An applicative theory for the #P hierarchy FCH

Reinhard Kahle[1], Isabel Oitavem[1], and Thomas Strahm[2]

[1]CMA and DM, FCT, Universidade Nova de Lisboa, Portugal
[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*.

FCT Fundação para a Ciência e a Tecnologia
MINISTÉRIO DA EDUCAÇÃO E CIÊNCIA

Function Algebras for Complexity Classes (Cobham-style)

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:

$$\begin{aligned} f(\epsilon, \bar{x}) &= g(\bar{x}) \\ f(yi, \bar{x}) &= h_i(y, \bar{x}, f(y, \bar{x}))|_{t(y,\bar{x})}, \quad i \in \{0, 1\} \end{aligned}$$

Proposition (Cobham)

$$[\mathcal{I}; \mathrm{C}, BRN] = \mathrm{FPtime}$$

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.

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

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

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

$$\begin{aligned} f(\epsilon, \bar{x};\bar{y}) &= g(\bar{x};\bar{y}), \\ f(zi, \bar{x};\bar{y}) &= h_i(z, \bar{x};\bar{y}, f(z, \bar{x};\bar{z})), \qquad i \in \{0,1\} \end{aligned}$$

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

Proposition (Bellantion-Cook, 1992)

$$\mathbf{ST}_0 = \mathrm{FPtime}.$$

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 (λ 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 $\mathbf{W}$ for binary words.
  - Induction
- Complexity Classes are captures by the notion of *provably total function*.

Definition

For an applicative theory T, a function $F : \mathbb{W}^n \rightarrow \mathbb{W}$ 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 $w_1,\ldots,w_n \in \mathbb{W}$, and
2. $T \vdash t_F : \mathbf{W}^n \rightarrow \mathbf{W}$.

Applicative Theories for Complexity Classes

- 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 $\mathbf{W}$, $\mathbf{W}_0$ and $\mathbf{W}_1$, to mimick 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, $(\Sigma^{\mathsf{b}}_{\mathbf{W}}\text{-}\mathsf{I}_{\mathbf{W}})$

$$f : \mathbf{W} \to \mathbf{W} \wedge \phi(\epsilon) \wedge (\forall x \in \mathbf{W}.\phi(x) \to \phi(\mathsf{s}_0\, x) \wedge \phi(\mathsf{s}_1\, x)) \to \forall x \in \mathbf{W}.\phi(x),$$

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

- Let PT be $\mathsf{B} + (\Sigma^{\mathsf{b}}_{\mathbf{W}}\text{-}\mathsf{I}_{\mathbf{W}})$.

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 ρ "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.

Definition

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

$$
\begin{array}{rcl}
\rho \triangleright \mathbf{W}(t) & \text{if} & \mathcal{M}(\lambda\eta) \models t = \overline{\rho}, \\
\rho \triangleright (t_1 = t_2) & \text{if} & \rho = \epsilon \text{ and } \mathcal{M}(\lambda\eta) \models t_1 = t_2, \\
\rho \triangleright (\phi \wedge \psi) & \text{if} & \rho = \langle \rho_0, \rho_1 \rangle \text{ and } \rho_0 \triangleright \phi \text{ and } \rho_1 \triangleright \psi, \\
\rho \triangleright (\phi \vee \psi) & \text{if} & \rho = \langle i, \rho_0 \rangle \text{ and} \\
 & & \text{either } i = 0 \text{ and } \rho_0 \triangleright \phi \text{ or } i = 1 \text{ and } \rho_0 \triangleright \psi, \\
\rho \triangleright (\forall x.\phi(x)) & \text{if} & \rho \triangleright \phi(u) \text{ for a fresh variable } u, \\
\rho \triangleright (\exists x.\phi(x)) & \text{if} & \rho \triangleright \phi(t) \text{ for some term } t.
\end{array}
$$

ρ realizes a sequence Δ of n formulas $\phi_1, \dots, \phi_n$, if $\rho = \langle \overline{i}, \rho_0 \rangle$, $1 \leq i \leq n$;
$\overline{i}$ 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

#P and FCH

- 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 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 FCH.

