq b \supset \mathrm{D}xyab = y$

The semantics for our theories are given using the well-known equivalence between combinatory logic with extensionality and $\lambda\eta$ (See Troelstra and van Dalen [19] for details). The open term model $\mathcal{M}(\lambda\eta)$ is based on $\lambda\eta$ reduction of the untyped lambda calculus and one simply extends $\lambda\eta$ reduction with the obvious clauses for the new constants. The universe of $\mathcal{M}(\lambda\eta)$ is the set of all terms. The equality $=$ is reduction to a common reduct and $W$ is interpreted as the set of all terms $t$ so that $t$ reduces to a standard binary word. The set of all binary words is denoted by $\mathbb{W}$. Finally the application $s \cdot t$ is simply the term $ts$. As usual $\mathcal{M}(\lambda\eta) \models A$ if $A$ is valid in the model $\mathcal{M}(\lambda\eta)$ for any formula $A$. set of all terms.

## 3 Theories for circuit complexity

We now state some results relating weak applicative theories and boolean circuit complexity classes. First we weill give some additional axioms which are used to form applicative systems for constant depth cirucuit class $AC^0$.
Concatenation recursion on notation

(29) $f : W \to W \wedge g \in P(W^2) \supset \mathsf{CR}fg : W^2 \to W$

(30)

$$\begin{aligned} &f : W \to W \ \wedge g \in P(W^2) \wedge x \in W \wedge y \in W \wedge y \neq \varepsilon \wedge \wedge h = \mathsf{CR}fg \\ &\supset hx\varepsilon = fx \wedge hxy = hx(\mathsf{PR}y) * gx(\mathsf{PR}y) \end{aligned}$$

Set induction on $W$

(31)

$$\begin{aligned} &f \in P(W) \wedge f\varepsilon = 0 \wedge (\forall x \in W)(f(\mathsf{PR}x) = 0 \supset fx = 0) \\ &\supset (\forall x \in W)(fx = 0) \end{aligned}$$

The system $CDD$ is a $\mathcal{L}_{CDD}$ theory whose axioms are (1) to (28) of $CDO$ extended by the following axioms:
Notation induction on $W$ for open $W$-free semi-positive $\mathcal{L}_{CDD}$ formulae

(32) $A(\varepsilon) \wedge \forall x \in W(A(\mathsf{PR}x) \supset A(x)) \supset (\forall x \in W)A(x)$,
where $A$ is an open $W$-free and semi-positive formula.

Bit comprehension for open $W$-free semi-positive $\mathcal{L}_{CDD}$ formulae

(33)
$$\begin{aligned} &(\forall \vec{x} \in W)(\exists z \in W)(\mathsf{l}_W z < \mathsf{l}_W(f\vec{x}) \\ &\qquad \wedge (\forall y \subseteq f\vec{x})(A(\vec{x}, y, f) \leftrightarrow \mathsf{BIT}zy = 1)). \end{aligned}$$

where $\mathsf{BIT}xy = \mathsf{LD}(\mathsf{MSP}xy)$ and $A$ is an open $W$-free and semi-positive formula.

As usual $AC^0$ is the class of functions computable by $U_E^*$-uniform circuit family of constant depth and polynomial size. For the precise definitions of these notions, see [?]. In [13] the author showed that the probably total first order terms of $CDO$ and $CDD$ are exactly functions in $AC^0$. More precisely,

**Definition 1** *A function $F : \mathbb{W} \to \mathbb{W}$ is provably total in a theory $T$ if there exists a term $t_F$ in the language of the theory in concern such that*

1. *$T \vdash t_F : W^n \to W$ and*

2. *$T \vdash t_F \bar{w}_1 \cdots \bar{w}_n = \overline{F(w_1, \dots, w_n)}$ for all $w_1, \dots, w_n \in \mathbb{W}$.*

**Theorem 1 (Kuroda [13])** *A function is provably total in CDD or CDT if and only if it is computable by $AC^0$ functions.*

By extending the method in [13] we can show that the provably total type 2 functionals of $CDT$ are type 2 $AC^0$ functionals. First we will clarify what is meant by type 2 $AC^0$ functionals.

