Logical relations and feasibility in higher types

Ulrich Berger
Department of Computer Science
Swansea University

Shonan Seminar 151
Higher-order complexity and its applications
9 October 2019

1 Introduction

How can the polytime functions on natural numbers be extended to higher-types such that the resulting hierarchy is λ-*closed*, that is, closed under definability by typed λ-terms?

Answers can be found for example in [Cob65, CK90, Coo92, CU93, KC96, IKR01].

Up to type level 2 these hierarchies coincide and are considered to define the 'right' notion of type 2 feasibility.

At types beyond level 3 it is unclear what is the best extension.

Can one use *logical relations* to define feasibility in higher types?

2 Type structures

- $A \to B :=$ set of all functions from A to B.
- Combinators:
  $\mathsf{K}_{AB} \colon A \to B \to A$, $\quad \mathsf{K}_{AB}\, a\, b = a$
  $\mathsf{S}_{ABC} \colon (A \to B \to C) \to (A \to B) \to A \to C$, $\quad \mathsf{S}_{ABC}\, f\, g\, a = f\, a\, (g\, a)$
- *Types:* Given a set (of symbols) T_0 of *base types* we define the set T of *simple types* as the closure of the base types under the formal function type construction, $\rho \to \sigma$.
- A *type structure* is a family of sets $D = \{D(\rho) \mid \rho \in T\}$ such that $D(\rho \to \sigma) \subseteq D(\rho) \to D(\sigma)$ for all $\rho, \sigma \in T$, and D is *combinatorially complete*, i.e.

$$\mathsf{K}_{D(\rho)D(\sigma)} \in D(\rho \to \sigma \to \rho)$$
$$\mathsf{S}_{D(\rho)D(\sigma)D(\eta)} \in D((\rho \to \sigma \to \eta) \to (\rho \to \sigma) \to \rho \to \eta)$$

  for all $\rho, \sigma, \eta \in T$.

3 Typed λ-terms

- *Typed λ-terms (terms)*, are constructed from typed variables, $x^\rho, y^\rho, \ldots$, by abstraction, $(\lambda x^\rho M^\sigma)^{\rho\to\sigma}$, and application, $(M^{\rho\to\sigma} N^\rho)^\sigma$.

- Λ ($\Lambda(\rho)$) denotes the set of of all terms (of type ρ).

- Terms can be interpreted in any type structure D. More precisely, there is a map $[\![\cdot]\!][\cdot \mapsto \cdot]$ such that for each term M^ρ with $\mathsf{FV(M)} \subseteq \tilde{\mathsf{x}}^{\tilde{\sigma}}$ and $\vec{a} \in D(\vec{\sigma})$ (the latter stands for $a_i \in D(\sigma_i)$ for $i = 1, \ldots, k$) we have $[\![M]\!][\vec{x} \mapsto \vec{a}] \in D(\rho)$ and the following equations hold

$$\begin{aligned} [\![x_i]\!][\vec{x} \mapsto \vec{a}] &= a_i \\ ([\![\lambda y\, M]\!][\vec{x} \mapsto \vec{a}])\, b &= [\![M]\!][y, \vec{x} \mapsto b, \vec{a}] \\ [\![M\, N]\!][\vec{x} \mapsto \vec{a}] &= ([\![M]\!][\vec{x} \mapsto \vec{a}])\,([\![N]\!][\vec{x} \mapsto \vec{a}]) \end{aligned}$$

- An element b of a type structure D is λ-*definable* over a set $A \subseteq D$ (more precisely $A \subseteq \bigcup\{D(\rho) \mid \rho \in T\}$) if there is a term M and $\vec{a} \in A$ such that $b = [\![M]\!][\vec{x} \mapsto \vec{a}]$.

4 Substructures

- The *level* of a type is defined by

$$\mathsf{level}(\tau) = 0, \text{ for } \tau \in T_0,$$
$$\mathsf{level}(\rho \to \sigma) = \max(\mathsf{level}(\rho) + 1, \mathsf{level}(\sigma)).$$

  Hence if τ is a base type, then $\mathsf{level}(\rho_1 \to \ldots \to \rho_n \to \tau)$ is the least natural number greater than all $\mathsf{level}(\rho_i)$.

In the following let $1 \leq k \leq \infty$.

- $T(\leq k) := \{\rho \in T \mid \mathsf{level}(\rho) \leq k\}$
- $D(\leq k) := \bigcup\{D(\rho) \mid \rho \in T(\leq k)\}$.
- $\rho \leq k$ means $\rho \in T(\leq k)$. The notations $T(< k)$, $D(< k)$, $\rho < k$, $\Lambda(\leq k)$, e.t.c. have similar meanings.
- A *k-substructure* of D is a family $P = \{P(\rho) \mid \rho \leq k\}$ such that $P(\rho) \subseteq D(\rho)$ for each $\rho \leq k$.
- A k-substructure P is *λ-closed* if it contains all elements in $D(\leq k)$ which are λ-definable over P (in particular P is closed under application).
- Let $l \geq k$. An l-substructure Q is an *extension* of a k-substructure P if $Q(\rho) = P(\rho)$ for all $\rho \leq k$.

