

The λ -calculus, from minimal to classical logic

Webpage of the course

Davide Barbarossa

db2437@bath.ac.uk

Dept of Computer Science



Giulio Guerrieri

g.guerrieri@sussex.ac.uk

Dept of Informatics



ESSLLI Summer School, Bochum (Germany)

28/07/2025 – 01/08/2025

The λ -calculus, from minimal to classical logic

What will you (hopefully!) learn: You will have an idea of some basic but fundamental aspects of the relationship between (*λ -calculus based*) programming language theory and (*Curry-Howard based*) mathematical logic.

Lecture 1: Topology/domain-theory of the “graph model” over $\mathcal{P}(\mathbb{N})$

Davide Barbarossa

Lecture 2: The λ -calculus

Davide Barbarossa

Lecture 3: Category theory for denotational semantics

Giulio Guerrieri

Lecture 4: Curry-Howard and minimal logic

Giulio Guerrieri

Lecture 5: Krivine’s approach to classical logic

Davide Barbarossa

The λ -calculus,
from minimal to classical logic

Lecture 1:

Topology/domain-theory
of the “graph model” over $\mathcal{P}(\mathbb{N})$

Read the [notes](#): they are full of details, proofs, explanations, exercises, bibliography!

Davide Barbarossa

db2437@bath.ac.uk

Dept of Computer Science



- 1 What do we mean by *computable*?
- 2 Let's make this rigorous: a topology over $\mathcal{P}(\mathbb{N})$
- 3 The Scott-topology of a poset
- 4 Encoding (continuous) functions as points
- 5 Interesting properties of continuous functions!
- 6 Summary, exercises, bibliography

What do we mean by *computable*?

- 1 What do we mean by *computable*?
- 2 Let's make this rigorous: a topology over $\mathcal{P}(\mathbb{N})$
- 3 The Scott-topology of a poset
- 4 Encoding (continuous) functions as points
- 5 Interesting properties of continuous functions!
- 6 Summary, exercises, bibliography

What do we mean by *computable*?

Problem: Compute \sqrt{S} , for some given $S \in \mathbb{R}$

What do we mean by *computable*?

Problem: Compute $\sqrt{54899321104}$

What do we mean by *computable*?



Problem: Compute $\sqrt{54899321104}$

What do we mean by *computable*?



Problem: Compute $\sqrt{54899321104}$

$$x_0 \quad := \quad \textit{any positive number}$$

$$x_{n+1} \quad := \quad \frac{1}{2} \left(x_n + \frac{54899321104}{x_n} \right)$$

What do we mean by *computable*?



Problem: Compute $\sqrt{54899321104}$

$$x_0 \quad := \quad \text{any positive number}$$

$$x_{n+1} \quad := \quad \frac{1}{2} \left(x_n + \frac{54899321104}{x_n} \right)$$

$$\text{Then } \lim_{n \rightarrow +\infty} x_n = \sqrt{54899321104}$$

What do we mean by *computable*?

Problem: Compute $\sqrt{54899321104}$



$$x_0 \quad := \quad \text{any positive number}$$

$$x_{n+1} \quad := \quad \frac{1}{2} \left(x_n + \frac{54899321104}{x_n} \right)$$

$$\text{Then } \lim_{n \rightarrow +\infty} x_n = \sqrt{54899321104}$$

Heron

~~We~~ defined a function

$$H : \mathbb{N} \rightarrow \mathbb{R}, \quad H(n) := x_n$$

$$\text{such that } \lim_{n \rightarrow +\infty} H(n) = \sqrt{54899321104}$$

What do we mean by *computable*?

Problem: Compute $\sqrt{54899321104}$

$$x_0 \quad := \quad \text{any positive number}$$

$$x_{n+1} \quad := \quad \frac{1}{2} \left(x_n + \frac{54899321104}{x_n} \right)$$

$$\text{Then } \lim_{n \rightarrow +\infty} x_n = \sqrt{54899321104}$$



Heron

~~We~~ defined a function

$$H : \mathbb{N} \cup \{+\infty\} \rightarrow \mathbb{R}, \quad \begin{array}{ll} H(n) & := x_n \\ H(+\infty) & := \sqrt{54899321104} \end{array}$$

such that

$$\lim_{n \rightarrow +\infty} H(n) = \sqrt{54899321104} = H(+\infty) = H \left(\sup_{n < +\infty} n \right)$$

What do we mean by *computable*?

Problem: Compute all the prime numbers

What do we mean by *computable*?



Problem: Compute all the prime numbers

What do we mean by *computable*?



Problem: Compute all the prime numbers

Sieve of Eratosthenes!

What do we mean by *computable*?

Problem: Compute all the prime numbers

Sieve of Eratosthenes!

Eratosthenes

~~We~~ defined a function

$$E : \mathbb{N} \rightarrow \mathcal{P}(\mathbb{N}),$$

$E(n) :=$ *set of the first n prime numbers*

so that

$$\bigcup_{n \in \mathbb{N}} E(n) = \textit{set of all the prime numbers}$$



What do we mean by *computable*?

