Code-free recursion & realizability

Eric Faber

Universiteit Utrecht

February 24, 2014

Outline

Combinatory Logic

- Combinatory logic was founded by Moses Schönfinkel in his article *"Über die Bausteine der mathematischen Logik"* in 1924.
- Intended as a *pre-logic* that would solve foundational issues that arise from the use of variables by eliminating them.
- Further development was mostly due to Haskell Curry in the 50s, after which combinatory logic regained interest in theoretical computer science

Moses Schönfinkel

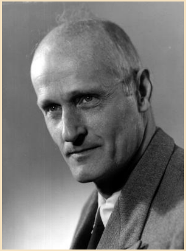
Haskell Curry

Computability Theory

- Founded in 1936 by work of Alan Turing, Alonzo Church, Stephen Kleene and Emil Post
- Mathematical rigorous definition of a *computation*
- First results on *undecidability*

Turing

Church



Kleene

Post

The Effective Topos

- In 1982, Martin Hyland discovered the *“Effective Topos”*
- From the viewpoint of a Topos as a *“constructive universe”*, the Effective Topos is an *effective* universe.
- The internal first-order logic of the Effective Topos coincides with Kleene’s notion of *realizability*

Martin Hyland

Why Study PCAs and Realizability Toposes?

- PCAs give rise to a lot of interpretations of constructive proofs
- Realizability Toposes give higher-order interpretations of this logic and help understand them
- Applications in Computer Science (e.g. Domain theory, programming language semantics)
- Applications in Topos theory and foundations, e.g. independence proofs.

Definition

A *partial applicative structure* (pas) is a set A together with a partial map $A \times A \to A$, denoted $(a, b) \mapsto ab$.

- We often refer to elements of A as "indices", since they index a set of partial functions defined by
$$b \mapsto ab$$
for each $a \in A$.
- Using a countable set of variables $V = \{x_0, x_1, \ldots\}$ we can build *terms*, e.g.:
$$t(x_0, x_1, x_2) = x_0 x_2 (x_1 x_2).$$
We can *evaluate* terms, e.g. for $a, b, c \in A$ $t(a, b, c)$ is defined if and only if $ac(bc)$ is defined, and in that case they are equal. Notation:
$$t(a, b, c) \simeq ac(bc).$$

Definition

A pas A is *combinatory complete* if there are $\mathsf{k}, \mathsf{s} \in A$ such that for all $a, b \in A$:

1. $\mathsf{s}ab$ is defined
2. $\mathsf{k}ab = a$
3. $\mathsf{s}abc \simeq ac(bc)$.

In that case we call A a *partial combinatory algebra* (pca).

Examples of pcas include:

- Any singleton set $\{*\}$ with $** = *$ is a pca, the *trivial pca*. Any pca with $\mathsf{k} = \mathsf{s}$ is trivial.
- *Kleene's* $\mathcal{K}_1$, the pca on $\mathbb{N}$ defined by

$$nm \simeq \varphi_n(m)$$

  where φ_n is the partial recursive function with index n.
- Any model of untyped λ-calculus is a *total* pca. This means that application is always defined.

Import Facts

Theorem (Abstraction)

A pas A is a PCA if and only if for every term $t(x, x_1, \ldots, x_n)$ there is a term $\langle x\rangle t(x_1, \ldots, x_n)$ such that for all $a, a_1, \ldots, a_n \in A$:

$$\langle x\rangle t(a_1, \ldots, a_n)\downarrow$$
$$(\langle x\rangle t(a_1, \ldots, a_n))\, a \simeq t(a, a_1, \ldots, a_n).$$

Compare this term to $\lambda x.t(x, x_1, \ldots, x_n)$ in λ-calculus.

Theorem (Recursion theorem)

Let A be a pca. There are $y, z \in A$ such that for every $f \in A$:

(i) $yf \simeq f(yf)$

(ii) $zf\downarrow$ *and for all* $a \in A$:

$$zfx \simeq f(zf)x.$$

Elementary building blocks in a PCA

- We have terms for true/false: $\mathsf{T} = k$, $\mathsf{F} = \overline{k}$ where $\overline{k}$ satisfies:

$$\overline{k}ab = b.$$

- Consider the term

$$t := \langle v \rangle vab.$$

Then $t\mathsf{T} = a$, $t\mathsf{F} = b$. In other words:

$$tv := \text{ if } v \text{ then } a \text{ else } b.$$

- With only the combinators k, s, we can construct a pairing combinator p with projections $\mathsf{p}_0, \mathsf{p}_1$:

$$\mathsf{p}_0(\mathsf{p}ab) = a$$
$$\mathsf{p}_1(\mathsf{p}ab) = b$$

Every PCA has a set of natural numbers

Definition (Curry numerals)

Let A be a non-trivial pca. Then we define for every $n \in \mathbb{N}$ the *Curry numeral* $\overline{n} \in A$ as follows:

- $\overline{0} = \mathsf{i} = \mathsf{skk}$
- $\overline{n+1} = \mathsf{p}\overline{\mathsf{k}}\overline{n}$.

