Concrete Abstract Computability Theory

Jetze Zoethout

Mathematical Institute talk
10 May 2022

Table of Contents

Table of Contents

Computable functions

Computable *partial* functions $\mathbb{N} \rightharpoonup \mathbb{N}$.

Computable functions

Computable *partial* functions $\mathbb{N} \rightharpoonup \mathbb{N}$.

Need a notion of algorithm:

- imperative: Turing machines, register machines, . . . , Python;
- declarative: recursive functions, λ-calculus, . . . , Haskell.

Computable functions

Computable *partial* functions $\mathbb{N} \rightharpoonup \mathbb{N}$.

Need a notion of algorithm:

- imperative: Turing machines, register machines, . . . , Python;
- declarative: recursive functions, λ-calculus, . . . , Haskell.

Two important observations:

- all these models yield the same notion of computable function $\mathbb{N} \rightharpoonup \mathbb{N}$;
- the number of computable functions is *countable*.

Leveling the playing field

Write φ_n for the computable partial function $\mathbb{N} \rightharpoonup \mathbb{N}$ given by the n-th algorithm (coding).

Leveling the playing field

Write φ_n for the computable partial function $\mathbb{N} \rightharpoonup \mathbb{N}$ given by the n-th algorithm (coding).

Example

There exists an $n \in \mathbb{N}$ such that $\varphi_n(m) = m^2$.

Leveling the playing field

Write φ_n for the computable partial function $\mathbb{N} \rightharpoonup \mathbb{N}$ given by the n-th algorithm (coding).

Example

There exists an $n \in \mathbb{N}$ such that $\varphi_n(m) = m^2$.

If your coding is reasonable, $\varphi_n(m)$ is computable in terms of n and m.

Leveling the playing field

Write φ_n for the computable partial function $\mathbb{N} \rightharpoonup \mathbb{N}$ given by the n-th algorithm (coding).

Example

There exists an $n \in \mathbb{N}$ such that $\varphi_n(m) = m^2$.

If your coding is reasonable, $\varphi_n(m)$ is computable in terms of n and m.

Application function

We define a partial function $\mathbb{N} \times \mathbb{N} \rightharpoonup \mathbb{N}$ by:

$$n \cdot m \simeq \varphi_n(m).$$

Currying

What about functions with multiple arguments (e.g., addition)?

Currying

What about functions with multiple arguments (e.g., addition)?

Example

There exists an $n \in \mathbb{N}$ such that $(n \cdot m) \cdot m' = m + m'$.
Here φ_n is a function that, given m, computes a *code* for an algorithm for the function $x \mapsto m + x$.

Currying

What about functions with multiple arguments (e.g., addition)?

Example

There exists an $n \in \mathbb{N}$ such that $(n \cdot m) \cdot m' = m + m'$.
Here φ_n is a function that, given m, computes a *code* for an algorithm for the function $x \mapsto m + x$.

In general: a function $f : \mathbb{N}^k \rightharpoonup \mathbb{N}$ is computable iff there exists an $n \in \mathbb{N}$ such that

$$(\cdots((n \cdot m_1) \cdot m_2) \cdots) \cdot m_k \simeq f(m_1, \ldots, m_k).$$

Currying

What about functions with multiple arguments (e.g., addition)?

Example

There exists an $n \in \mathbb{N}$ such that $(n \cdot m) \cdot m' = m + m'$.
Here φ_n is a function that, given m, computes a *code* for an algorithm for the function $x \mapsto m + x$.

In general: a function $f : \mathbb{N}^k \rightharpoonup \mathbb{N}$ is computable iff there exists an $n \in \mathbb{N}$ such that

$$(\cdots((n \cdot m_1) \cdot m_2) \cdots) \cdot m_k \simeq f(m_1, \ldots, m_k).$$

Write LHS as: $nm_1m_2 \cdots m_k$.

Combinatory completeness

A *term* is an expression built from variables $x_1, x_2, \ldots$ and application.

Combinatory completeness

A *term* is an expression built from variables $x_1, x_2, \ldots$ and application.

Theorem

For every term $t(x_1, \ldots, x_k)$, there exists an $n \in \mathbb{N}$ such that

$$nm_1 \cdots m_k \simeq t(m_1, \ldots, m_k).$$

Combinatory completeness

A *term* is an expression built from variables $x_1, x_2, \ldots$ and application.