Problem: Compute all the prime numbers

Sieve of Eratosthenes!



Eratosthenes

~~We~~ defined a function

$$E : \mathbb{N} \cup \{+\infty\} \rightarrow \mathcal{P}(\mathbb{N}),$$

$$E(n) \quad := \quad \text{set of the first } n \text{ prime numbers}$$

$$E(+\infty) \quad := \quad \text{set of all the prime numbers}$$

such that

$$\bigcup_{n \in \mathbb{N}} E(n) = \text{set of all the prime numbers} = E(+\infty) = E\left(\sup_{n < +\infty} n\right)$$

What do we mean by *computable*?

Why do we say that H and E are computable?

What property do they share from the fact that they are given by an algorithm?

What do we mean by *computable*?

Why do we say that H and E are computable?

What property do they share from the fact that they are given by an algorithm?

(potentially) infinitary entity	$a \in \mathbb{N} \cup \{+\infty\}$	$r \in \mathbb{R}$
finitary entity	$n \in \mathbb{N}$	$q \in \mathbb{Q}$
approximation of infinitary by finitary	$n < +\infty$	$q \in \mathbb{Q}$ is “close” to $r \in \mathbb{R}$
it works because	$\sup_{n \in \mathbb{N}} n = +\infty$	\mathbb{Q} is dense in \mathbb{R}

What do we mean by *computable*?

Why do we say that H and E are computable?

What property do they share from the fact that they are given by an algorithm?

(potentially) infinitary entity	$a \in \mathbb{N} \cup \{+\infty\}$	$r \in \mathbb{R}$
finitary entity	$n \in \mathbb{N}$	$q \in \mathbb{Q}$
approximation of infinitary by finitary	$n < +\infty$	$q \in \mathbb{Q}$ is “close” to $r \in \mathbb{R}$
it works because	$\sup_{n \in \mathbb{N}} n = +\infty$	\mathbb{Q} is dense in \mathbb{R}

$$H : \mathbb{N} \cup \{+\infty\} \rightarrow \mathbb{R}$$

*For all q close to $H(+\infty)$,
there is $n < +\infty$ such that
 $H(n)$ is closer to $\sqrt{54899321104}$ than q is*

What do we mean by *computable*?

Why do we say that H and E are computable?

What property do they share from the fact that they are given by an algorithm?

(potentially) infinitary entity	$a \in \mathbb{N} \cup \{+\infty\}$	$r \in \mathbb{R}$	$a \subseteq \mathbb{N}$
finitary entity	$n \in \mathbb{N}$	$q \in \mathbb{Q}$	$e \subseteq_{\text{fin}} \mathbb{N}$
approximation of infinitary by finitary	$n < +\infty$	$q \in \mathbb{Q}$ is “close” to $r \in \mathbb{R}$	$e \subseteq_{\text{fin}} a$
it works because	$\sup_{n \in \mathbb{N}} n = +\infty$	\mathbb{Q} is dense in \mathbb{R}	$\bigcup_{e \subseteq_{\text{fin}} a} e = a$

$$H : \mathbb{N} \cup \{+\infty\} \rightarrow \mathbb{R}$$

For all q close to $H(+\infty)$,
there is $n < +\infty$ such that
 $H(n)$ is closer to $\sqrt{54899321104}$ than q is

What do we mean by *computable*?

Why do we say that H and E are computable?

What property do they share from the fact that they are given by an algorithm?

(potentially) infinitary entity	$a \in \mathbb{N} \cup \{+\infty\}$	$r \in \mathbb{R}$	$a \subseteq \mathbb{N}$
finitary entity	$n \in \mathbb{N}$	$q \in \mathbb{Q}$	$e \subseteq_{\text{fin}} \mathbb{N}$
approximation of infinitary by finitary	$n < +\infty$	$q \in \mathbb{Q}$ is “close” to $r \in \mathbb{R}$	$e \subseteq_{\text{fin}} a$
it works because	$\sup_{n \in \mathbb{N}} n = +\infty$	\mathbb{Q} is dense in \mathbb{R}	$\bigcup_{e \subseteq_{\text{fin}} a} e = a$

$$E : \mathbb{N} \cup \{+\infty\} \rightarrow \mathcal{P}(\mathbb{N})$$

For all $d \subseteq_{\text{fin}} E(+\infty)$,
there is $n < +\infty$ such that
 $E(n) \supseteq d$

What do we mean by *computable*?

Why do we say that H and E are computable?

What property do they share from the fact that they are given by an algorithm?

(potentially) infinitary entity	$a \in \mathbb{N} \cup \{+\infty\}$	$r \in \mathbb{R}$	$a \subseteq \mathbb{N}$
finitary entity	$n \in \mathbb{N}$	$q \in \mathbb{Q}$	$e \subseteq_{\text{fin}} \mathbb{N}$
approximation of infinitary by finitary	$n < +\infty$	$q \in \mathbb{Q}$ is “close” to $r \in \mathbb{R}$	$e \subseteq_{\text{fin}} a$
it works because	$\sup_{n \in \mathbb{N}} n = +\infty$	\mathbb{Q} is dense in \mathbb{R}	$\bigcup_{e \subseteq_{\text{fin}} a} e = a$

A generic f

*For all approximation d of $f(x)$,
there is an approximation e of x such that
 $f(e)$ is a better approximation of $f(x)$ than d is*

What do we mean by *computable*?

