An applicative theory for the #P hierarchy FCH

Reinhard Kahle\textsuperscript{1}, Isabel Oitavem\textsuperscript{1}, and Thomas Strahm\textsuperscript{2}

\textsuperscript{1}CMA and DM, FCT, Universidade Nova de Lisboa, Portugal
\textsuperscript{2}IAM, University of Berne, Switzerland

DICE 2016

Eindhoven

3.4.2016

This work of the first two authors was partially supported by the Portuguese Science Foundation, FCT, through the projects UID/MAT/00297/2013 (Centro de Matemática e Aplicações) and and by the project Método axiomática e teoria de categorias of the cooperation Portugal/France in the Programa PESSOA – 2015/2016.
We will work over binary words, $\mathbb{W}$.

- Initial Functions, $\mathcal{I}$
- Composition, $C$
- Bounded Recursion on Notation, $BRN$
  
  Given $g$, $h_0$, $h_1$, and $t$, the bounded recursion on notation $f = BRN(g, h_0, h_1, t)$ is given by:

  
  $f(\epsilon, \bar{x}) = g(\bar{x})$

  $f(y_i, \bar{x}) = h_i(y, \bar{x}, f(y, \bar{x}))|_{t(y, \bar{x})}, \quad i \in \{0, 1\}$

**Proposition (Cobham)**

$[\mathcal{I}; C, BRN] = FPtime$
Function Algebras for Complexity Classes (BC-style)

Function terms have sorted inputs: *normal* and *safe*. $f(\bar{x}; \bar{y})$ means that $\bar{x}$ are variables in normal input position, and $\bar{y}$ are variables in safe input position.

- $\mathcal{B}$ the set of basic functions.
- Sorted Composition: Given $g, \bar{r}, \bar{s}, f = \text{SC}(g, \bar{r}, \bar{s})$ is defined by:

  $$f(\bar{x}; \bar{y}) = g(\bar{r}(\bar{x}; \bar{y}); \bar{s}(\bar{x}; \bar{y})).$$

- Predicative recursion on notation: Given $g, h_0, h_1, f = \text{PRN}(g, h_0, h_1)$ is given by:

  $$f(\epsilon, \bar{x}; \bar{y}) = g(\bar{x}; \bar{y}),$$
  $$f(zi, \bar{x}; \bar{y}) = h_i(z, \bar{x}; \bar{y}, f(z, \bar{x}; \bar{z})), \quad i \in \{0, 1\}$$

We set $\text{ST}_0 = [\mathcal{B}; \text{SC}, \text{PRN}].$

Proposition (Bellantion-Cook, 1992)

$$\text{ST}_0 = \text{FPtime}.$$
Applicative Theories for Complexity Classes

- Applicative Theories are a framework which allows to tailor theories over natural numbers or binary words of different proof-theoretic strength.
- They are first-order theories comprising
  - (Untyped) Combinatory Algebra ($\lambda$ terms) and Pairing
    ($\rightarrow$ PCA in Robin’s talk this morning)
  - Axiomatized Natural Numbers or Binary Words.
    In the latter case, the theory provides a predicate $W$ for binary words.
  - Induction
- Complexity Classes are captures by the notion of provably total function.

**Definition**

For an applicative theory $T$, a function $F : W^n \rightarrow W$ is called provably total in $T$, if there exists a closed term $t_F$ such that

1. $T \vdash t_F \overline{w_1} \ldots \overline{w_n} = \overline{F(w_1, \ldots, w_n)}$ for all $w_1, \ldots, w_n \in W$, and
2. $T \vdash t_F : W^n \rightarrow W$. 
In general, the induction scheme is defined in close analogy to a given recursion scheme.

Then, it is rather straightforward that the functions of a given complexity class are provably total in a certain applicative theory.

For the upper bound, i.e., the proof that the applicative theory does not contain more provably total functions, one usually uses a realization argument.

Theories for BC-style function algebras use two copies of the datatype $W$, $W_0$ and $W_1$, to mimic the normal-safe distinction of input positions ("Cantini-style").
An applicative theory for FPtime

- Classical first order logic
- The base theory B
  - Combinatory algebra and pairing
  - Definition by cases on $W$
  - Closure, binary successors, and predecessors
  - Lexicographic successor and predecessor
  - Initial subword relation; word concatenation; word multiplication
- Induction on notation, $(\Sigma^b_W - \text{l}_W)$

$$f : W \rightarrow W \land \phi(\epsilon) \land (\forall x \in W. \phi(x) \rightarrow \phi(s_0 x) \land \phi(s_1 x)) \rightarrow \forall x \in W. \phi(x),$$

where $\phi(x)$ is of the form $\exists y \leq f \cdot x. \psi(f, x, y)$ for $\psi(f, x, y)$ a positive and $W$-free formula.

- Let PT be $B + (\Sigma^b_W - \text{l}_W)$.