Theorem

For every term $t(x_1, \ldots, x_k)$, there exists an $n \in \mathbb{N}$ such that

$$n m_1 \cdots m_k \simeq t(m_1, \ldots, m_k).$$

A *partial combinatory algebra* (PCA) is a set A equipped with an application function $A \times A \rightharpoonup A$ that has this property.

Table of Contents

The Scott topology

The underlying set of Scott's graph model is the powerset of the natural numbers: $\mathcal{P}\omega$.

The Scott topology

The underlying set of Scott's graph model is the powerset of the natural numbers: $\mathcal{P}\omega$.

The Scott topology

The basic opens of the Scott topology on $\mathcal{P}\omega$ are:

$$U_p = \{A \subseteq \mathbb{N} \mid p \subseteq A\}$$

where $p \subseteq \mathbb{N}$ is *finite*.

The Scott topology

The underlying set of Scott's graph model is the powerset of the natural numbers: $\mathcal{P}\omega$.

The Scott topology

The basic opens of the Scott topology on $\mathcal{P}\omega$ are:

$$U_p = \{A \subseteq \mathbb{N} \mid p \subseteq A\}$$

where $p \subseteq \mathbb{N}$ is *finite*.

Under $\mathcal{P}\omega \cong \{0,1\}^{\mathbb{N}}$, this is the product topology, where

$$\mathcal{O}(\{0,1\}) = \{\emptyset, \{1\}, \{0,1\}\}.$$

Scott-continuous functions

Observation

A function $F\colon \mathcal{P}\omega \to \mathcal{P}\omega$ is Scott-continuous iff

$$F(B) = \bigcup_{p \subseteq B \text{ finite}} F(p),$$

for all $B \subseteq \mathbb{N}$.

Scott-continuous functions

Observation

A function $F\colon \mathcal{P}\omega \to \mathcal{P}\omega$ is Scott-continuous iff

$$F(B) = \bigcup_{p \subseteq B \text{ finite}} F(p),$$

for all $B \subseteq \mathbb{N}$.

In particular:

- a Scott-continuous function is order-preserving;
- there are $|\mathcal{P}\omega|$ Scott-continuous functions.

Coding continuous functions I

A Scott-continuous function F is determined by the set of all pairs (p, n), where $p \subseteq \mathbb{N}$ finite and $n \in F(p)$.

A Scott-continuous function F is determined by the set of all pairs (p, n), where $p \subseteq \mathbb{N}$ finite and $n \in F(p)$.

Coding of pairs

Define the bijection $\mathbb{N} \times \mathbb{N} \to \mathbb{N}, (m, n) \mapsto \langle m, n \rangle$ by:

$$\langle m, n \rangle = \frac{1}{2}(m + n)(m + n + 1) + n.$$

Coding continuous functions I

A Scott-continuous function F is determined by the set of all pairs (p, n), where $p \subseteq \mathbb{N}$ finite and $n \in F(p)$.

Coding of pairs

Define the bijection $\mathbb{N} \times \mathbb{N} \to \mathbb{N}, (m, n) \mapsto \langle m, n \rangle$ by:

$$\langle m, n \rangle = \frac{1}{2}(m+n)(m+n+1) + n.$$

Coding of finite sets

Define the bijection $\mathbb{N} \to \mathcal{P}_{\text{fin}}\mathbb{N}, n \mapsto e_n$ by:

$$e_n = p \quad \text{iff} \quad n = \sum_{i \in p} 2^i.$$

Coding continuous functions II

Coding

For a Scott-continuous function $F\colon \mathcal{P}\omega \to \mathcal{P}\omega$, define

$$\mathrm{code}(F) = \{\langle m, n\rangle \mid n \in F(e_m)\}.$$

Coding continuous functions II

Coding

For a Scott-continuous function $F\colon \mathcal{P}\omega \to \mathcal{P}\omega$, define

$$\mathrm{code}(F) = \{\langle m, n\rangle \mid n \in F(e_m)\}.$$

Decoding

For $A, B \subseteq \mathbb{N}$, define:

$$A \cdot B = \{n \in \mathbb{N} \mid \exists m \in \mathbb{N}(\langle m, n\rangle \in A \text{ and } e_m \subseteq B)\}.$$