Why do we say that H and E are computable?

What property do they share from the fact that they are given by an algorithm?

(potentially) infinitary entity	$a \in \mathbb{N} \cup \{+\infty\}$	$r \in \mathbb{R}$	$a \subseteq \mathbb{N}$ amount of “information”
finitary entity	$n \in \mathbb{N}$	$q \in \mathbb{Q}$	$e \subseteq_{\text{fin}} \mathbb{N}$ finite amount of “information”
approximation of infinitary by finitary	$n < +\infty$	$q \in \mathbb{Q}$ is “close” to $r \in \mathbb{R}$	$e \subseteq_{\text{fin}} a$
it works because	$\sup_{n \in \mathbb{N}} n = +\infty$	\mathbb{Q} is dense in \mathbb{R}	$\bigcup_{e \subseteq_{\text{fin}} a} e = a$

A generic f

*For all approximation d of $f(x)$,
there is an approximation e of x such that
 $f(e)$ is a better approximation of $f(x)$ than d is*

What do we mean by *computable*?

Why do we say that H and E are computable?

What property do they share from the fact that they are given by an algorithm?

(potentially) infinitary entity	$a \in \mathbb{N} \cup \{+\infty\}$	$r \in \mathbb{R}$	$a \subseteq \mathbb{N}$ amount of “information”
finitary entity	$n \in \mathbb{N}$	$q \in \mathbb{Q}$	$e \subseteq_{\text{fin}} \mathbb{N}$ finite amount of “information”
approximation of infinitary by finitary	$n < +\infty$	$q \in \mathbb{Q}$ is “close” to $r \in \mathbb{R}$	$e \subseteq_{\text{fin}} a$
it works because	$\sup_{n \in \mathbb{N}} n = +\infty$	\mathbb{Q} is dense in \mathbb{R}	$\bigcup_{e \subseteq_{\text{fin}} a} e = a$

$$f : \mathcal{P}(\mathbb{N}) \rightarrow \mathcal{P}(\mathbb{N})$$

For all $d \subseteq_{\text{fin}} f(a)$,
there is $e \subseteq_{\text{fin}} a$ such that
 $f(e) \supseteq d$

What do we mean by *computable*?

Why do we say that H and E are computable?

What property do they share from the fact that they are given by an algorithm?

(potentially) infinitary entity	$a \in \mathbb{N} \cup \{+\infty\}$	$r \in \mathbb{R}$	$a \subseteq \mathbb{N}$ amount of “information”
finitary entity	$n \in \mathbb{N}$	$q \in \mathbb{Q}$	$e \subseteq_{\text{fin}} \mathbb{N}$ finite amount of “information”
approximation of infinitary by finitary	$n < +\infty$	$q \in \mathbb{Q}$ is “close” to $r \in \mathbb{R}$	$e \subseteq_{\text{fin}} a$
it works because	$\sup_{n \in \mathbb{N}} n = +\infty$	\mathbb{Q} is dense in \mathbb{R}	$\bigcup_{e \subseteq_{\text{fin}} a} e = a$

$$f : \mathcal{P}(\mathbb{N}) \rightarrow \mathcal{P}(\mathbb{N})$$

*For all finite portion of the output,
there is a finite portion of the input such that
the latter suffices to compute the former*

Let's make this rigorous: a topology over $\mathcal{P}(\mathbb{N})$

- 1 What do we mean by *computable*?
- 2 Let's make this rigorous: a topology over $\mathcal{P}(\mathbb{N})$
- 3 The Scott-topology of a poset
- 4 Encoding (continuous) functions as points
- 5 Interesting properties of continuous functions!
- 6 Summary, exercises, bibliography

Let's make this rigorous: a topology over $\mathcal{P}(\mathbb{N})$

The fastest course on topology ever:

Let's make this rigorous: a topology over $\mathcal{P}(\mathbb{N})$

The fastest course on topology ever:

A topology on a space X is the choice of special sets of points of X , called open sets, provide the set of points which can be approximated by each other

Continuous functions on X are those for which every set of possible outcomes that can be approximated by each other, gives rise to a set of points in the inputs that can be approximated by each other.

Let's make this rigorous: a topology over $\mathcal{P}(\mathbb{N})$

The fastest course on topology ever:

A topology on a space X is the choice of special sets of points of X , called open sets, provide the set of points which can be approximated by each other

Continuous functions on X are those for which every set of possible outcomes that can be approximated by each other, gives rise to a set of points in the inputs that can be approximated by each other.