5 Logical relations

A 0-substructure P (that is $P = \{P(\tau) \subseteq D(\tau) \mid \tau \text{ base type}\}$) can be extended to an ∞-substructure, called the *(unary) logical relation over* P by defining

$$P(\rho \to \sigma) = \{f \in D(\rho \to \sigma) \mid \forall a \in D(\rho)\,(a \in P(\rho) \to f\,a \in P(\sigma))\}$$

Logical relations of higher arities are defined similarly.

Theorem 5.1 (Fundamental lemma of logical relations)**.** *Logical relations are λ-closed.*

Not good for lifting polytime to higher types since the polytime functions are not a 0-substructure but a 1-substructure.

One way out: Parametrize P [BH12, DLGY19, Xu19].

6 Extensions of substructures

Fix a type structure D.

Given a λ-closed k-substructure P is there a smallest/largest λ-closed ∞-substructure extending P?

The answer depends on how we order ∞-substructures.

Typewise inclusion:

- There is always a smallest ∞-extension, namely all λ-definable elements over P.
- By Zorn's Lemma, there are maximal extensions.
- In general, there might be no largest extension.

Lexicographic levelwise inclusion:

$Q <_{\text{lex}} Q'$ iff
there is $l < \infty$ such that $Q(\leq l) = Q'(\leq l)$ but $Q(l+1) \subset Q'(l+1)$.

Theorem 6.1 (Extension)**.** *Every λ-closed k substructure has a (unique) $<_{lex}$-largest λ-closed ∞-extension.*

7 Proof of the Extension Theorem

Let P be a λ-closed k-substructure of D.

An element $a \in D$ is *conservative over* P if all elements of $D(\leq k)$ which are λ-definable over $P \cup \{a\}$ are in P. We set

$$P^+ := \{a \in D(\leq k+1) \mid a \text{ is conservative over } P\}$$

The idea to use conservativity to define λ-closed substructures essentially goes back to ideas of [LM84] and [SV93].

The Extension Theorem follows immediately from the following:

Proposition 7.1. *P^+ is a λ-closed $k+1$-substructure which extends P. P^+ is largest in the sense that for any other λ-closed $k+1$-substructure Q extending P we have $Q \subseteq P^+$.*

It is clear that P^+ is an extension of P and that any λ-closed $k+1$-substructure extending P must be contained in P^+. It remains to be proven that P^+ is λ-closed.

Let $\rho \leq k+1$, say $\rho = \vec{\rho} \to \tau$ where $\rho = \rho_1, \ldots, \rho_n$ and τ is a base type. Hence $\vec{\rho} \leq n$ i.e. $\rho_i \leq k$ for all i.

We call $a \in D(\rho)$ *Banach-Mazur over* P if for all tuples of types $\vec{\sigma} < n$ and $\vec{g}$ where $g_i \in P(\vec{\sigma} \to \rho_i)$ we have $a \circ (\vec{g}) \in P(\vec{\sigma} \to \tau)$

(note that the expressions involved make sense since $\vec{\sigma} \to \rho_i \leq k$ and $\vec{\sigma} \to \tau \leq k$). The meaning of $a \circ (\vec{g})$ is the obvious one, namely $(a \circ (\vec{g}))\vec{b} = a(g_1\vec{b}) \ldots (g_n\vec{b})$.

Lemma 7.2. *Let P be a λ-closed k-substructure of D and let $a \in D(\leq k+1)$. Then a is conservative over P iff a is Banach-Mazur over P.*

Proof. If a is conservative over P, then a is also Banach-Mazur over P, since for any $\vec{g} \in P$ of appropriate types, $a \circ (\vec{g})$ is λ-definable over $P \cup \{a\}$ and hence in P.

For the converse, assume a is Banach-Mazur over P. We show, by induction on the lengths of terms $M \in \Lambda(\leq k)$ in long normal form with free variables among $u, \vec{x}$ of appropriate types that for any $\vec{b} \in P$ of appropriate types we have $c := [\![M]\!][u, \vec{x} \mapsto a, \vec{b}] \in P$. Let $M = \lambda\vec{y}.z\vec{M}$ and set $d_i := [\![\lambda\vec{y}.M_i]\!][u, \vec{x} \mapsto a, \vec{b}]$. By induction hypothesis, we have $d_i \in P$. If the head variable z is u, then $c = a \circ (\vec{d}) \in P$ because a was assumed to be Banach-Mazur over P. Otherwise, i.e. if z is one of the variables $\vec{x}, \vec{y}$, then c is λ-definable over P and hence in P because P is assumed to be λ-closed. □