Coding continuous functions II

Coding

For a Scott-continuous function $F\colon \mathcal{P}\omega \to \mathcal{P}\omega$, define

$$\mathrm{code}(F) = \{\langle m, n\rangle \mid n \in F(e_m)\}.$$

Decoding

For $A, B \subseteq \mathbb{N}$, define:

$$A \cdot B = \{n \in \mathbb{N} \mid \exists m \in \mathbb{N}(\langle m, n\rangle \in A \text{ and } e_m \subseteq B)\}.$$

- If F is Scott-continuous, then $\mathrm{code}(F) \cdot B = F(B)$;
- The function $\mathcal{P}\omega \times \mathcal{P}\omega \to \mathcal{P}\omega, (A, B) \mapsto A \cdot B$ is itself Scott-continuous.

Combinatory completeness

Let $t(x_1, \ldots, x_n)$ be a term.

Combinatory completeness

Let $t(x_1, \ldots, x_n)$ be a term.

Then the function $(\mathcal{P}\omega)^n \to \mathcal{P}\omega, (B_1, \ldots, B_n) \mapsto t(B_1, \ldots, B_n)$ is Scott-continuous.

Combinatory completeness

Let $t(x_1, \ldots, x_n)$ be a term.

Then the function $(\mathcal{P}\omega)^n \to \mathcal{P}\omega, (B_1, \ldots, B_n) \mapsto t(B_1, \ldots, B_n)$ is Scott-continuous.

By repeatedly applying code$(-)$, we obtain $A \subseteq \mathbb{N}$ such that:

$$AB_1 \cdots B_n = t(B_1, \ldots, B_n)$$

for all $B_1, \ldots, B_n \subseteq \mathbb{N}$.

Examples of (non-)continuous functions

Example

The following functions are Scott-continuous ('computable'):

- $(A, B) \mapsto A \cup B$;
- $(A, B) \mapsto A \cap B$;
- $A \mapsto$ the closure of A under finite sums.

Examples of (non-)continuous functions

Example

The following functions are Scott-continuous ('computable'):

- $(A, B) \mapsto A \cup B$;
- $(A, B) \mapsto A \cap B$;
- $A \mapsto$ the closure of A under finite sums.

Non-example

The function $A \mapsto \mathbb{N} - A$ is not Scott-continuous.

Table of Contents

Oracle functions

The underlying set of the Van Oosten model is the set $\mathcal{B}$ of all partial functions $\mathbb{N} \rightharpoonup \mathbb{N}$.

Oracle functions

The underlying set of the Van Oosten model is the set $\mathcal{B}$ of all partial functions $\mathbb{N} \rightharpoonup \mathbb{N}$.

We consider functions $F\colon \mathcal{B} \to \mathcal{B}$ where the input acts as an **oracle**, i.e., a resource that may be consulted finitely many times.

Example of an oracle function

Example

The function $F\colon \mathcal{B} \to \mathcal{B}$ is defined as follows. If $\beta \in \mathcal{B}$ and $n \in \mathbb{N}$, we define $x_0, x_1, \ldots, x_n \in \mathbb{N}$ by:

$$x_0 = n, \quad x_{i+1} \simeq x_i + \beta(x_i),$$

and we set $F(\beta)(n) \simeq x_n$.

Example of an oracle function

Example

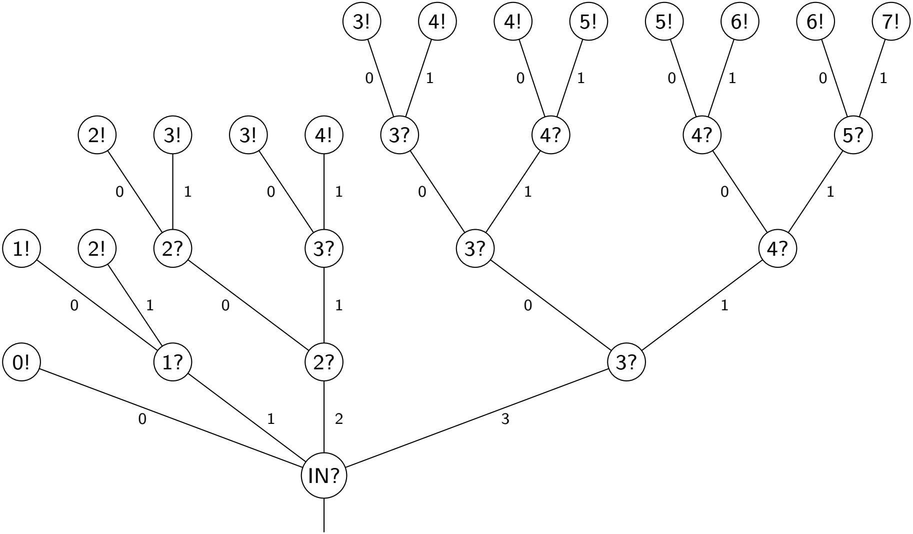
The function $F\colon \mathcal{B} \to \mathcal{B}$ is defined as follows. If $\beta \in \mathcal{B}$ and $n \in \mathbb{N}$, we define $x_0, x_1, \ldots, x_n \in \mathbb{N}$ by:

$$x_0 = n, \quad x_{i+1} \simeq x_i + \beta(x_i),$$

and we set $F(\beta)(n) \simeq x_n$.

- $F(\beta)(0) = 0$
- $F(\beta)(1) \simeq 1 + \beta(1)$
- $F(\beta)(2) \simeq 2 + \beta(2) + \beta(2 + \beta(2))$
- $F(\beta)(3) \simeq 3 + \beta(3) + \beta(3 + \beta(3)) + \beta(3 + \beta(3) + \beta(3 + \beta(3)))$
- ...

Tree representation



Coding oracle computation I

We can describe F by the function that maps tree positions to either queries or final outcomes.

Coding oracle computation I

We can describe F by the function that maps tree positions to either queries or final outcomes.

Coding of queries or outcomes

- we represent the query

  by $2q + 1$;
- we represent the outcome

  by $2m$.

Coding oracle computation II

Tree positions are really just finite sequences, telling us how we got there.

Coding oracle computation II

Tree positions are really just finite sequences, telling us how we got there.

Coding of finite sequences

Define the injective function $\mathbb{N}^* \to \mathbb{N}, (a_1, a_2, \ldots, a_k) \mapsto [a_1, a_2, \ldots, a_k]$ by:

$$[a_1, a_2, \ldots, a_k] = \prod_{i=1}^{k} p_i^{a_i+1} = 2^{a_1+1} 3^{a_2+1} \cdots p_k^{a_k+1},$$

where p_i is the i^{th} prime number.

Coding oracle computation III

For $\alpha, \beta \in \mathcal{B}$, we define $\alpha \cdot \beta \in \mathcal{B}$ as follows.

Coding oracle computation III

For $\alpha, \beta \in \mathcal{B}$, we define $\alpha \cdot \beta \in \mathcal{B}$ as follows.

We say that $(\alpha \cdot \beta)(n) = m$ iff there exist $u_0, \ldots, u_{k-1} \in \mathbb{N}$ such that:

Coding oracle computation III

For $\alpha, \beta \in \mathcal{B}$, we define $\alpha \cdot \beta \in \mathcal{B}$ as follows.

We say that $(\alpha \cdot \beta)(n) = m$ iff there exist $u_0, \ldots, u_{k-1} \in \mathbb{N}$ such that:

- there is $q_0 \in \mathbb{N}$ such that $\alpha([n]) = 2q_0 + 1$ and $\beta(q_0) = u_0$;

Coding oracle computation III

For $\alpha, \beta \in \mathcal{B}$, we define $\alpha \cdot \beta \in \mathcal{B}$ as follows.

We say that $(\alpha \cdot \beta)(n) = m$ iff there exist $u_0, \ldots, u_{k-1} \in \mathbb{N}$ such that:

- there is $q_0 \in \mathbb{N}$ such that $\alpha([n]) = 2q_0 + 1$ and $\beta(q_0) = u_0$;
- there is $q_1 \in \mathbb{N}$ such that $\alpha([n, u_0]) = 2q_1 + 1$ and $\beta(q_1) = u_1$;

Coding oracle computation III

For $\alpha, \beta \in \mathcal{B}$, we define $\alpha \cdot \beta \in \mathcal{B}$ as follows.