*For all approximation d of $f(x)$,
there is an approximation e of x
such that $f(e)$ is a better approximation of $f(x)$ than d is*

Let's make this rigorous: a topology over $\mathcal{P}(\mathbb{N})$

The fastest course on topology ever:

A topology on a space X is the choice of special sets of points of X , called open sets, provide the set of points which can be approximated by each other

Continuous functions on X are those for which every set of possible outcomes that can be approximated by each other, gives rise to a set of points in the inputs that can be approximated by each other.

*For all open set O containing $f(x)$,
there is an open Q containing x
such that $fQ \subseteq O$*

Let's make this rigorous: a topology over $\mathcal{P}(\mathbb{N})$

The fastest course on topology ever:

A topology on a space X is the choice of special sets of points of X , called open sets, provide the set of points which can be approximated by each other

Continuous functions on X are those for which every set of possible outcomes that can be approximated by each other, gives rise to a set of points in the inputs that can be approximated by each other.

*For all open set O containing $f(x)$,
there is an open Q containing x
such that $Q \subseteq f^{-1}O$*

Let's make this rigorous: a topology over $\mathcal{P}(\mathbb{N})$

The fastest course on topology ever:

A topology on a space X is the choice of special sets of points of X , called open sets, provide the set of points which can be approximated by each other

Continuous functions on X are those for which every set of possible outcomes that can be approximated by each other, gives rise to a set of points in the inputs that can be approximated by each other.

$$O \text{ open} \Rightarrow f^{-1}O \text{ open}$$

Let's make this rigorous: a topology over $\mathcal{P}(\mathbb{N})$

A topology on a space X is the choice of special sets of points of X , called open sets, provide the set of points which can be approximated by each other

Continuous functions on X are those for which

$$O \text{ open} \Rightarrow f^{-1}O \text{ open}$$

We set $R := \mathcal{P}(\mathbb{N})$ and $R_{fin} := \mathcal{P}_{fin}(\mathbb{N})$. For $a \in R$, let $\uparrow a := \{b \in R \mid b \supseteq a\}$.

Definition

The open sets $O \subseteq R$ are the ones of shape $O = \bigcup_{e \in I} \uparrow e$, for some $I \subseteq R_{fin}$.

This is possible as $\{\uparrow e \subseteq R \mid e \subseteq_{fin} \mathbb{N}\} \subseteq \mathcal{P}(R)$ is a *base* for a topology on $\mathcal{P}(\mathbb{N})$.

Let's make this rigorous: a topology over $\mathcal{P}(\mathbb{N})$

A topology on a space X is the choice of special sets of points of X , called open sets, provide the set of points which can be approximated by each other

Continuous functions on X are those for which

$$O \text{ open} \Rightarrow f^{-1}O \text{ open}$$

We set $R := \mathcal{P}(\mathbb{N})$ and $R_{fin} := \mathcal{P}_{fin}(\mathbb{N})$. For $a \in R$, let $\uparrow a := \{b \in R \mid b \supseteq a\}$.

Definition

The open sets $O \subseteq R$ are the ones of shape $O = \bigcup_{e \in I} \uparrow e$, for some $I \subseteq R_{fin}$.

This is possible as $\{\uparrow e \subseteq R \mid e \subseteq_{fin} \mathbb{N}\} \subseteq \mathcal{P}(R)$ is a *base* for a topology on $\mathcal{P}(\mathbb{N})$.

Theorem

The open sets of our topology are exactly the $O \subseteq R$ which satisfy the following condition for all $a \in R$:

$$a \in O \Leftrightarrow O \ni e \subseteq_{fin} a, \text{ for some } e \in R.$$

Let's make this rigorous: a topology over $\mathcal{P}(\mathbb{N})$

Theorem

Let $f : R \rightarrow R$. The following are equivalent:

- ❶ f is continuous.
- ❷ f is monotone for \subseteq and, for all $a, d \in R$ we have

$$d \subseteq_{\text{fin}} f(a) \implies \text{there is } e \subseteq_{\text{fin}} a \text{ such that } d \subseteq f(e).$$

- ❸ for all $a \in R$ we have

$$f(a) = \bigcup_{e \subseteq_{\text{fin}} a} f(e).$$

Let's make this rigorous: a topology over $\mathcal{P}(\mathbb{N})$

Theorem

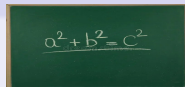
Let $f : R \rightarrow R$. The following are equivalent:

- ① f is continuous.
- ② f is monotone for \subseteq and, for all $a, d \in R$ we have

$$d \subseteq_{\text{fin}} f(a) \implies \text{there is } e \subseteq_{\text{fin}} a \text{ such that } d \subseteq f(e).$$

- ③ for all $a \in R$ we have

$$f(a) = \bigcup_{e \subseteq_{\text{fin}} a} f(e).$$



The Scott-topology of a poset

- 1 What do we mean by *computable*?
- 2 Let's make this rigorous: a topology over $\mathcal{P}(\mathbb{N})$
- 3 The Scott-topology of a poset
- 4 Encoding (continuous) functions as points
- 5 Interesting properties of continuous functions!
- 6 Summary, exercises, bibliography

The Scott-topology of a poset

A poset is the data of a set X and a partial order relation \leq (i.e. reflexive, antisymmetric and transitive).