Function algebra for #P and FCH (BC-style)

- Let $\mathbf{ST}_0$ the Bellantoni-Cook function algebra for FPtime.
- SC_k (sorted composition for $\mathbf{ST}_k$): sorted composition of Bellantoni-Cook, but $\bar{r}$ and $\bar{s}$ taken from $\mathbf{ST}_{k-1}$.
- TR[+] (tree recursion for counting): given g, $f = \mathrm{TR}[+](g)$ is:

$$
\begin{aligned}
f(p, \epsilon, \bar{x};) &= g(p, \epsilon, \bar{x};) \\
f(p, z\mathbf{0}, \bar{x};) &= +(f(p\mathbf{0}, z, \bar{x};), f(p\mathbf{1}, z, \bar{x};);) \\
f(p, z\mathbf{1}, \bar{x};) &= +(f(p\mathbf{0}, z, \bar{x};), f(p\mathbf{1}, z, \bar{x};);)
\end{aligned}
$$

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

Proposition (Dal Lago/K/O)

$$
\begin{aligned}
\mathbf{ST}_k &= (\#\mathrm{P})_k, \\
\mathbf{ST}_\infty &= \mathrm{FCH}.
\end{aligned}
$$

TR[+] (tree recursion for counting)

- TR[+] (tree recursion for counting): given g, $f = \text{TR}[+](g)$ is:

$$\begin{aligned} f(p, \epsilon, \bar{x};) &= g(p, \epsilon, \bar{x};) \\ f(p, z\mathbf{0}, \bar{x};) &= +(f(p\mathbf{0}, z, \bar{x};), f(p\mathbf{1}, z, \bar{x};);) \\ f(p, z\mathbf{1}, \bar{x};) &= +(f(p\mathbf{0}, z, \bar{x};), f(p\mathbf{1}, z, \bar{x};);) \end{aligned}$$

- 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.

Function algebra for FCH (Cobham-style)

One may note that the safe/normal distinction is obsolate for $\mathbf{ST}_\infty$.

- Let $\mathbf{T}_0$ the Cobham algebra for FPtime.
- Let $\mathbf{T}_\infty = [\mathbf{T}_0, \mathsf{C}, \mathsf{TR}[+]]$, with C usual composition.

Proposition

$$\mathbf{T}_\infty = \mathsf{FCH}.$$

- As variation we get also

Proposition

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

- $\mathbf{T}'_\infty$ 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[+]):

$$
\begin{aligned}
&(\forall p, \bar{x}.\exists z.t(p, \epsilon, \bar{x}) = z) \wedge \\
&\big(\forall p, \bar{x}, y, z, w.t(p\mathbf{0}, y, \bar{x}) = z \wedge t(p\mathbf{1}, y, \bar{x}) = w \rightarrow \\
&\qquad\qquad t(p, y\mathbf{0}, \bar{x}) = z + w \wedge t(p, y\mathbf{1}, \bar{x}) = z + w\big) \rightarrow \\
&\qquad\qquad\qquad\qquad\qquad\qquad \forall p, \bar{x}, y.\exists z.t(p, y, \bar{x}) = z
\end{aligned}
$$

- The quantifiers range over $\mathbf{W}$.
- 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

$$
\begin{aligned}
f(p,\epsilon,\bar{x};) &= g(p,\epsilon,\bar{x};) \\
f(p,z\mathbf{0},\bar{x};) &= +(f(p\mathbf{0},z,\bar{x};), f(p\mathbf{1},z,\bar{x};);) \\
f(p,z\mathbf{1},\bar{x};) &= +(f(p\mathbf{0},z,\bar{x};), f(p\mathbf{1},z,\bar{x};);)
\end{aligned}
$$

$$
\begin{aligned}
&(\forall p,\bar{x}.\exists z.t(p,\epsilon,\bar{x}) = z) \wedge \\
&\big(\forall p,\bar{x},y,z,w.t(p\mathbf{0},y,\bar{x}) = z \wedge t(p\mathbf{1},y,\bar{x}) = w \rightarrow \\
&\qquad t(p,y\mathbf{0},\bar{x}) = z + w \wedge t(p,y\mathbf{1},\bar{x}) = z + w\big) \rightarrow \\
&\qquad\qquad \forall p,\bar{x},y.\exists z.t(p,y,\bar{x}) = z
\end{aligned}
$$

TR[+] and bookkeeping?

- The induction of PT was formulated for "positive and **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 ρ.
- 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?

realizer

An applicative theory for #P?

- #P is uses the normal-safe distinction of input positions; this distinction needs to be mimcked in an applicative theorie 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:

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

- 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?

Perspectives: PP, BPP, etc.

- 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 *incorporating the semantic condition in the theory*.
- This is the ultimative goal of the work presented here.