Proposition (Strahm)

*The provably total functions of PT are exactly the functions of FPtime.*
Realizing Applicative Theories

In general, a realization of applicative follows the following steps:

- One reformulates the theories in Gentzen’s classical sequent calculus.
- One proves partial cut elimination, such that the remaining cuts are restricted to positive formulas.
- One realizes positive derivations with realizers from the appropriate complexity class.

The realizer $\rho$ “stores” essentially the values of arguments of the predicate $W$ together with the structure of the formula, even trivializing quantifiers. Just in the case of disjunctions, it has to “choose” the appropriate alternative.
Realizing Applicative Theories

**Definition**

Let $\rho \in \mathbb{W}$ and $\phi$ a positive formula. Then $\rho \triangleright \phi$ is inductively defined:

- $\rho \triangleright \mathbb{W}(t)$ if $\mathcal{M}(\lambda \eta) \models t = \bar{\rho}$,
- $\rho \triangleright (t_1 = t_2)$ if $\rho = \epsilon$ and $\mathcal{M}(\lambda \eta) \models t_1 = t_2$,
- $\rho \triangleright (\phi \land \psi)$ if $\rho = \langle \rho_0, \rho_1 \rangle$ and $\rho_0 \triangleright \phi$ and $\rho_1 \triangleright \psi$,
- $\rho \triangleright (\phi \lor \psi)$ if $\rho = \langle i, \rho_0 \rangle$ and either $i = 0$ and $\rho_0 \triangleright \phi$ or $i = 1$ and $\rho_0 \triangleright \psi$,
- $\rho \triangleright (\forall x. \phi(x))$ if $\rho \triangleright \phi(u)$ for a fresh variable $u$,
- $\rho \triangleright (\exists x. \phi(x))$ if $\rho \triangleright \phi(t)$ for some term $t$.

$\rho$ realizes a sequence $\Delta$ of $n$ formulas $\phi_1, \ldots, \phi_n$, if $\rho = \langle \bar{i}, \rho_0 \rangle$, $1 \leq i \leq n$; $\bar{i}$ the dyadic representation of the natural number $i$, and $\rho_0 \triangleright \phi_i$;

$\langle \cdot, \cdot \rangle$ is a low-level pairing function on binary words with projections $(\cdot)_0$ and $(\cdot)_1$. 

bookkeeping
The class $\#P$ was introduced by Valiant and consists of functions which count the number of accepting computations of non-deterministic Turing machines working in polynomial time;

for this class, Wagner introduced a hierarchy of counting functions $\text{FCH}$, by allowing queries to functions of the previous level;

Vollmer and Wagner gave a characterization of $\#P$ which uses a closure under a sum with an exponential number of terms;

Dal Lago, Kahle, Oitavem defined a corresponding function algebra for $\#P$ in Bellantoni-Cook-style using recursion instead of sum;

A hierarchy for this function algebra leads to $\text{FCH}$. 
Function algebra for $\#P$ and FCH (BC-style)

- Let $\text{ST}_0$ the Bellantoni-Cook function algebra for FPtime.
- $\text{SC}_k$ (sorted composition for $\text{ST}_k$): sorted composition of Bellantoni-Cook, but $\overline{r}$ and $\overline{s}$ taken from $\text{ST}_{k-1}$.
- $\text{TR}[+]$ (tree recursion for counting): given $g$, $f = \text{TR}[+](g)$ is:
  
  \[
  f(p, \epsilon, \overline{x};) = g(p, \epsilon, \overline{x};) \\
  f(p, z0, \overline{x};) = + (f(p0, z, \overline{x};) + f(p1, z, \overline{x};)) \\
  f(p, z1, \overline{x};) = + (f(p0, z, \overline{x};) + f(p1, z, \overline{x};))
  \]

- $\text{ST}_k = [\text{ST}_0; \text{SC}_{k-1}, \text{TR}[+]], \quad k \geq 1$, and $\text{ST}_\infty = [\text{ST}_0; \text{SC}, \text{TR}[+]]$.

Proposition (Dal Lago/K/O)

$\text{ST}_k = (\#P)_k$, $\text{ST}_\infty = \text{FCH}$. 
TR[+] (tree recursion for counting)

- TR[+] (tree recursion for counting): given \( g \), \( f = \text{TR}[+](g) \) is:
  
  \[
  f(p, \epsilon, \bar{x}; ) = g(p, \epsilon, \bar{x}; )
  
  f(p, z0, \bar{x}; ) = + (f(p0, z, \bar{x}; ), f(p1, z, \bar{x}; ); )
  
  f(p, z1, \bar{x}; ) = + (f(p0, z, \bar{x}; ), f(p1, z, \bar{x}; ); )
  
- \( f \) recurses along a tree spun by the second argument;
- \( p \) is a pointer which allows to identify in which branch of the tree the recursion is following;
- considering \( g \) at the leaves of the tree just as a zero-one function corresponding to rejecting/accepting computations, the step function \(+\) simply sums up the accepting computations of a non-deterministic Turing machine;
- it does not harm to allow \( g \) to be an arbitrary function.
One may note that the safe/normal distinction is obsolete for $\text{ST}_\infty$.