The Scott-topology of a poset

A poset is the data of a set X and a partial order relation \leq (i.e. reflexive, antisymmetric and transitive).

Example

R with \subseteq is a poset.

The Scott-topology of a poset

A poset is the data of a set X and a partial order relation \leq (i.e. reflexive, antisymmetric and transitive).

A subset D of a poset X is called *directed* whenever $D \neq \emptyset$ and all $d, d' \in D$ admit an upper bound in D (i.e. there is $d'' \in D$ such that $d \leq d''$ and $d' \leq d''$).

Example

In R , the set $D = \{\{5, 2\}, \emptyset, \{8\}\}$ is not directed, because D contains $\{5, 2\}, \{8\}$ but no element of D is bigger than both $\{5, 2\}$ and $\{8\}$.

The Scott-topology of a poset

A poset is the data of a set X and a partial order relation \leq (i.e. reflexive, antisymmetric and transitive).

A subset D of a poset X is called *directed* whenever $D \neq \emptyset$ and all $d, d' \in D$ admit an upper bound in D (i.e. there is $d'' \in D$ such that $d \leq d''$ and $d' \leq d''$).

Example

In R , the set $D = \{e \in R \mid e \subseteq_{\text{fin}} \{5, 2\}\} = \{\emptyset, \{5, 2\}, \{5\}, \{2\}\}$ is directed (in this case it even has a maximum element).

The Scott-topology of a poset

A poset is the data of a set X and a partial order relation \leq (i.e. reflexive, antisymmetric and transitive).

A subset D of a poset X is called *directed* whenever $D \neq \emptyset$ and all $d, d' \in D$ admit an upper bound in D (i.e. there is $d'' \in D$ such that $d \leq d''$ and $d' \leq d''$).

Example

In R , given $a \in R$, the set $D = \{e \in R \mid e \subseteq_{\text{fin}} a\} =: \downarrow_{\text{fin}} a = \mathcal{P}_{\text{fin}}(a)$ is directed: the upper bound in D of $e, e' \in D$ is $e \cup e' \in D$ (in this case it is even their sup).

The Scott-topology of a poset

A poset is the data of a set X and a partial order relation \leq (i.e. reflexive, antisymmetric and transitive).

A subset D of a poset X is called *directed* whenever $D \neq \emptyset$ and all $d, d' \in D$ admit an upper bound in D (i.e. there is $d'' \in D$ such that $d \leq d''$ and $d' \leq d''$).

Example

In R , given $a \in R$, the set $D = \{e \in R \mid e \subseteq_{\text{fin}} a\} =: \downarrow_{\text{fin}} a = \mathcal{P}_{\text{fin}}(a)$ is directed: the upper bound in D of $e, e' \in D$ is $e \cup e' \in D$ (in this case it is even their sup).

Definition

Let (X, \leq) be a poset. Its *Scott-topology* is defined by declaring open the subsets U of X such that:

- ① U is upward closed (i.e. $U \ni x \leq y \Rightarrow U \ni y$)
- ② for all directed $D \subseteq X$ which admits a lub $\bigvee D \in U$, we have $D \cap U \neq \emptyset$.

The Scott-topology of a poset

A poset is the data of a set X and a partial order relation \leq (i.e. reflexive, antisymmetric and transitive).

A subset D of a poset X is called *directed* whenever $D \neq \emptyset$ and all $d, d' \in D$ admit an upper bound in D (i.e. there is $d'' \in D$ such that $d \leq d''$ and $d' \leq d''$).

Example

In R , given $a \in R$, the set $D = \{e \in R \mid e \subseteq_{\text{fin}} a\} =: \downarrow_{\text{fin}} a = \mathcal{P}_{\text{fin}}(a)$ is directed: the upper bound in D of $e, e' \in D$ is $e \cup e' \in D$ (in this case it is even their sup).

Definition

Let (X, \leq) be a poset. Its *Scott-topology* is defined by declaring open the subsets U of X such that:

- 1 U is upward closed (i.e. $U \ni x \leq y \Rightarrow U \ni y$)
- 2 for all directed $D \subseteq X$ which admits a lub $\bigvee D \in U$, we have $D \cap U \neq \emptyset$.

Example (Proposition)

Our topology on R coincides with the Scott-topology of the poset (R, \subseteq) .

The Scott-topology of a poset

In a poset (X, \leq) , we say that an element $e \in X$ is *compact* iff for all directed $D \subseteq X$ admitting $\bigvee D$, we have: if $e \leq \bigvee D$ then $e \leq d$ for some $d \in D$.

The Scott-topology of a poset

In a poset (X, \leq) , we say that an element $e \in X$ is *compact* iff for all directed $D \subseteq X$ admitting $\bigvee D$, we have: if $e \leq \bigvee D$ then $e \leq d$ for some $d \in D$.

Example (Proposition)

In (R, \subseteq) , the set of compact elements is R_{fin} .