Let $\mathbb{W}^{\mathbb{W}}$ be the set of type 1 functions, that is, a mapping from $\mathbb{W}$ to $\mathbb{W}$. A type 2 functional of rank $(k, l)$ is a mapping from $(\mathbb{W}^{\mathbb{W}})^k \times \mathbb{W}^l$ to $\mathbb{W}$. We denote type 2 functionals by upper case letters $F, G, \dots$ and so on. Let $\mathsf{Ap}$ be a type 2 functional defined by $\mathsf{Ap}(f, x) = f(x)$ for all $f, x$.

**Definition 2** *A functional $F$ is defined by functional composition from $G$, $H_1, \dots, H_n$ if*

$$F(\vec{f}, \vec{x}) = G(\vec{f}, H_1(\vec{f}, \vec{x}), \dots, H_n(\vec{f}, \vec{x}))$$

*for all $\vec{f}, \vec{x}$.*

**Definition 3** *A functional $F$ is defined by expansion from $G$ if*

$$F(\vec{f}, \vec{g}, \vec{x}, \vec{y}) = G(\vec{f}, \vec{x})$$

*for all $\vec{f}, \vec{g}, \vec{x}, \vec{y}$.*

**Definition 4** *A functional $F$ is defined by concatenation recursion on notation (CRN) from $G$, $H_0$ and $H_1$ if*

$$\begin{array}{rcll} F(\vec{f}, \vec{x}, \varepsilon) & = & G(\vec{f}, \vec{x}) & \\ F(\vec{f}, \vec{x}, y * i) & = & F_i(\vec{f}, \vec{x}, y) * H_i(\vec{f}, \vec{x}, y) & i = 0, 1 \end{array}$$

*provided $H_i(\vec{f}, \vec{x}) \leq 1$ for all $\vec{f}, \vec{x}$.*

**Definition 5** *The class $F_2AC^0$ of type 2 functionals is the smallest class of functions containing 0 (zero function), $s_0$, $s_1$ (successors on binary words), $P_k^n$ (projection), $x \# y = 2^{|x| \cdot |y|}$, $\mathsf{Ap}$ and closed under functional composition, expansion and concatenation recursion on notation operations.*

**Theorem 2 (Clote, Kapron, Ignjatovic [8])** *A functional $F(\vec{f}, \vec{x})$ belongs to $F_2AC^0$ if and only if it is computable by a $U_E^*$-uniform circuit family with constant depth and $P(|\vec{f}|, |\vec{x}|)$ size for some polynomial $P$.*

Now we will extend the realizability theorem for $CDD$ to type 2 functionals. In [13] we defined the realizability relation for semi-positive formulae. Here we will extend this so that it can manipulate type 2 case. First we must clarify what it means for a type 2 functional to be definable in the structure $\mathcal{M}$. Let $\mathcal{W}^{\mathcal{W}}$ denote the set of all individuals $f$ in the universe $|\mathcal{M}|$ such that $f : \mathcal{M} \to \mathcal{M}$ is true in $\mathcal{M}$. Denote by $\hat{f}$ the function $\mathbb{W} \to \mathbb{W}$ which is defined by $f$ in $\mathcal{M}$.

**Definition 6** *A type 2 functional $F$ of rank $(k, l)$ is definable in $\mathcal{M}$ if there exists a closed term $t_F$ such*

$$\mathcal{M} \models t_F f_1 \cdots f_k \bar{w}_1 \cdots \bar{w}_l = \overline{F(\hat{f}_1, \dots, \hat{f}_k, w_1, \dots, w_l)}$$

*for all $f_1, \dots, f_k \in \mathcal{W}^{\mathcal{W}}$ and $w_1, \dots, w_l \in \mathbb{W}$.*

