The λ-calculus, from intuitionistic to classical logic

Webpage of the course

Davide Barbarossa
db2437@bath.ac.uk
Dept of Computer Science
UNIVERSITY OF BATH

Giulio Guerrieri
g.guerrieri@sussex.ac.uk
Dept of Computer Science
UNIVERSITY OF SUSSEX

ESSLLI Summer School, Bochum (Germany) *28/07/2025 – 01/08/2025*

The λ-calculus, from intuitionistic 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 intuitionistic logic
Giulio Guerrieri
Lecture 5: Krivine's approach to classical logic
Davide Barbarossa

The λ-calculus,
from intuitionistic 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
UNIVERSITY OF BATH

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 $\sqrt{54899321104}$

$$
\begin{aligned}
x_0 &:= \textit{any positive number} \\
x_{n+1} &:= \frac{1}{2}\left(x_n + \frac{54899321104}{x_n}\right)
\end{aligned}
$$

What do we mean by *computable*?

Problem: Compute $\sqrt{54899321104}$

$$
\begin{aligned}
x_0 &:= \textit{any positive number} \\
x_{n+1} &:= \frac{1}{2}\left(x_n + \frac{54899321104}{x_n}\right)
\end{aligned}
$$

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

What do we mean by *computable*?

Problem: Compute $\sqrt{54899321104}$

$$
\begin{aligned}
x_0 &:= \textit{any positive number} \\
x_{n+1} &:= \frac{1}{2}\left(x_n + \frac{54899321104}{x_n}\right)
\end{aligned}
$$

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

$\overset{\text{Heron}}{\cancel{\text{We}}}$ defined a function

$$H : \mathbb{N} \to \mathbb{R}, \qquad H(n) := x_n$$

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

What do we mean by *computable*?

Problem: Compute $\sqrt{54899321104}$

$$\begin{aligned} x_0 &:= \textit{any positive number} \\ x_{n+1} &:= \frac{1}{2}\left(x_n + \frac{54899321104}{x_n}\right) \end{aligned}$$

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

$\overset{\text{Heron}}{\cancel{\text{We}}}$ defined a function

$$H : \mathbb{N} \cup \{+\infty\} \to \mathbb{R}, \qquad \begin{aligned} H(n) &:= x_n \\ H(+\infty) &:= \sqrt{54899321104} \end{aligned}$$

such that

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

What do we mean by *computable*?

Problem: Compute all the prime numbers

What do we mean by *computable*?

Problem: Compute all the prime numbers

Problem: Compute all the prime numbers

Sieve of Eratosthenes!

What do we mean by *computable*?

Problem: Compute all the prime numbers

Sieve of Eratosthenes!

$\overset{\text{Eratosthenes}}{\cancel{\text{We}}}$ defined a function

$$E : \mathbb{N} \to \mathcal{P}(\mathbb{N}), \qquad E(n) := \textit{set of the first } n \textit{ prime numbers}$$

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!

$\overset{\text{Eratosthenes}}{\cancel{\text{We}}}$ defined a function

$$E : \mathbb{N} \cup \{+\infty\} \to \mathcal{P}(\mathbb{N}), \qquad \begin{array}{lcl} E(n) & := & \textit{set of the first } n \textit{ prime numbers} \\ E(+\infty) & := & \textit{set of all the prime numbers} \end{array}$$

such that

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

Why do we say that H and E are computable?
What property do the 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 the 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 the 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 $\mathbb{R}$

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

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 the 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	$\sup\limits_{n \in \mathbb{N}} n = +\infty$	$\mathbb{Q}$ is dense in $\mathbb{R}$	$\bigcup\limits_{e \subseteq_{\text{fin}} a} e = a$

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

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 the 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 $\mathbb{R}$	$\bigcup_{e \subseteq_{\text{fin}} a} e = a$

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

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 the 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 the 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 the 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 $\mathbb{R}$	$\bigcup_{e \subseteq_{\text{fin}} a} e = a$

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

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 the 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	$\sup\limits_{n \in \mathbb{N}} n = +\infty$	$\mathbb{Q}$ is dense in $\mathbb{R}$	$\bigcup\limits_{e \subseteq_{\text{fin}} a} e = a$	

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

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

The fastest course on topology ever:

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.

