The λ-Calculus, from Minimal to Classical Logic

Webpage of the course

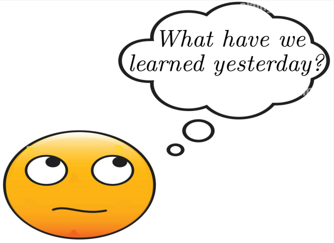
Davide Barbarossa
db2437@bath.ac.uk
Dept of Computer Science

Giulio Guerrieri
g.guerrieri@sussex.ac.uk
Dept of Ìnformatics

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

Previously...

- We introduced the λ-calculus, a basic functional programming language inspired by the graph model.
- We gave it a denotational semantics in the graph model.
- We gave it a operational semantics.
- Even if minimal, the operational semantics makes it a Turing-complete programming language.
- We programmed some basic functions on simple datatypes.



The λ-Calculus,
from Minimal to Classical Logic

Lecture 3:

Category Theory for Denotational Semantics

Read the notes: they are full of details, proofs, explanations, exercises, bibliography!

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

ESSLLI Summer School, Bochum (Germany) *30/07/2025*

Outline

1. What is Denotational Semantics for Programming Languages?
2. Category Theory in a Nutshell
3. Categorical Semantics for the (Untyped) λ-Calculus
4. Summary, Exercises, Bibliography

1. What is Denotational Semantics for Programming Languages?
2. Category Theory in a Nutshell
3. Categorical Semantics for the (Untyped) λ-Calculus
4. Summary, Exercises, Bibliography

What is Denotational Semantics for Programming Languages?

Denotational semantics describes the *meaning* of programs via mathematical objects.

Idea: The denotation of a program π describes *what* π does, regardless of *how*.

What is Denotational Semantics for Programming Languages?

Denotational semantics describes the *meaning* of programs via mathematical objects.

Idea: The denotation of a program π describes *what* π does, regardless of *how*.

Some tenets of denotational semantics $[\![\pi]\!]$ of a program π:

1. Contextuality: If $[\![\pi]\!] = [\![\pi']\!]$ then $[\![\mathtt{C}\langle\pi\rangle]\!] = [\![\mathtt{C}\langle\pi'\rangle]\!]$ for every context $\mathtt{C}$.
   $\rightsquigarrow$ The denotation of a program is built from the denotations of its subprograms.

What is Denotational Semantics for Programming Languages?

Denotational semantics describes the *meaning* of programs via mathematical objects.

Idea: The denotation of a program π describes *what* π does, regardless of *how*.

Some tenets of denotational semantics $[\![\pi]\!]$ of a program π:

1. Contextuality: If $[\![\pi]\!] = [\![\pi']\!]$ then $[\![\mathtt{C}\langle\pi\rangle]\!] = [\![\mathtt{C}\langle\pi'\rangle]\!]$ for every context $\mathtt{C}$.
   $\rightsquigarrow$ The denotation of a program is built from the denotations of its subprograms.

2. Invariance under evaluation: If $\pi \to_\beta \pi'$ then $[\![\pi]\!] = [\![\pi']\!]$.
   $\rightsquigarrow$ The denotation is invariant under evaluation, the interest is *what*, not *how*.

What is Denotational Semantics for Programming Languages?

Denotational semantics describes the *meaning* of programs via mathematical objects.

Idea: The denotation of a program π describes *what* π does, regardless of *how*.

Some tenets of denotational semantics $[\![\pi]\!]$ of a program π:

1. Contextuality: If $[\![\pi]\!] = [\![\pi']\!]$ then $[\![\mathtt{C}\langle\pi\rangle]\!] = [\![\mathtt{C}\langle\pi'\rangle]\!]$ for every context $\mathtt{C}$.
   $\leadsto$ The denotation of a program is built from the denotations of its subprograms.
2. Invariance under evaluation: If $\pi \to_\beta \pi'$ then $[\![\pi]\!] = [\![\pi']\!]$.
   $\leadsto$ The denotation is invariant under evaluation, the interest is *what*, not *how*.
3. Consistence: There are programs π and π' such that $[\![\pi]\!] \neq [\![\pi']\!]$.
   $\leadsto$ The denotation of a program is informative, it does not collapse everything.