We also define the provably totality of type 2 functionals in a given applicative system. We use the following abbreviation:

$$t : (W^W)^k \times W^l \to W \equiv (\forall \vec{f} : W \to W)(\forall \vec{x} \in W) t\vec{f}\vec{x} \in W.$$

**Definition 7** *A type 2 functional $F$ of rank $(k, l)$ is provably total in a theory $T$ if there exists a closed term $t_F$ in the language of $T$ such that*

1. *$T \vdash t : (W^W)^k \times W^l \to W$, and*

2. *$t_F$ defines $F$ in the open term model $\mathcal{M}(\lambda\eta)$.*

Now let us extend the realizability for $CDD$ in [13] to type 2 functionals using the method by Strahm [18]. Let $\mathcal{SP}$ denote the set of semi-positive formulae. A formula is in the class $\mathcal{C}_0$ if either $A$ is in $\mathcal{SP}$ or $A$ is of the form $(B \supset C)$ or $(\forall x)(B \supset C)$ for some $B, C \in \mathcal{SP}$. Then cut elimination for $CDD$ implies the following:

**Theorem 3** *Let $\Gamma \to \Delta$ be a sequent where $\Gamma$ and $\Delta$ consists of $\mathcal{C}_0$ and $\mathcal{SP}$ formulae respectively. Suppose that $CDD \vdash \Gamma \to \Delta$. Then there exists a $CDD$-proof of $\Gamma \to \Delta$ whose sequents consists of $\mathcal{C}_0$ formula in the antecedent and $\mathcal{SP}$ formulae in the succedent.*

Next we will define the realizability relation for $\mathcal{C}_0$ formulae. For formulae in $\mathcal{SP}$ the set $\mathcal{R}$ of realizers is defined inductively as

1. if $\alpha$ is a binary string then $\alpha \in \mathcal{R}$ and

2. if $\alpha, \beta \in \mathcal{R}$ then $\langle \alpha, \beta \rangle \in \mathcal{R}$

where $\langle \cdot, \cdot \rangle$ is a pairing function.

**Definition 8** *For $\rho \in \mathcal{R}$ and a formula $A$ in csp, the realizability relation $\rho \rhd A$ (read "$\rho$ realizes $A$) is defined inductively as follows:*

- *For a $W$-free atomic formula $A$ $\rho \rhd t = s \Leftrightarrow \rho$ is any binary string and $\mathcal{M}(\lambda\eta) \models A$.*
- *$\rho \rhd t \in W_i \Leftrightarrow \mathcal{M}(\lambda\eta) \models t = \rho$ for some binary string $\rho$.*
- *$\rho \rhd A \vee B \Leftrightarrow \rho = \langle i, \rho_0 \rangle$ and either $i = 0$ and $\rho_0 \rhd A$ or $i = 1$ and $\rho_0 \rhd B$.*
- *$\rho \rhd A \wedge B \Leftrightarrow \rho = \langle \rho_1, \rho_2 \rangle$ with $\rho_1 \rhd A$ and $\rho_2 \rhd B$.*
- *$\rho \rhd A \supset B \Leftrightarrow \rho = \langle i, \rho_0 \rangle$ with either $i = 0$ and $\rho_0 \not\rhd A$ or $i = 1$ and $\rho \rhd B$.*
- *$\rho \rhd (\forall x)A \Leftrightarrow \rho \rhd A[x := a]$ where $a$ is not free in $A$.*
- *$\rho \rhd (\exists x)A \Leftrightarrow \rho \rhd A[x := t]$ for some term $t$ which is not free for $x$ in $A$.*
- *$\rho \rhd (\forall x \subseteq t)A(x) \Leftrightarrow \rho = \langle \rho_\varepsilon, \ldots, \rho_t \rangle$ where each element $\rho_a$ of $\rho$ corresponds to $a$ with $\varepsilon \subseteq a \subseteq t$ and $\rho_a \rhd A(a)$.*

The crucial part is the case for formulae in $\mathcal{C}_0 \setminus \mathcal{SP}$. Realizers $\Theta, \Phi, \Psi, \ldots$ for formulae in $\mathcal{C}_0 \setminus \mathcal{SP}$ are functions from $\mathbb{W}$ to $\mathbb{W}$. For $A, B \in \mathcal{SP}$ define

- $\Theta \rhd (A \supset B) \Leftrightarrow$ if $\rho \rhd A$ then $\Theta(\rho) \rhd B$ for all $\rho$,
- $\Theta \rhd (\forall x)(A(x) \supset B(x)) \Leftrightarrow \Theta \rhd (A(t) \supset B(t))$ for all term $t$.