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

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 $fQ \subseteq O$

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 $Q \subseteq f^{-1}O$

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.

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

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

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

For today and tomorrow, R means $\mathcal{P}(\mathbb{N})$. For $a \in R$, let $\uparrow a := \{b \in R \mid b \supseteq a\}$.

Definition

Let us take as open sets $O \subseteq R$ the ones of shape $O = \bigcup_{e \in I} \uparrow e$, for some $I \subseteq R_{\mathit{fin}}$.

This is possible as $\{\uparrow e \subseteq R \mid e \subseteq_{\mathit{fin}} \mathbb{N}\} \subseteq \mathcal{P}(R)$ 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

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

For today and tomorrow, R means $\mathcal{P}(\mathbb{N})$. For $a \in R$, let $\uparrow a := \{b \in R \mid b \supseteq a\}$.

Definition

Let us take as open sets $O \subseteq R$ 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 $a \in R$:

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

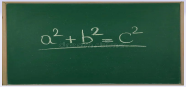
Let's make this rigorous: a topology over $\mathcal{P}(\mathbb{N})$

Theorem

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

1. *f is continuous.*
2. *f is monotone for $\subseteq$ and, for all $a, d \in R$ we have*
$$d \subseteq_{\text{fin}} f(a) \quad \Longrightarrow \quad \text{there is } \; e \subseteq_{\text{fin}} a \; \text{ such that } \; d \subseteq f(e).$$
3. *for all $a \in R$ we have*
$$f(a) = \bigcup_{e \subseteq_{fin} a} f(e).$$

Theorem



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

1. *f is continuous.*
2. *f is monotone for $\subseteq$ and, for all $a, d \in R$ we have*
$$d \subseteq_{\text{fin}} f(a) \quad \Longrightarrow \quad \text{there is } \ e \subseteq_{\text{fin}} a \ \text{ such that } \ d \subseteq f(e).$$
3. *for all $a \in R$ we have*
$$f(a) = \bigcup_{e \subseteq_{fin} a} f(e).$$

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 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.

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 a common upper bound in 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 $\{8\}$.

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 a common upper bound in 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).

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 a common upper bound in 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).

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 a common upper bound in D.

Example

In R, given $a \in R$, the set $D = \{e \in R \mid e \subseteq_{\mathrm{fin}} a\} =: \downarrow_{\mathrm{fin}} a = \mathcal{P}_{\mathrm{fin}}(a)$ 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 $\bigvee D \in U$, 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 a common upper bound in 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 $\bigvee D \in U$, 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 $d \in D$.

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 $f : X \to Y$. 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:*

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

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

3. *f is monotone for $\leq$ and, for all $a \in X$ and $d \in Y$, we have*

$$d \text{ compact } \leq f(a) \quad \Longrightarrow \quad \text{there is } e \text{ compact } \leq a \text{ such that } d \leq f(e).$$

4. *For all $a \in X$ we have $f(a) = \bigvee_{e \text{ compact} \leq a} f(e)$*

Encoding (continuous) functions as points

"*We can encode everything into* $\mathbb{N}$*!* "

Encoding (continuous) functions as points

"*We can encode everything into* $\mathbb{N}$*!*"

$$\mathrm{pair} : \mathbb{N} \times \mathbb{N} \to \mathbb{N}, \qquad \mathrm{pair}(n, m) := 2^n(2m+1) - 1$$

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

$$\langle \_, \_ \rangle : \mathbb{N}^* \times \mathbb{N} \to \mathbb{N} \qquad \langle l, n \rangle := \mathrm{pair}(\mathrm{list}(l), n)$$

"*We can encode everything into* $\mathbb{N}$*!*"