We say that $(\alpha \cdot \beta)(n) = m$ iff there exist $u_0, \ldots, u_{k-1} \in \mathbb{N}$ such that:

- there is $q_0 \in \mathbb{N}$ such that $\alpha([n]) = 2q_0 + 1$ and $\beta(q_0) = u_0$;
- there is $q_1 \in \mathbb{N}$ such that $\alpha([n, u_0]) = 2q_1 + 1$ and $\beta(q_1) = u_1$;
- there is $q_2 \in \mathbb{N}$ such that $\alpha([n, u_0, u_1]) = 2q_2 + 1$ and $\beta(q_2) = u_2$;

Coding oracle computation III

For $\alpha, \beta \in \mathcal{B}$, we define $\alpha \cdot \beta \in \mathcal{B}$ as follows.

We say that $(\alpha \cdot \beta)(n) = m$ iff there exist $u_0, \ldots, u_{k-1} \in \mathbb{N}$ such that:

- there is $q_0 \in \mathbb{N}$ such that $\alpha([n]) = 2q_0 + 1$ and $\beta(q_0) = u_0$;
- there is $q_1 \in \mathbb{N}$ such that $\alpha([n, u_0]) = 2q_1 + 1$ and $\beta(q_1) = u_1$;
- there is $q_2 \in \mathbb{N}$ such that $\alpha([n, u_0, u_1]) = 2q_2 + 1$ and $\beta(q_2) = u_2$;
- $\cdots$
- there is $q_{k-1} \in \mathbb{N}$ such that $\alpha([n, u_0, u_1, \ldots, u_{k-2}]) = 2q_{k-1} + 1$ and $\beta(q_{k-1}) = u_{k-1}$;

Coding oracle computation III

For $\alpha, \beta \in \mathcal{B}$, we define $\alpha \cdot \beta \in \mathcal{B}$ as follows.

We say that $(\alpha \cdot \beta)(n) = m$ iff there exist $u_0, \ldots, u_{k-1} \in \mathbb{N}$ such that:

- there is $q_0 \in \mathbb{N}$ such that $\alpha([n]) = 2q_0 + 1$ and $\beta(q_0) = u_0$;
- there is $q_1 \in \mathbb{N}$ such that $\alpha([n, u_0]) = 2q_1 + 1$ and $\beta(q_1) = u_1$;
- there is $q_2 \in \mathbb{N}$ such that $\alpha([n, u_0, u_1]) = 2q_2 + 1$ and $\beta(q_2) = u_2$;
- $\cdots$
- there is $q_{k-1} \in \mathbb{N}$ such that $\alpha([n, u_0, u_1, \ldots, u_{k-2}]) = 2q_{k-1} + 1$ and $\beta(q_{k-1}) = u_{k-1}$;
- $\alpha([n, u_0, u_1, \ldots, u_{k-1}]) = 2m$.

Coding oracle computation III

For $\alpha, \beta \in \mathcal{B}$, we define $\alpha \cdot \beta \in \mathcal{B}$ as follows.

We say that $(\alpha \cdot \beta)(n) = m$ iff there exist $u_0, \ldots, u_{k-1} \in \mathbb{N}$ such that:

- there is $q_0 \in \mathbb{N}$ such that $\alpha([n]) = 2q_0 + 1$ and $\beta(q_0) = u_0$;
- there is $q_1 \in \mathbb{N}$ such that $\alpha([n, u_0]) = 2q_1 + 1$ and $\beta(q_1) = u_1$;
- there is $q_2 \in \mathbb{N}$ such that $\alpha([n, u_0, u_1]) = 2q_2 + 1$ and $\beta(q_2) = u_2$;
- $\cdots$
- there is $q_{k-1} \in \mathbb{N}$ such that $\alpha([n, u_0, u_1, \ldots, u_{k-2}]) = 2q_{k-1} + 1$ and $\beta(q_{k-1}) = u_{k-1}$;
- $\alpha([n, u_0, u_1, \ldots, u_{k-1}]) = 2m$.

This makes $\mathcal{B}$ into a partial combinatory algebra.

Table of Contents

Applicative morphisms

Applicative morphisms

Let A, B be PCAs. An *applicative* morphism $A \to B$ is a function $f \colon A \to \mathcal{P}_{\neq\emptyset} B$ for which there exists an $r \in B$ such that:

$$\text{if } b \in f(a), b' \in f(a') \text{ and } aa' \text{ is defined, then } rbb' \in f(aa').$$

Applicative morphisms

Applicative morphisms

Let A, B be PCAs. An *applicative* morphism $A \to B$ is a function $f\colon A \to \mathcal{P}_{\neq\emptyset}B$ for which there exists an $r \in B$ such that:

$$\text{if } b \in f(a), b' \in f(a') \text{ and } aa' \text{ is defined, then } rbb' \in f(aa').$$

Intuition: f is a simulation of A inside B, and the elements in $f(a)$ represent a.