Let $\Gamma$ be a finite sequence of formulae $A_1, \ldots, A_m, B_1, \ldots, B_n$ where $A_i \in \mathcal{SP}$ for each $1 \le i \le m$ and $B_j \in \mathcal{C}_0 \setminus \mathcal{SP}$ for each $1 \le j \le n$. Let $\vec{\Theta} = \Theta_1, \ldots, \Theta_m$ be functions from $\mathcal{M}$ to $\mathcal{M}$ and $\vec{\rho} = \rho_1, \ldots, \rho_n \in \mathcal{R}$. Then $\vec{\Theta}, \vec{\rho} \rhd \Gamma$ if $\Theta_i \rhd A_i$ for $1 \le i \le m$ and $\rho_j \rhd B_j$ for $1 \le j \le n$. Now we will state the main theorem;

**Theorem 4** *Let $\Gamma \to \Delta$ be a sequent such that $\Gamma$ and $\Delta$ consists of $\mathcal{C}_0$ and $\mathcal{SP}$ formulae respectively. Suppose that $CDD \vdash \Gamma[\vec{u}] \Rightarrow \Delta[\vec{u}]$. Then there exists $F$ in $F_2AC^0$ such that for all terms $\vec{s}$ and all $\vec{\Theta}$ and $\vec{\rho}$, if $\vec{\Theta}, \vec{\rho} \rhd \Gamma[\vec{s}]$ then for some $B$ in $\Delta[\vec{s}]$, $F(\vec{\Theta}, \vec{\rho}) \rhd \Delta[\vec{s}]$.*

**Corollary 1** *Let $t$ be a closed term such that the sequent*

$$f_1 : W \to W, \ldots, f_k : W \to W, x_1 \in W, \ldots, x_l \in W \Rightarrow t f_1 \cdots f_k x_1 \cdots x_l \in W$$

*is provable in $CDD$. Then there exists type 2 functional $F$ in $F_2AC^0$ of rank $(k, l)$ such that $t$ defines $F$ in the open term model $\mathcal{M}(\lambda\eta)$.*

## 4 Future research

In this section we give some questions concerning weak applicative theories and their connections with computational complexity theory.

## 4.1 Applicative theory for ALOGTIME

Find a right applicative theory whose provably total functions are exactly those in ALOGTIME. Some possible approaches are listed below:

1. Construct a theory based on Arai's AID [1] or Pitt's $T_1$ [14]. Note that the former does not refer to ALOGTIME functions but only to ALOGTIME predicates while the latter is a quantifier-free theory.

2. Another possibility is to use safe characterization of ALOGTIME as a recursion theoretical base of the theory. An applicative theory based on safe recursion is obtained by Cantini [5] for poly-time functions. In this formulation Cantini introduced two predicates for binary words, namely $W_0$ for safe binary words and $W_1$ for normal binary words. In [3], Bloch presented a safe characterization of ALOGTIME using very safe divide-and-conquer recursion

$$f(z,b,\vec{x};\vec{y}) = \begin{cases} g(z,\vec{x};\vec{y}) & \text{if } |z| \leq \max(b,1) \\ h(;z,\vec{x},\vec{y},f(Fh(;z),b,\vec{x},\vec{y}), & \\ \qquad\qquad f(Bh(;z),b,\vec{x},\vec{y})) & \text{if } |z| > \max(b,1) \end{cases}$$

So Cantini's formulation might be used to characterize ALOGTIME in the applicative context with a slight modification. Since the step function $h$ in very safe DCR contains no normal parameters, it follows that the definition tree of $h$ cannot contain recursion scheme. The difficulty lies in finding a corresponding restrictions on the proof system.