Corollary

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

“*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:

$$\begin{array}{ll} \mathrm{set} : \mathbb{N}^* \to R_{\mathit{fin}} & \mathrm{set}(n_1, \ldots, n_k) := \{n_1, \ldots, n_k\} \\ \mathrm{kl} : R \to \mathcal{P}(\mathbb{N}^*) & \mathrm{kl}(a) := \{l \in \mathbb{N}^* \mid \mathrm{set}(l) \subseteq a\} \end{array}$$

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:

$$
\begin{array}{ll}
\mathrm{set} : \mathbb{N}^* \to R_{\mathit{fin}} & \mathrm{set}(n_1, \ldots, n_k) := \{n_1, \ldots, n_k\} \\
\mathrm{kl} : R \to \mathcal{P}(\mathbb{N}^*) & \mathrm{kl}(a) := \{l \in \mathbb{N}^* \mid \mathrm{set}(l) \subseteq a\}
\end{array}
$$

Definition

$$
\begin{array}{ll}
\mathrm{fun} : R \to (R \Rightarrow R), & \mathrm{fun}(a)(b) := \{n \in \mathbb{N} \mid \textit{ there is } l \in \mathrm{kl}(b) \textit{ s.t. } \langle l, n\rangle \in a\} \\
@ : R \times R \to R, & @ := \mathrm{uncurry}(\mathrm{fun}) \\
\lambda : (R \Rightarrow R) \to R, & \lambda(f) := \{\langle l, m\rangle \in \mathbb{N} \mid l \in \mathbb{N}^*,\ m \in f(\mathrm{set}(l))\}
\end{array}
$$

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:

$$\begin{array}{ll} \mathrm{set} : \mathbb{N}^* \to R_{\mathit{fin}} & \mathrm{set}(n_1, \ldots, n_k) := \{n_1, \ldots, n_k\} \\ \mathrm{kl} : R \to \mathcal{P}(\mathbb{N}^*) & \mathrm{kl}(a) := \{l \in \mathbb{N}^* \mid \mathrm{set}(l) \subseteq a\} \end{array}$$

Definition

$$\mathrm{fun} : R \to (R \Rightarrow R), \quad \mathrm{fun}(a)(b) := \{n \in \mathbb{N} \mid \text{ there is } l \in \mathrm{kl}(b) \text{ s.t. } \langle l, n \rangle \in a\}$$

$$@ : R \times R \to R, \quad @ := \mathrm{uncurry}(\mathrm{fun})$$

$$\lambda : (R \Rightarrow R) \to R, \quad \lambda(f) := \{\langle l, m \rangle \in \mathbb{N} \mid l \in \mathbb{N}^*, \, m \in f(\mathrm{set}(l))\}$$

Theorem

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

"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:

$$\begin{array}{ll} \mathrm{set} : \mathbb{N}^* \to R_{fin} & \mathrm{set}(n_1, \ldots, n_k) := \{n_1, \ldots, n_k\} \\ \mathrm{kl} : R \to \mathcal{P}(\mathbb{N}^*) & \mathrm{kl}(a) := \{l \in \mathbb{N}^* \mid \mathrm{set}(l) \subseteq a\} \end{array}$$

Definition

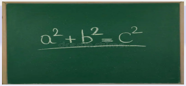
$$\mathrm{fun} : R \to (R \Rightarrow R), \quad \mathrm{fun}(a)(b) := \{n \in \mathbb{N} \mid \textit{ there is } l \in \mathrm{kl}(b) \textit{ s.t. } \langle l, n \rangle \in a\}$$

$$@ : R \times R \to R, \quad @ := \mathrm{uncurry}(\mathrm{fun})$$

$$\lambda : (R \Rightarrow R) \to R, \quad \lambda(f) := \{\langle l, m \rangle \in \mathbb{N} \mid l \in \mathbb{N}^*,\, m \in f(\mathrm{set}(l))\}$$