The Scott-topology of a poset

In a poset (X, \leq) , we say that an element $e \in X$ is *compact* iff for all directed $D \subseteq X$ admitting $\bigvee D$, we have: if $e \leq \bigvee D$ then $e \leq d$ for some $d \in D$.

Example (Proposition)

In (R, \subseteq) , the set of compact elements is R_{fin} .

Theorem

Let X, Y be posets and $f : X \rightarrow Y$. The following are equivalent:

- 1 f is continuous wrt the Scott-topologies on X, Y .
- 2 For all directed $D \subseteq X$ admitting $\bigvee D$ in X , there is $\bigvee(fD)$ in Y and:

$$\bigvee(fD) = f(\bigvee D). \quad (\text{Scott-Continuity})$$

If moreover X, Y are “algebraic” (like R), then the above are equivalent to:

- 3 f is monotone (i.e. $a \leq b$ implies $f(a) \leq f(b)$) and, for all $a \in X$, $d \in Y$,
 $d \text{ compact} \leq f(a) \implies \text{there is } e \text{ compact} \leq a \text{ such that } d \leq f(e).$
- 4 For all $a \in X$ we have $f(a) = \bigvee_{e \text{ compact} \leq a} f(e)$

Encoding (continuous) functions as points

- 1 What do we mean by *computable*?
- 2 Let's make this rigorous: a topology over $\mathcal{P}(\mathbb{N})$
- 3 The Scott-topology of a poset
- 4 Encoding (continuous) functions as points
- 5 Interesting properties of continuous functions!
- 6 Summary, exercises, bibliography

Encoding (continuous) functions as points

“We can encode everything into \mathbb{N} !”

Encoding (continuous) functions as points

“We can encode everything into \mathbb{N} !”

$$\text{pair} : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}, \quad \text{pair}(n, m) := 2^n(2m + 1) - 1$$

$$\text{list} : \mathbb{N}^* \rightarrow \mathbb{N} \quad \text{list}([]) := 0 \quad \text{and} \quad \text{list}(n :: l) := 1 + \text{pair}(n, \text{list}(l))$$

$$\langle _, _ \rangle : \mathbb{N}^* \times \mathbb{N} \rightarrow \mathbb{N} \quad \langle l, n \rangle := \text{pair}(\text{list}(l), n)$$

Encoding (continuous) functions as points

“We can encode everything into \mathbb{N} !”

Corollary

We have a bijective encoding $\langle _, _ \rangle$ of $\mathbb{N}^ \times \mathbb{N}$ into \mathbb{N}*

Encoding (continuous) functions as points

“We can encode everything into \mathbb{N} !”

Corollary

We have a bijective encoding $\langle _, _ \rangle$ of $\mathbb{N}^ \times \mathbb{N}$ into \mathbb{N}*

Consider the following two auxiliary functions:

$$\begin{aligned} \text{set} : \mathbb{N}^* &\rightarrow R_{fin} & \text{set}(n_1, \dots, n_k) &:= \{n_1, \dots, n_k\} \\ \text{kl} : R &\rightarrow \mathcal{P}(\mathbb{N}^*) & \text{kl}(a) &:= \{l \in \mathbb{N}^* \mid \text{set}(l) \subseteq a\} \end{aligned}$$

Encoding (continuous) functions as points

“We can encode everything into \mathbb{N} !”

Corollary

We have a bijective encoding $\langle _, _ \rangle$ of $\mathbb{N}^ \times \mathbb{N}$ into \mathbb{N}*

Consider the following two auxiliary functions:

$$\begin{aligned} \text{set} : \mathbb{N}^* &\rightarrow R_{fin} & \text{set}(n_1, \dots, n_k) &:= \{n_1, \dots, n_k\} \\ \text{kl} : R &\rightarrow \mathcal{P}(\mathbb{N}^*) & \text{kl}(a) &:= \{l \in \mathbb{N}^* \mid \text{set}(l) \subseteq a\} \end{aligned}$$

Definition

$$\begin{aligned} \text{fun} : R &\rightarrow (R \Rightarrow R), & \text{fun}(a)(b) &:= \{n \in \mathbb{N} \mid \text{there is } l \in \text{kl}(b) \text{ s.t. } \langle l, n \rangle \in a\} \\ @ : R \times R &\rightarrow R, & @ &:= \text{uncurry}(\text{fun}) \quad (\text{i.e. } @(a, b) := \text{fun}(a)(b)) \\ \lambda : (R \Rightarrow R) &\rightarrow R, & \lambda(f) &:= \{\langle l, m \rangle \in \mathbb{N} \mid l \in \mathbb{N}^*, m \in f(\text{set}(l))\} \end{aligned}$$

Encoding (continuous) functions as points

“We can encode everything into \mathbb{N} !”