- Let $T_0$ the Cobham algebra for FPtime.
- Let $T_\infty = [T_0, C, \text{TR}[+]$, with $C$ usual composition.

**Proposition**

$$T_\infty = \text{FCH}.$$  

- As variation we get also

**Proposition**

$$T'_\infty = [I; C, \text{BRN}, \text{TR}[+]] = \text{FCH},$$

- $T'_\infty$ will be used to set up a corresponding applicative theory.
An applicative theory for FCH

Let ACH be the **intuitionistic** version of PT for FPtime augmented by the following induction scheme (corresponding to TR[+]):

\[(\forall p, \bar{x}. \exists z. t(p, \epsilon, \bar{x}) = z) \land \\
(\forall p, \bar{x}, y, z, w. t(p0, y, \bar{x}) = z \land t(p1, y, \bar{x}) = w \rightarrow \\
\quad t(p, y0, \bar{x}) = z + w \land t(p, y1, \bar{x}) = z + w) \rightarrow \\
\quad \forall p, \bar{x}, y. \exists z. t(p, y, \bar{x}) = z)\]

- The quantifiers range over \(W\).
- \(t\) is a term which can represent a function \(f\) defined by TR[+].
- It is immediate that a function defined by TR[+] is provably total in ACH.
Tree recursion and tree induction

\[ f(p, \epsilon, \bar{x};) = g(p, \epsilon, \bar{x};) \]
\[ f(p, z_0, \bar{x};) = +\left( f(p_0, z, \bar{x};), f(p_1, z, \bar{x};); \right) \]
\[ f(p, z_1, \bar{x};) = +\left( f(p_0, z, \bar{x};), f(p_1, z, \bar{x};); \right) \]

\[(\forall p, \bar{x}. \exists z. t(p, \epsilon, \bar{x}) = z) \land \]
\[(\forall p, \bar{x}, y, z, w. t(p_0, y, \bar{x}) = z \land t(p_1, y, \bar{x}) = w \rightarrow)
\[ t(p, y_0, \bar{x}) = z + w \land t(p, y_1, \bar{x}) = z + w \rightarrow)
\[ \forall p, \bar{x}, y. \exists z. t(p, y, \bar{x}) = z \]
TR[+] and bookkeeping?

- The induction of PT was formulated for “positive and $\mathbf{W}$-free formulae”; the induction here is restricted to equalities.
- This is because TR[+] allows only for addition as step function. Extending the class of formulae in the induction scheme would require that the step function could take care of the bookkeeping of the realization relation $\rho$.
- For the same reason, we have to restrict ourselves to an intuitionistic version, where — in the sequent calculus — only one formula occurs in the succedence and no further bookkeeping is needed.

**Question**

Is there a way to overcome this problem?
An applicative theory for \#P?

- \#P is uses the normal-safe distinction of input positions; this distinction needs to be mimicked in an applicative theory by the use of two copies of the datatype $\mathbf{W}$, $\mathbf{W}_0$ and $\mathbf{W}_1$ ("Cantini-style").

- In this context, function are in general to be "typed" as $f : \mathbf{W}_1 \to \mathbf{W}_0$.

- Now, consider (intuitionistic version of) the cut rule:

$$
\frac{
\Gamma, A \Rightarrow B \\
\Gamma \Rightarrow A
}{
\Gamma \Rightarrow B
}
$$

- The attempt to realize this rule gives us two functions $f_1(\bar{x}, y) : \mathbf{W}_1^{n+1} \to \mathbf{W}_0$ and $f_2(\bar{x}) : \mathbf{W}_1^n \to \mathbf{W}_0$ realizing $\Gamma, A \Rightarrow B$ and $\Gamma \Rightarrow A$, respectively.

- We need to define (in \#P) a function $g : \mathbf{W}_1^n \to \mathbf{W}_0$ realizing $\Gamma \Rightarrow B$.

- $g = f_1(\bar{x}, f_2(\bar{x}))$, however, does not work as it does not respect the normal and safe distinction in the composition scheme.
An applicative theory for $\#P$?

- The problem appears to be intrinsic for function algebras not closed under composition.
- Cantini solved it, for BC, by a sophisticated cut elimination argument which is not directly transferable to our case.
- For now, we stay with the

Question

How to define applicative theories for complexity classes not closed under composition?
One may observe that many probabilistic complexity classes can be obtained from \( \#P \) by a certain “post-processing”.

One may replace the addition \(+\) in \( TR[+] \) by a function which, at the top node of its call, compares the number of accepting with rejecting computations.

With this comparison, it is possible, for instance, to characterize the probabilistic complexity class \( PP \).

Such a comparison can also be performed in the corresponding induction scheme.

If we would have an applicative theory for \( \#P \), we could propose theories relating to \( BPP \), by \textit{incorporating the semantic condition in the theory}.

This is the ultimative goal of the work presented here.