In every pca A, we can make definitions by recursion:

Proposition (Definition by Recursion)

For every $a, R \in A$, there is an $f \in A$ (recursive in a, R) such that

$$f\overline{0} = a$$

$$f\overline{n+1} = R\overline{n}(f\overline{n}).$$

Every PCA is Turing complete

Theorem

Let A be a non-trivial pca. For every partial recursive function $F : \mathbb{N}^k \to \mathbb{N}$, there exists $f \in A$ such that

$$f\overline{n_1}\cdots\overline{n_k} \simeq \overline{F(n_1, \ldots, n_k)}.$$

- Using the pairing combinator and definition by recursion, we can define tuples $[u_0, \ldots, u_n]$ of elements, such that functions determining *length*, as well as *concatenation* and projections are all recursive.
- We only need the combinators k, s and the requirement $\mathsf{k} \neq \mathsf{s}$!
- The programming language *Unlambda* consists only of these k, s operators and "application" as build-in functions. In theory, we can write any program we like in Unlambda!

Relative recursion can be generalized to PCAs

- For A a pca, $f : A \to A$ a function we can define a pca $A[f]$ in which f is adjoined as an oracle. An element $a \in A$ *interrogates* $b \in A$ if there exists $u = [u_0, \ldots, u_n]$ such that for all $i \leq n$:

$$a([b, u_0, \ldots, u_{i-1}]) = \mathrm{pF}v_i \text{ and } u_i = f(v_i).$$

- Define an application on A by: $a \cdot b \downarrow, a \cdot b = c$ if a interrogates b and

$$a([b, u_0, \ldots, u_n] = \mathrm{pT}c$$

- This yields a pca structure on A in which f is recursive. One can show that for $A = \mathcal{K}_1$, this is essentially the same thing as ordinary relative recursion in an oracle.

Recursion in a type 2 oracle can be generalized too

Theorem

*Let $F : A^A \to A$ be a functional. There exists a pca $A[F]$, with application $\cdot$, so that F becomes **representable**, i.e. there is $r \in A$ so that for all $f : A \to A$:*

$$(\forall b)a \cdot b = f(a) \Rightarrow r \cdot a = F(f).$$

For $A = \mathcal{K}_1$, this is essentially equivalent to recursion in a type 2 functional as defined by P. Hinman.

Let $E : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ be the functional:

$$E(f) = \begin{cases} 0 & \text{if } (\exists m) f(m) = 0 \\ 1 & \text{otherwise} . \end{cases}$$

Then $\mathcal{K}_1[E]$ consists of precisely the Π^1_1 functions, so it computes every arithmetical subset of $\mathbb{N}$.

Categories of Assemblies

Definition

Let A be a pca. An **assembly** is a pair (X, E) with X a set, and

$$E : X \to \mathcal{P}^*(A)$$

a function, where $\mathcal{P}^*(A)$ is the set of non-empty subsets of A.
A **morphism of assemblies** $f : (X, E) \to (Y, E')$ is given by a function $f : X \to Y$ and an element $r \in A$ such that:

$$(\forall x \in X)(\forall a \in E(x))\ ra\downarrow \text{ and } ra \in E'(f(x)).$$

- Example: Consider the assembly $(\mathbb{N}, N)$ where

$$N(n) = \{\overline{n}\}.$$

Assemblies have a rich structure

Proposition

For a pca A, there is a category $\mathrm{Ass}(A)$:

- *Objects are assemblies on A*
- *Arrows are morphisms of assemblies.*

Moreover, $\mathrm{Ass}(A)$ is regular, cartesian closed and has finite colimits.

- In fact, $\mathrm{Ass}(A)$ is a little more: it is a *quasi-topos*. Also, it has a natural numbers object:

$$(\mathbb{N}, N) \text{ where } N(n) = \{\overline{n}\}$$

- There is an embedding $\nabla : \mathrm{Set} \to \mathrm{Ass}(A)$:

$$\nabla(X) = (X, E) \text{ where } E(x) = A \text{ for all } x$$

Subobject classifiers

Definition

A **subobject classifier** for a category $\mathcal{C}$ is a pair (Ω, true) where Ω is an object of $\mathcal{C}$ and $\mathrm{true} : 1 \to \Omega$ is an arrow, such that for every subobject $A \to X$ there is a unique $\varphi : X \to \Omega$ with the property that

$$\begin{tikzcd} A \arrow[r] \arrow[d, tail, "m"] & 1 \arrow[d, "\mathrm{true}"] \\ X \arrow[r, "\varphi"] & \Omega \end{tikzcd}$$

is a pullback diagram.