What is Denotational Semantics for Programming Languages?

In which kind of algebraic structures can λ-terms be denoted/interpreted? A λ-term

1. can serve as an *argument* $\rightsquigarrow$ it should be denoted in a algebraic structure D;
2. can serve as a *function* to apply to an argument $\rightsquigarrow$ it should be denoted in $D \Rightarrow D$.

As a λ-term $\mathtt{M}$ can be applied to itself, it is natural to ask for $D \simeq D \Rightarrow D$.

What is Denotational Semantics for Programming Languages?

In which kind of algebraic structures can λ-terms be denoted/interpreted? A λ-term

1. can serve as an *argument* $\rightsquigarrow$ it should be denoted in a algebraic structure D;
2. can serve as a *function* to apply to an argument $\rightsquigarrow$ it should be denoted in $D \Rightarrow D$.

As a λ-term $\mathtt{M}$ can be applied to itself, it is natural to ask for $D \simeq D \Rightarrow D$.

- By Cantor's theorem, D cannot be a set where $D \Rightarrow D$ is its function space.
- Restricting $D \Rightarrow D$ to the set of some *specific* maps, it is possible that $D \simeq D \Rightarrow D$.
  Ex. D is a cpo, $D \Rightarrow D$ is the set of Scott-continuous maps on D ordered pointwise.

In which kind of algebraic structures can λ-terms be denoted/interpreted? A λ-term

1. can serve as an *argument* $\leadsto$ it should be denoted in a algebraic structure D;
2. can serve as a *function* to apply to an argument $\leadsto$ it should be denoted in $D \Rightarrow D$.

As a λ-term `M` can be applied to itself, it is natural to ask for $D \simeq D \Rightarrow D$.

- By Cantor's theorem, D cannot be a set where $D \Rightarrow D$ is its function space.
- Restricting $D \Rightarrow D$ to the set of some *specific* maps, it is possible that $D \simeq D \Rightarrow D$.
  Ex. D is a cpo, $D \Rightarrow D$ is the set of Scott-continuous maps on D ordered pointwise.

Currying: it is natural to require that $(D \times D) \Rightarrow D \simeq D \Rightarrow (D \Rightarrow D)$ (recall λ and @).
... (many other requirements/desiderata)

What is Denotational Semantics for Programming Languages?

In which kind of algebraic structures can λ-terms be denoted/interpreted? A λ-term

1. can serve as an *argument* $\rightsquigarrow$ it should be denoted in a algebraic structure D;
2. can serve as a *function* to apply to an argument $\rightsquigarrow$ it should be denoted in $D \Rightarrow D$.

As a λ-term $\mathtt{M}$ can be applied to itself, it is natural to ask for $D \simeq D \Rightarrow D$.

- By Cantor's theorem, D cannot be a set where $D \Rightarrow D$ is its function space.
- Restricting $D \Rightarrow D$ to the set of some *specific* maps, it is possible that $D \simeq D \Rightarrow D$.
  Ex. D is a cpo, $D \Rightarrow D$ is the set of Scott-continuous maps on D ordered pointwise.

Currying: it is natural to require that $(D \times D) \Rightarrow D \simeq D \Rightarrow (D \Rightarrow D)$ (recall λ and @).
... (many other requirements/desiderata)

Question: Are there some abstract conditions that, when satisfied by D, guarantee that D is a denotational model for the untyped λ-calculus with all/some desiderata?
Question: Where should these algebraic structures D live as a general setting?

What is Denotational Semantics for Programming Languages?

In which kind of algebraic structures can λ-terms be denoted/interpreted? A λ-term

1. can serve as an *argument* $\rightsquigarrow$ it should be denoted in a algebraic structure D;
2. can serve as a *function* to apply to an argument $\rightsquigarrow$ it should be denoted in $D \Rightarrow D$.