Theorem

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



Encoding (continuous) functions as points

Theorem

- *@, fun and λ are continuous.*
- *The pair (λ, fun) defines a topological retraction of R onto $\mathrm{Im}(\lambda) \approx (R \Rightarrow R)$, i.e.*
$$\mathrm{fun} \circ \lambda = \mathrm{id}_{R \Rightarrow R}. \tag{β}$$
- *For all $f : R \Rightarrow R$ we have:*
$$\lambda(\mathrm{fun}(f)) \supseteq f \quad \textit{and} \quad \lambda(f) = \bigcup_{\mathrm{fun}(a)=f} a.$$

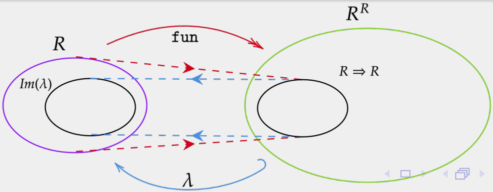
Encoding (continuous) functions as points

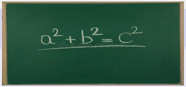
Theorem

- @, fun *and* λ *are continuous.*
- *The pair* (λ, fun) *defines a topological retraction of* R *onto* $\mathrm{Im}(\lambda) \approx (R \Rightarrow R)$, *i.e.*
$$\mathrm{fun} \circ \lambda = \mathrm{id}_{R \Rightarrow R}. \tag{β}$$
- *For all* $f : R \Rightarrow R$ *we have:*
$$\lambda(\mathrm{fun}(f)) \supseteq f \quad \textit{and} \quad \lambda(f) = \bigcup_{\mathrm{fun}(a) = f} a.$$

Definition

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





Theorem

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

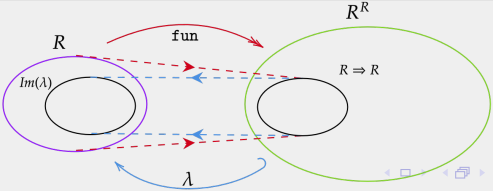
$$\mathrm{fun} \circ \lambda = \mathrm{id}_{R \Rightarrow R}. \tag{β}$$

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

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

Definition

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



Interesting properties of continuous functions!

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

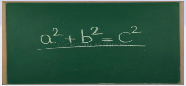
The function

$$Y : (R \Rightarrow R) \to R \qquad Y(f) := @(\Delta_f, \Delta_f)$$

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

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

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



The function

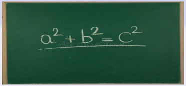
$$Y : (R \Rightarrow R) \to R \qquad Y(f) := @(\Delta_f, \Delta_f)$$

where $\Delta_f := \lambda(f \circ @ \circ \delta) \in R$ and $\delta : R \to R \times R$ 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 $f : R \to R$ admit fixed points.



The function

$$Y : (R \Rightarrow R) \to R \qquad Y(f) := @(\Delta_f, \Delta_f)$$

where $\Delta_f := \lambda(f \circ @ \circ \delta) \in R$ and $\delta : R \to R \times R$ 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:

$$\frac{}{\lambda(\lambda \circ \mathrm{curry}(\cdots(\lambda \circ \mathrm{curry}(\mathrm{proj}_i^n))\cdots))}$$

$$\frac{f : R \Rightarrow R \text{ and computable}}{\lambda(f)} \qquad \frac{a \qquad b}{@(a, b)}$$

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

- What do we intuitively mean when we say that a function is computable
- That this relates to topology
- That $R = (\mathcal{P}(\mathbb{N}), Scott)$ is a good topological space for modeling this
- That its topology only depends from 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

Some exercises and details

- Do the proofs of the statements in the slides
- Look at my **notes** on the webpage of the course: there are plenty more **exercises** and **details**
- The exercises have **solutions** (but try to do them by yourself before looking at them!)

Some (classic) bibliography to go further

- 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-lambdacalculi/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, 2019**. The first half is an interesting historical background, and the second half includes what we saw today