- A category of assemblies $\mathrm{Ass}(A)$ does not have a subobject classifier :-(

Definition of a Topos

Definition

A (elementary) **Topos** is a category with the following properties:

- It is cartesian closed (binary products & exponentials)
- It has all finite limits
- It has a subobject classifier $\mathrm{true} : 1 \to \Omega$.

- The categorical properties of a Topos are “essentially the same” as in Set, e.g. we have a powerset, and very often we have a natural numbers object.
- Every topos is a model for higher-order intuitionistic logic
- Examples: Set, FinSet, $\mathrm{Set}^{\mathcal{C}^{\mathrm{op}}}$, $\mathrm{Sh}(\mathcal{C}, \mathrm{Cov})$.
- Realizability Toposes

Assemblies can be completed to a Topos

- For a pca A, $\mathrm{Ass}(A)$ is in general not *exact*. This roughly means that for equivalence relations on objects, there is not always a "good" quotient.
- A regular category $\mathcal{C}$ admits an *exact/regular completion* to an exact category $\mathcal{C}_{\mathrm{ex/reg}}$.

Theorem

For a pca A, $Ass(A)_{ex/reg}$ is a topos. It is called the Realizability Topos on $Ass(A)$, and we write

$$RT(A) := Ass(A)_{ex/reg}$$

- $\mathrm{RT}(\mathcal{K}_1)$ is called the *Effective Topos*
- For a lot of constructions we can work with assemblies.

Some facts about Realizability Toposes

- The first order logic of $\mathrm{RT}(\mathcal{K}_1)$ coincides with *Kleene realizability*.
- Every realizability topos $\mathrm{RT}(A)$ has a natural numbers object $\mathcal{N}$, it is the same as in $\mathrm{Ass}(A)$.
- In $\mathrm{RT}(\mathcal{K}_1)$, the morphisms $\mathcal{N} \to \mathcal{N}$ are precisely the computable functions.
- A lot of "strange theorems" hold in $\mathrm{RT}(\mathcal{K}_1)$. For example *"Brouwer's theorem"*: Every function from the reals to reals is continuous.
- Recall the functional $E : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$:

$$E(f) = \begin{cases} 0 & \text{if } (\exists m) f(m) = 0 \\ 1 & \text{otherwise .} \end{cases}$$

One can show that $\mathrm{RT}(\mathcal{K}_1[E])$ satisfies classical arithmetic.

Morphisms between PCAs

Definition

Let A, B be pcas. An *applicative morphism* $\gamma : A \to B$ is a function $\gamma : A \to \mathcal{P}^*(B)$ such that there exists $r \in B$ with the following property:

$$(\forall a, a' \in A) aa' \downarrow \Rightarrow r\gamma(a)\gamma(a') \downarrow \subseteq \gamma(aa').$$

We define a preorder $\leq$ on applicative morphisms; for $\gamma, \delta : A \to B$, define

$$\gamma \leq \delta \iff (\exists t)(\forall a) t\gamma(a) \subseteq \delta(a).$$

- We have to check that applicative morphisms are closed under composition!
- We obtain a preorder-enriched category **PCA**.

Morphisms of categories of Assemblies

Definition (van Oosten)

A functor $F : \mathrm{Ass}(A) \to \mathrm{Ass}(B)$ is an S–*functor* if it is the identity on the level of sets, i.e.

$$F(X, E) = (X, E').$$

Theorem (Longley)

(i) *Every applicative morphism $\gamma : A \to B$ gives rise to a regular S–functor $\gamma^* : Ass(A) \to Ass(B)$.*
Moreover, if $\gamma \leq \delta$ then there is a natural transformation $\gamma^* \Rightarrow \delta^*$.

(ii) *For every regular S–functor $F : Ass(A) \to Ass(B)$ there is an applicative morphism $\tilde{F} : A \to B$.*
Moreover, if there is a natural transformation $F \Rightarrow G$, then $\tilde{F} \leq \tilde{G}$.

(iii) *For γ, F as above, $(\tilde{\gamma^*}) \simeq \gamma$ and $(\tilde{F})^* \cong F$.*

Morphisms of Realizability Toposes

Proposition

Regular S–functors $Ass(A) \to Ass(B)$ correspond (up to isomorphism) precisely to regular functors $F : RT(A) \to RT(B)$ such that

$$F \circ \nabla_A \cong \nabla_B.$$

Definition

Let $\mathcal{E}, \mathcal{F}$ be toposes. A **geometric morphism** $f : \mathcal{F} \to \mathcal{E}$ is an adjoint pair $f^* \dashv f_*$ where $f^* : \mathcal{E} \to \mathcal{F}$, $f_* : \mathcal{F} \to \mathcal{E}$ and f^* preserves finite limits.