As a λ-term `M` can be applied to itself, it is natural to ask for $D \simeq D \Rightarrow D$.

- By Cantor's theorem, D cannot be a set where $D \Rightarrow D$ is its function space.
- Restricting $D \Rightarrow D$ to the set of some *specific* maps, it is possible that $D \simeq D \Rightarrow D$.
  Ex. D is a cpo, $D \Rightarrow D$ is the set of Scott-continuous maps on D ordered pointwise.

Currying: it is natural to require that $(D \times D) \Rightarrow D \simeq D \Rightarrow (D \Rightarrow D)$ (recall λ and @).
... (many other requirements/desiderata)

Question: Are there some abstract conditions that, when satisfied by D, guarantee that D is a denotational model for the untyped λ-calculus with all/some desiderata?
Question: Where should these algebraic structures D live as a general setting?

Answer: To be as general as possible, let us have a quick look at category theory!

1 What is Denotational Semantics for Programming Languages?

2 Category Theory in a Nutshell

3 Categorical Semantics for the (Untyped) λ-Calculus

4 Summary, Exercises, Bibliography

A category $\boldsymbol{C}$ is given by:

- a collection of *objects*;
- for each pair of objects A and B, a collection $\boldsymbol{C}(A, B)$ of *morphisms* (aka *arrows*);
- for each triple A, B, C of objects, a *composition* operation $\circ\colon \boldsymbol{C}(B, C) \times \boldsymbol{C}(A, B) \to \boldsymbol{C}(A, C)$ satisfying *associativity*, i.e.

$$f \circ (g \circ h) = (f \circ g) \circ h \qquad \text{for all } f \in \boldsymbol{C}(C, D),\ g \in \boldsymbol{C}(B, C),\ h \in \boldsymbol{C}(A, B);$$

- for every object A, a morphism $\mathrm{id}_A \in \boldsymbol{C}(A, A)$, called *identity* on A, such that

$$\mathrm{id}_B \circ f = f = f \circ \mathrm{id}_A \qquad \text{for all } f \in \boldsymbol{C}(A, B).$$

A category $\boldsymbol{C}$ is given by:

- a collection of *objects*;
- for each pair of objects A and B, a collection $\boldsymbol{C}(A, B)$ of *morphisms* (aka *arrows*);
- for each triple A, B, C of objects, a *composition* operation $\circ \colon \boldsymbol{C}(B, C) \times \boldsymbol{C}(A, B) \to \boldsymbol{C}(A, C)$ satisfying *associativity*, i.e.
$$f \circ (g \circ h) = (f \circ g) \circ h \qquad \text{for all } f \in \boldsymbol{C}(C, D),\ g \in \boldsymbol{C}(B, C),\ h \in \boldsymbol{C}(A, B);$$
- for every object A, a morphism $\mathrm{id}_A \in \boldsymbol{C}(A, A)$, called *identity* on A, such that
$$\mathrm{id}_B \circ f = f = f \circ \mathrm{id}_A \qquad \text{for all } f \in \boldsymbol{C}(A, B).$$

A category $\boldsymbol{C}$ is *large* or *small* depending on whether its collection of objects is a proper class or a set, respectively.

A large category $\boldsymbol{C}$ is *locally small* if $\boldsymbol{C}(A, B)$ is a set, for every pair of objects A, B. Under this hypothesis (which holds in our examples), $\boldsymbol{C}(A, B)$ is also called a hom-set.