We are now in a position to complete the proof of proposition 7.1. We show, by induction on the lengths of terms $M \in \Lambda(\leq k+1)$ in long normal form with free variables among $\vec{x}$ of types $\vec{\rho} \in T(\leq k+1)$, that for any tuple $\vec{a} \in P^+(\vec{\rho})$ we have $c := [\![M]\!][\vec{x} \mapsto \vec{a}] \in P^+$. By lemma 7.2 it suffices to show that c is Banach-Mazur over P. To this end let $g_i \in P(\vec{\sigma} \to \rho_i)$ where $\vec{\sigma} < k$. We need to show $c \circ (\vec{g}) \in P$. Let $M = \lambda\vec{y}.z\vec{M}$ and set $d_i := [\![\lambda\vec{y}.M_i]\!][\vec{x} \mapsto \vec{a}]$. By induction hypothesis, we have $d_i \in P^+(\leq k)$ and therefore $e_i := d_i \circ (\vec{g}) \in P$. Now, if $z = x_i$, then $c = a_i \circ (\vec{d})$ and therefore $c \circ (\vec{g}) = a_i \circ (\vec{e}) \in P$ because $a_i \in P^+$. If $z = y_i$, then $c \circ (\vec{g})$ clearly is λ-definable from $\vec{g}$ and $\vec{e}$ and therefore in P, since we assumed P to be λ-closed.

Set

$$P^k = P, \quad P^{n+1} = (P^n)^+, \quad P^\infty = \bigcup_{n \geq k} P^n$$

Then P^∞ is the $<_{\text{lex}}$-largest extension of P.

The construction of P^∞ is a generalization of a logical relation over a 0-substructure.

The Extension Theorem generalizes the Fundamental Lemma of logical relations.

8 Partial continuous functionals

In this section we build the ambient type structure D within the cartesian closed category of effective Scott domains as the type structure of

partial continuous functionals over the partial natural numbers

$D(\mathsf{nat}) = \mathbb{N}_\bot = \mathbb{N} \cup \{\bot\}$ with the Scott-topology.

$D(\rho \to \sigma) = \{f \in D(\rho) \to D(\sigma) f$ continuous $\}$ with the pointwise topology.

- D has a natural notion of computability via effective enumerations of compact elements.
- The polytime functions **PTIME** form a λ-closed 1-substructure of D
- More generally, **BFF**, The Basic Feasible Functionals form a λ-closed ∞-substructure of D.

Questions:

1. Is for every computable λ-closed 1-substructure P the largest extension P^∞ computable? ('computable' means 'all elements are computable')
2. Is $\mathbf{PTIME}^\infty$ computable?
3. Is **BFF** the smallest λ-closed extension of **PTIME**?
4. Is **BFF** the largest λ-closed extension of **PTIME**, i.e., is $\mathbf{BFF} = \mathbf{PTIME}^\infty$?
5. is $\mathbf{BFF}(2) = \mathbf{PTIME}^+$?

Answers: 1 (NO), 2 (YES), 3 (NO), 4 (NO), 5 (?)

References

[BH12] U. Berger and T. Hou. Typed vs. untyped realizability. *Electr. Notes in Comp. Sci.*, 286:57–71, 2012. http://www.sciencedirect.com/science/article/pii/S1571066112000357.

[CK90] S. Cook and B. Kapron. Characterizations of the basic feasible functionals of finite type. In S. Buss and P. Scott, editors, *Feasible Mathematics*, pages 71–96. Birkhäuser, 1990.

[Cob65] A. Cobham. The intrinsic computational difficulty of functions. In Y. Bar-Hillel, editor, *Logic, Methodology and Philosophy of Science II*, pages 24–30. North–Holland, 1965.

[Coo92] S. Cook. Computability and complexity of higher type functions. In Y.N. Moschovakis, editor, *Logic from Computer Science, Proceedings of a Workshop held November 13–17, 1989*, number 21 in MSRI Publications, pages 51–72. Springer, 1992.

[CU93] S. Cook and A. Urquhart. Functional interpretations of feasibly constructive arithmetic. *Ann. Pure Appl. Logic*, 63:103–200, 1993.

[DLGY19] U. Dal Lago, F. Gavazzo, and A. Yoshimizu. Differential logical relations, part i: The simply-typed case. arXiv:1904.12137, 2019.

[IKR01] R. Irwin, B. Kapron, and J. Royer. On characterizations of the basic feasible functionals, Part I. *Journal of Functional Programming*, 11(1):117–153, 2001.

[KC96] B. Kapron and S. Cook. A new characterization of type 2 feasibility. *SIAM Journal on Computing*, 25:117–132, 1996.

[LM84] G. Longo and E. Moggi. The hereditarily partial effective functionals and recursion theory in higher types. *Jour. Symb. Logic*, 49(4):1319–1332, 1984.

[SV93] V. Sazonov and A. Voronkov. A construction of typed lambda models related to feasible computability. In *Computational Logic and Proof Theory. Proceedings of the Third Kurt Gödel Colloquium*, volume 713 of *LNCS*, pages 301–313. Springer, 1993.

[Xu19] Xhuangjie Xu. A syntactic approach to continuity of t-definable functionals. arXiv:1904.09794, 2019.