Geometric morphism of Realizability Toposes I

Theorem (Johnstone, 2013)

Every geometric morphism $f : RT(B) \to RT(A)$ is induced by an applicative morphism $\gamma : A \to B$.
These are precisely the applicative morphisms that are ***computationally dense*** *(Hofstra-van Oosten).*

Definition (Hofstra, van Oosten)

An applicative morphism $\gamma : A \to B$ is computationally dense if there exists $m \in A$ such that:

$$(\forall b \in B)(\exists a \in A)(\forall a' \in A) b\gamma(a') \downarrow \Rightarrow aa' \downarrow, m\gamma(aa') \subseteq b\gamma(a').$$

Geometric morphism of Realizability Toposes II

Proposition

An applicative morphism $\gamma : A \to B$ *is computationally dense iff there exists* $\delta : B \to A$ *such that*

$$\gamma\delta \leq \iota_B,$$

where $\iota_B : B \to B$ *is the identity (i.e.* $b \mapsto \{b\}$*).*

- So a geometric morphism $f^* \dashv f_* : \mathrm{RT}(B) \to \mathrm{RT}(A)$ corresponds to a "half-adjoint" pair

$$\gamma : A \to B$$
$$\delta : B \to A$$

If f_* were regular, this would be an adjoint pair. It seems that "applicative morphism" is a little too restrictive.

Extending applicative morphisms

- Define an application on $\mathcal{P}^*(A)$ by:

$$\alpha\alpha' = \begin{cases} \{aa' \mid a \in \alpha, a \in \alpha'\} & \text{if } aa' \downarrow \text{ for all } a \in \alpha, a' \in \alpha' \\ \text{undefined} & \text{else} \end{cases}.$$

  This yields an *order-pca*, a certain generalization of a pca.
- Define a "proto applicative morphism" $A \to B$ as a function

$$\gamma : \mathcal{P}^*(A) \to \mathcal{P}^*(B)$$

  for which there is $r \in B$ such that for all $\alpha, \alpha' \in \mathcal{P}^*(A)$, whenever $\alpha\alpha' \downarrow$, then

$$r\gamma(\alpha)\gamma(\alpha') \subseteq \gamma(\alpha\alpha').$$

Geometric morphism of Realizability Toposes III

Theorem

Proto applicative morphisms $A \to B$ correspond precisely to left-exact functors between categories of assemblies.
Consequently, geometric morphisms $f : RT(B) \to RT(A)$ correspond precisely to adjoint pairs of "proto applicative morphisms" $\gamma \dashv \delta$.

- For every pca A and function $f : A \to A$, $a \mapsto \{a\}$ is a computationally dense applicative morphism $A \to A[f]$. Therefore we have a geometric morphism:

$$\mathrm{RT}(A[f]) \to \mathrm{RT}(A).$$

Definition

Let $\mathcal{E}, \mathcal{F}$ be toposes. A geometric morphism $f^* \dashv f_* : \mathcal{F} \to \mathcal{E}$ is an *embedding* if $f^* f_* \cong 1_{\mathcal{F}}$. For such an embedding, we call $\mathcal{F}$ a **subtopos** of $\mathcal{E}$.

- Some subtoposes of $\mathrm{RT}(A)$ are realizability toposes. For example

$$\mathrm{RT}(A[f]) \to \mathrm{RT}(A)$$

  is a subtopos.
- The constant object functor $\nabla : \mathrm{Set} \to \mathrm{RT}(A)$ has a left adjoint Γ, and they give a geometric embedding $\mathrm{Set} \to \mathrm{RT}(A)$ is an embedding. In fact Set is the subtopos of $\neg\neg$-*sheaves*.
- For every subobject $m : S \rightarrowtail X$ of an object $X \in \mathrm{RT}(A)$, there is a “smallest subtopos” in which m becomes an isomorphism.

Turing degrees embed into the lattice of subtoposes

Theorem

Let $f : A \to A$ be a function. The smallest subtopos of $RT(A)$ in which f becomes computable is $RT(A[f])$.
This yields an embedding of the (generalized) Turing degrees in the lattice of subtoposes of $RT(A)$.

- We can show a similar thing for type 2 functionals $F : A^A \to A$.

A cute fact and an open question

Theorem

The least subtopos of $RT(A)$ that forces ∇ to preserve finite coproducts is Set if and only if there exists $g \in A$ such that

$$(\forall a \in A)(\exists n \in \mathbb{N}) g\overline{n} = a.$$

In other words, g is a partial surjection $\mathbb{N} \to A$ computable in A.

Theorem

Let A be a countable pca. The least subtopos of $RT(A)$ that forces every function $f : A \to A$ computable is Set.

Question: Does this hold in the uncountable case?