Corollary

We have a bijective encoding $\langle _, _ \rangle$ of $\mathbb{N}^ \times \mathbb{N}$ into \mathbb{N}*

Consider the following two auxiliary functions:

$$\begin{aligned} \text{set} : \mathbb{N}^* &\rightarrow R_{fin} & \text{set}(n_1, \dots, n_k) &:= \{n_1, \dots, n_k\} \\ \text{kl} : R &\rightarrow \mathcal{P}(\mathbb{N}^*) & \text{kl}(a) &:= \{l \in \mathbb{N}^* \mid \text{set}(l) \subseteq a\} \end{aligned}$$

Definition

$$\begin{aligned} \text{fun} : R &\rightarrow (R \Rightarrow R), & \text{fun}(a)(b) &:= \{n \in \mathbb{N} \mid \text{there is } l \in \text{kl}(b) \text{ s.t. } \langle l, n \rangle \in a\} \\ @ : R \times R &\rightarrow R, & @ &:= \text{uncurry}(\text{fun}) \quad (\text{i.e. } @(a, b) := \text{fun}(a)(b)) \\ \lambda : (R \Rightarrow R) &\rightarrow R, & \lambda(f) &:= \{\langle l, m \rangle \in \mathbb{N} \mid l \in \mathbb{N}^*, m \in f(\text{set}(l))\} \end{aligned}$$

Theorem

We indeed have $\text{Im}(\text{fun}) \subseteq (R \Rightarrow R)$.

Encoding (continuous) functions as points

“We can encode everything into \mathbb{N} !”

Corollary

We have a bijective encoding $\langle _, _ \rangle$ of $\mathbb{N}^* \times \mathbb{N}$ into \mathbb{N}

Consider the following two auxiliary functions:

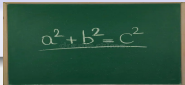
$$\begin{aligned} \text{set} : \mathbb{N}^* &\rightarrow R_{fin} & \text{set}(n_1, \dots, n_k) &:= \{n_1, \dots, n_k\} \\ \text{kl} : R &\rightarrow \mathcal{P}(\mathbb{N}^*) & \text{kl}(a) &:= \{l \in \mathbb{N}^* \mid \text{set}(l) \subseteq a\} \end{aligned}$$

Definition

$$\begin{aligned} \text{fun} : R &\rightarrow (R \Rightarrow R), & \text{fun}(a)(b) &:= \{n \in \mathbb{N} \mid \text{there is } l \in \text{kl}(b) \text{ s.t. } \langle l, n \rangle \in a\} \\ @ : R \times R &\rightarrow R, & @ &:= \text{uncurry}(\text{fun}) \quad (\text{i.e. } @(a, b) := \text{fun}(a)(b)) \\ \lambda : (R \Rightarrow R) &\rightarrow R, & \lambda(f) &:= \{\langle l, m \rangle \in \mathbb{N} \mid l \in \mathbb{N}^*, m \in f(\text{set}(l))\} \end{aligned}$$

Theorem

We indeed have $\text{Im}(\text{fun}) \subseteq (R \Rightarrow R)$.


$$a^2 + b^2 = c^2$$

Encoding (continuous) functions as points

Theorem

- $@$, fun and λ are continuous.
- The pair (λ, fun) defines a topological retraction of R onto $\text{Im}(\lambda) \approx (R \Rightarrow R)$, i.e.

$$\text{fun} \circ \lambda = \text{id}_{R \Rightarrow R}. \quad (\beta)$$

- For all $f : R \Rightarrow R$ we have:

$$\lambda(\text{fun}(f)) \supseteq f \quad \text{and} \quad \lambda(f) = \bigcup_{\text{fun}(a)=f} a.$$

Encoding (continuous) functions as points

Theorem

- $@$, fun and λ are continuous.
- The pair (λ, fun) defines a topological retraction of R onto $\text{Im}(\lambda) \approx (R \Rightarrow R)$, i.e.

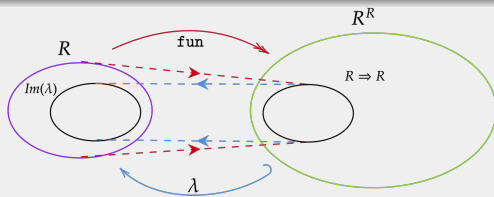
$$\text{fun} \circ \lambda = \text{id}_{R \Rightarrow R}. \quad (\beta)$$

- For all $f : R \Rightarrow R$ we have:

$$\lambda(\text{fun}(f)) \supseteq f \quad \text{and} \quad \lambda(f) = \bigcup_{\text{fun}(a)=f} a.$$

Definition

The structure $(R, \lambda, @)$ is called the *graph model*.



Encoding (continuous) functions as points

Theorem

$$a^2 + b^2 = c^2$$

- $@$, fun and λ are continuous.
- The pair (λ, fun) defines a topological retraction of R onto $\text{Im}(\lambda) \approx (R \Rightarrow R)$, i.e.