Applicative morphisms

Applicative morphisms

Let A, B be PCAs. An *applicative* morphism $A \to B$ is a function $f\colon A \to \mathcal{P}_{\neq\emptyset} B$ for which there exists an $r \in B$ such that:

$$\text{if } b \in f(a), b' \in f(a') \text{ and } aa' \text{ is defined, then } rbb' \in f(aa').$$

Intuition: f is a simulation of A inside B, and the elements in $f(a)$ represent a.

Category of PCAs

This yields a category of PCAs, where:

- $\mathrm{id}_A(a) = \{a\}$;
- if $A \xrightarrow{f} B \xrightarrow{g} C$, then $gf(a) = \bigcup_{b \in f(a)} g(b)$.

Example I

For $\alpha \in \mathcal{B}$, define $\mathrm{graph}(\alpha) = \{\langle n, \alpha(n) \rangle \mid n \in \mathrm{dom}\, \alpha\} \subseteq \mathbb{N}$.

Example I

For $\alpha \in \mathcal{B}$, define $\mathrm{graph}(\alpha) = \{\langle n, \alpha(n)\rangle \mid n \in \mathrm{dom}\,\alpha\} \subseteq \mathbb{N}$.

Example

There is an applicative morphism $f : \mathcal{B} \to \mathcal{P}\omega$ defined by:

$$f(\alpha) = \{\mathrm{graph}(\alpha)\}.$$

Example I

For $\alpha \in \mathcal{B}$, define $\mathrm{graph}(\alpha) = \{\langle n, \alpha(n)\rangle \mid n \in \mathrm{dom}\,\alpha\} \subseteq \mathbb{N}$.

Example

There is an applicative morphism $f\colon \mathcal{B} \to \mathcal{P}\omega$ defined by:

$$f(\alpha) = \{\mathrm{graph}(\alpha)\}.$$

Task: find a Scott-continuous function $F\colon \mathcal{P}\omega \times \mathcal{P}\omega \to \mathcal{P}\omega$ such that $F(\mathrm{graph}(\alpha), \mathrm{graph}(\beta)) = \mathrm{graph}(\alpha \cdot \beta)$.

Example I

For $\alpha \in \mathcal{B}$, define $\mathrm{graph}(\alpha) = \{\langle n, \alpha(n)\rangle \mid n \in \mathrm{dom}\,\alpha\} \subseteq \mathbb{N}$.

Example

There is an applicative morphism $f\colon \mathcal{B} \to \mathcal{P}\omega$ defined by:

$$f(\alpha) = \{\mathrm{graph}(\alpha)\}.$$

Task: find a Scott-continuous function $F\colon \mathcal{P}\omega \times \mathcal{P}\omega \to \mathcal{P}\omega$ such that $F(\mathrm{graph}(\alpha), \mathrm{graph}(\beta)) = \mathrm{graph}(\alpha \cdot \beta)$.

Idea: the fact that $(\alpha \cdot \beta)(n) = m$ depends on only finitely many values of α and β.

Example II

Example

There is an applicative morphism $g\colon \mathcal{P}\omega \to \mathcal{B}$ given by:

$$g(A) = \{\alpha \in \mathcal{B} \mid \mathrm{im}(\alpha) = A\}.$$

Example II

Example

There is an applicative morphism $g\colon \mathcal{P}\omega \to \mathcal{B}$ given by:

$$g(A) = \{\alpha \in \mathcal{B} \mid \mathrm{im}(\alpha) = A\}.$$

Task: find 'oracle function' $G\colon \mathcal{B} \times \mathcal{B} \to \mathcal{B}$ such that $\mathrm{im}(G(\alpha, \beta)) = \mathrm{im}(\alpha) \cdot \mathrm{im}(\beta)$.

Example II

Example

There is an applicative morphism $g\colon \mathcal{P}\omega \to \mathcal{B}$ given by:

$$g(A) = \{\alpha \in \mathcal{B} \mid \mathrm{im}(\alpha) = A\}.$$

Task: find 'oracle function' $G\colon \mathcal{B} \times \mathcal{B} \to \mathcal{B}$ such that $\mathrm{im}(G(\alpha, \beta)) = \mathrm{im}(\alpha) \cdot \mathrm{im}(\beta)$.