## 4.2 $Th - FO$ and CDD

Ferreira's string language theory $Th - FO$ [12] is another bounded arithmetic theory which corresponds to $AC^0$. This is based on the class $FO$ which consists of functions bitwise first order definable in finite structures. Let $Th - FO^\omega$ be a higher order analogue (define similarly as $PV^\omega$ of Cook and Urquhart). The main question here is whether there is a suitable interpretation from $Th - FO^\omega$ to $CDD$. Strahm [17] showed that this is true in the case of $PV^\omega$ and $PT$. The more important case is the converse, that is, is there a interpretation from $CDD$ to $Th - FO^\omega$? The positive answer to this question implies that all higher order $AC^0$ functions are provably total in $CDD$.

It is worth mentioning that safe recursion is closely related to feasible higher type functionals, see [2],[15]. So Cantini's characterization might be useful in considering the provably defined higher type functionals of weak applicative theories.

## References

[1] Arai, T. A bounded arithmetic AID for Frege system. Annals of Pure and Applied Logic. 103, 2000, pp.155-199.

[2] Bellantoni, S. J., K-H. Niggl, and H.Schwichtenberg. Higher type recursion, ramification and polynomial time. Annals of Pure and Applied Logic, 104, 2002, pp.17–30.

[3] Bloch, S. Function-algebraic characterizations of log and polylog parallel time, Computational Complexity, 4(2),1994, pp.175–205.

[4] Buss, S. R. Bounded Arithmetic. Bibliopolis, Napoli 1986.

[5] Cantini, A. Polytime, combinatory logic and positive safe induction, Archive for Mathematical Logic, 41, 2002, pp.169–189.

[6] Clote, P. Sequential, machine independent characterizations of the parallel complexity classes $ALOGTIME$, $AC^k$, $NC^k$ and $NC$. In: Feasible Mathematics, Birkhäuser, Boston-Basel 1990, pp.47–96.

[7] Clote, P. .Computation models and function algebras. Handbook of Computability Theory, E.R. Griffor ed., Elsevier Science B.V. 1999, pp. 589–681.

[8] Clote. P, B. Kapron and A. Ignjatovic. Parallel computable higher type functionals. Proc. IEEE 34th Symp. on Foundations of Computer Science, 1993. pp.72–83.

[9] Clote, P. and G.Takeuti First order bounded arithmetic and small Boolean circuit complexity class. In: Feasible Mathematics II, Birkhäuser, Boston-Basel 1995, pp.154–218.

[10] Cook, S. Feasibly constructive proofs and the propositional calculus. 7th ACM Symp. on Theory of Computing 1975, pp.83–97.

[11] Feferman, S. Logics for termination and correctness of functional programs. In: Logic and Computation W. Sieg, Ed., vol. 106 of Contemporary Mathematics. American Mathematical Society, 1990, pp.101–136.

[12] Ferreira, F. On end-extensions of models of $\neg Exp$. Mathematical Logic Quarterly. 42, 1996, pp.1–18.

[13] Kuroda, S. Applicative theories for boolean circuit computation. Submitted.

[14] Pitt, F. A quantifier-free theory based on a string algebra for $NC^1$, In: DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 39, 1998, pp.229–252.

[15] Schwitchtenberg, H and S. J. Bellantoni. Feasible computation with higher types. Preprint.

[16] Strahm, T. Polynomial time operations in explicit mathematics. J. of Sym. Log. 62(2), 1997, pp.575–594.

[17] Strahm, T. Proof theoretic contributions to explicit mathematics Habilitationsschrift, Institut für Informatik und angewandte Mathematik, Universität Bern, 2001.

[18] Strahm, T. A proof-theoretic characterization of the basic feasible functionals. preprint.

[19] Troelstra, A. and van Dalen, D Constructivism in Mathematics. vol. I. North-Holland, Amsterdam, 1988.