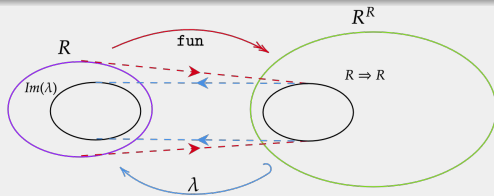
$$\text{fun} \circ \lambda = \text{id}_{R \Rightarrow R}. \quad (\beta)$$

- For all $f : R \Rightarrow R$ we have:

$$\lambda(\text{fun}(f)) \supseteq f \quad \text{and} \quad \lambda(f) = \bigcup_{\text{fun}(a)=f} a.$$

Definition

The structure $(R, \lambda, @)$ is called the *graph model*.



Interesting properties of continuous functions!

- 1 What do we mean by *computable*?
- 2 Let's make this rigorous: a topology over $\mathcal{P}(\mathbb{N})$
- 3 The Scott-topology of a poset
- 4 Encoding (continuous) functions as points
- 5 Interesting properties of continuous functions!
- 6 Summary, exercises, bibliography

Interesting properties of continuous functions!

All continuous $f : R \rightarrow R$ admit fixed points.

The function

$$Y : (R \Rightarrow R) \rightarrow R \quad Y(f) := @(\Delta_f, \Delta_f)$$

where $\Delta_f := \lambda(f \circ @ \circ \delta) \in R$ and $\delta : R \rightarrow R \times R$ is the diagonal, is a *fixed point combinator*, i.e. for all $f : R \Rightarrow R$, we have

$$f(Y(f)) = Y(f).$$

Interesting properties of continuous functions!

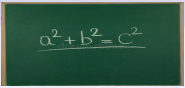
All continuous $f : R \rightarrow R$ admit fixed points.

The function

$$Y : (R \Rightarrow R) \rightarrow R \quad Y(f) := @(\Delta_f, \Delta_f)$$

where $\Delta_f := \lambda(f \circ @ \circ \delta) \in R$ and $\delta : R \rightarrow R \times R$ is the diagonal, is a *fixed point combinator*, i.e. for all $f : R \Rightarrow R$, we have

$$f(Y(f)) = Y(f).$$

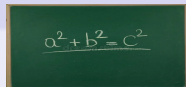

$$a^2 + b^2 = c^2$$

Interesting properties of continuous functions!

All continuous $f : R \rightarrow R$ admit fixed points.

The function

$$Y : (R \Rightarrow R) \rightarrow R \quad Y(f) := @(\Delta_f, \Delta_f)$$



where $\Delta_f := \lambda(f \circ @ \circ \delta) \in R$ and $\delta : R \rightarrow R \times R$ is the diagonal, is a *fixed point combinator*, i.e. for all $f : R \Rightarrow R$, we have

$$f(Y(f)) = Y(f).$$

Theorem

The set of RE sets is closed wrt the following rules:

$$\overline{\lambda(\lambda \circ \text{curry}(\cdots (\lambda \circ \text{curry}(\text{proj}_i^n)) \cdots))}$$

$$\frac{f : R \Rightarrow R \text{ computable}}{\lambda(f)}$$

$$\frac{a \quad b}{@(a, b)}$$

Summary, exercises, bibliography

- 1 What do we mean by *computable*?
- 2 Let's make this rigorous: a topology over $\mathcal{P}(\mathbb{N})$
- 3 The Scott-topology of a poset
- 4 Encoding (continuous) functions as points
- 5 Interesting properties of continuous functions!
- 6 Summary, exercises, bibliography

Summary, exercises, bibliography

- What do we intuitively mean when we say that a function is **computable**
- That this relates to **topology**
- That $R = (\mathcal{P}(\mathbb{N}), \text{Scott})$ is a good **topological space** for modeling this
- That its topology only depends on the **partial order** \subseteq
 - That actually, Scott-topology and Scott-continuity themselves are a property about “**approximations**” in posets
 - That Scott-continuous functions are given by their restriction on the **finite elements** of R . Also, they **embed into** their base space. This is done via the **retraction** (λ, fun) , i.e. equation (β)
 - That all Scott-continuous functions on R have **fixed points**
 - That λ produces **RE sets** and fun preserves them



- Do the proofs of the statements in the slides
- Look at **our notes** on the [webpage of the course](#), there are plenty of **exercises**
- The exercises have **solutions** (but try to do them by yourself before looking at them!)

Summary, exercises, bibliography

- Chapter 1 and Section 1 of Chapter 18 of:
Henk P. Barendregt, The lambda-calculus, its syntax and semantics, 1984,
<https://www.sciencedirect.com/bookseries/studies-in-logic-and-the-foundations-of-mathematics/vol/103>
- Domain theory chapter of:
Handbook of logic in computer science: semantic structures. 1995,
<https://dl.acm.org/doi/book/10.5555/218742> (an updated version is available [here](#))
- Chapter 1 of:
Amadio R., Curien P-L, Domains and lambda-calculi, 1996,
<https://www.cambridge.org/core/books/domains-and-lambda-calculi/4C6AB6938E436CFA8D5A8533B76A7F23>
- Chapter 1, 5 of:
PhD thesis of Giulio Manzonetto, Models and theories of lambda calculus, 2008,
<https://www.irif.fr/~gmanzone/ManzonettoPhdThesis.pdf>
- This [conference](#) of **Dana Scott**