Idea: systematically inspect every element of $\mathrm{im}(\alpha)$ and every finite subset of $\mathrm{im}(\beta)$.

Comparing morphisms

Example (silly)

If A and B are PCAs, then we have the applicative morphism $f\colon A \to B$ defined by $f(a) = B$.

Comparing morphisms

Example (silly)

If A and B are PCAs, then we have the applicative morphism $f\colon A \to B$ defined by $f(a) = B$.

Ordering on applicative morphisms

If $f, f'\colon A \to B$ are applicative morphisms, we say that $f \leq f'$ if there exists an $s \in B$ such that: if $b \in f(a)$, then $sb \in f'(a)$.

Comparing morphisms

Example (silly)

If A and B are PCAs, then we have the applicative morphism $f : A \to B$ defined by $f(a) = B$.

Ordering on applicative morphisms

If $f, f' : A \to B$ are applicative morphisms, we say that $f \leq f'$ if there exists an $s \in B$ such that: if $b \in f(a)$, then $sb \in f'(a)$.

(This makes the category of PCAs enriched over preorders.)

An adjunction

Example

We have $\mathrm{id}_{\mathcal{B}} \leq gf$.

An adjunction

Example

We have $\mathrm{id}_{\mathcal{B}} \leq gf$.

Task: find 'oracle function' $F\colon \mathcal{B} \to \mathcal{B}$ such that $\mathrm{im}(F(\beta)) = \mathrm{graph}(\beta)$, for $\beta \in \mathcal{B}$.

An adjunction

Example

We have $\mathrm{id}_{\mathcal{B}} \leq gf$.

Task: find 'oracle function' $F\colon \mathcal{B} \to \mathcal{B}$ such that $\mathrm{im}(F(\beta)) = \mathrm{graph}(\beta)$, for $\beta \in \mathcal{B}$.

For example: $F(\beta)(n) \simeq \langle n, \beta(n) \rangle$.

An adjunction

Example

We have $\mathrm{id}_{\mathcal{B}} \leq gf$.

Task: find 'oracle function' $F\colon \mathcal{B} \to \mathcal{B}$ such that $\mathrm{im}(F(\beta)) = \mathrm{graph}(\beta)$, for $\beta \in \mathcal{B}$.

For example: $F(\beta)(n) \simeq \langle n, \beta(n) \rangle$.

Example

We have $fg \leq \mathrm{id}_{\mathcal{P}\omega}$.

An adjunction

Example

We have $\mathrm{id}_{\mathcal{B}} \leq gf$.

Task: find 'oracle function' $F\colon \mathcal{B} \to \mathcal{B}$ such that $\mathrm{im}(F(\beta)) = \mathrm{graph}(\beta)$, for $\beta \in \mathcal{B}$.

For example: $F(\beta)(n) \simeq \langle n, \beta(n) \rangle$.

Example

We have $fg \leq \mathrm{id}_{\mathcal{P}\omega}$.

Task: find Scott-continuous $G\colon \mathcal{P}\omega \to \mathcal{P}\omega$ such that $G(\mathrm{graph}(\alpha)) = \mathrm{im}(\alpha)$, for $\alpha \in \mathcal{B}$.

An adjunction

Example

We have $\mathrm{id}_{\mathcal{B}} \leq gf$.

Task: find 'oracle function' $F\colon \mathcal{B} \to \mathcal{B}$ such that $\mathrm{im}(F(\beta)) = \mathrm{graph}(\beta)$, for $\beta \in \mathcal{B}$.

For example: $F(\beta)(n) \simeq \langle n, \beta(n) \rangle$.

Example

We have $fg \leq \mathrm{id}_{\mathcal{P}\omega}$.

Task: find Scott-continuous $G\colon \mathcal{P}\omega \to \mathcal{P}\omega$ such that $G(\mathrm{graph}(\alpha)) = \mathrm{im}(\alpha)$, for $\alpha \in \mathcal{B}$.

For example: $G(B) = \{n \in \mathbb{N} \mid \exists m \in \mathbb{N}(\langle m, n \rangle \in B)\}$.

Thank you!

Defense: Monday 30 May at 12:15