Category Theory in a Nutshell

A category $\boldsymbol{C}$ is given by:

- a collection of *objects*;
- for each pair of objects A and B, a collection $\boldsymbol{C}(A,B)$ of *morphisms* (aka *arrows*);
- for each triple A,B,C of objects, a *composition* operation $\circ\colon \boldsymbol{C}(B,C)\times\boldsymbol{C}(A,B)\to\boldsymbol{C}(A,C)$ satisfying *associativity*, i.e.
$$f\circ(g\circ h)=(f\circ g)\circ h \qquad \text{for all } f\in\boldsymbol{C}(C,D),\ g\in\boldsymbol{C}(B,C),\ h\in\boldsymbol{C}(A,B);$$
- for every object A, a morphism $\mathrm{id}_A\in\boldsymbol{C}(A,A)$, called *identity* on A, such that
$$\mathrm{id}_B\circ f=f=f\circ\mathrm{id}_A \qquad \text{for all } f\in\boldsymbol{C}(A,B).$$

Notation: $f\colon A\to B$ stands for $f\in\boldsymbol{C}(A,B)$, if the category $\boldsymbol{C}$ is unambiguously clear.

Rmk: Objects may not be sets. Morphisms may not be functions.

Notation: Equalities among morphisms (like $f\circ g=h$) are often depicted using commutative diagrams, where "points" stand for objects and "arrows" for morphisms.

$$\begin{tikzcd} A \arrow[r, "g"] \arrow[dr, "f\circ g"'] & B \arrow[d, "f"] \\ & C \end{tikzcd}
\qquad
\begin{tikzcd} A \arrow[r, "h"] \arrow[rr, bend left, "g\circ h"] \arrow[rrr, bend left=50, "f\circ(g\circ h)"] \arrow[rrr, bend right=50, "(f\circ g)\circ h"'] & B \arrow[r, "g"] \arrow[rr, bend right, "f\circ g"'] & C \arrow[r, "f"] & D \end{tikzcd}
\qquad
\begin{tikzcd} A \arrow[r, "f"] \arrow[dr, "f"'] & B \arrow[d, "\mathrm{id}_B"] \\ & B \end{tikzcd}
\qquad
\begin{tikzcd} A \arrow[r, "\mathrm{id}_A"] \arrow[dr, "f"'] & A \arrow[d, "f"] \\ & B \end{tikzcd}$$

A category $\boldsymbol{C}$ is given by:

- a collection of *objects*;
- for each pair of objects A and B, a collection $\boldsymbol{C}(A, B)$ of *morphisms* (aka *arrows*);
- for each triple A, B, C of objects, a *composition* operation $\circ \colon \boldsymbol{C}(B, C) \times \boldsymbol{C}(A, B) \to \boldsymbol{C}(A, C)$ satisfying *associativity*, i.e.
$$f \circ (g \circ h) = (f \circ g) \circ h \qquad \text{for all } f \in \boldsymbol{C}(C, D),\ g \in \boldsymbol{C}(B, C),\ h \in \boldsymbol{C}(A, B);$$
- for every object A, a morphism $\mathrm{id}_A \in \boldsymbol{C}(A, A)$, called *identity* on A, such that
$$\mathrm{id}_B \circ f = f = f \circ \mathrm{id}_A \qquad \text{for all } f \in \boldsymbol{C}(A, B).$$

Examples

1. The category $\boldsymbol{Set}$ has sets as objects and functions as morphisms. Identities and composition are defined as expected.

A category $\boldsymbol{C}$ is given by:

- a collection of *objects*;
- for each pair of objects A and B, a collection $\boldsymbol{C}(A,B)$ of *morphisms* (aka *arrows*);
- for each triple A,B,C of objects, a *composition* operation $\circ\colon \boldsymbol{C}(B,C) \times \boldsymbol{C}(A,B) \to \boldsymbol{C}(A,C)$ satisfying *associativity*, i.e.

$$f \circ (g \circ h) = (f \circ g) \circ h \qquad \text{for all } f \in \boldsymbol{C}(C,D),\ g \in \boldsymbol{C}(B,C),\ h \in \boldsymbol{C}(A,B);$$

- for every object A, a morphism $\mathrm{id}_A \in \boldsymbol{C}(A,A)$, called *identity* on A, such that

$$\mathrm{id}_B \circ f = f = f \circ \mathrm{id}_A \qquad \text{for all } f \in \boldsymbol{C}(A,B).$$

Examples

1. The category $\boldsymbol{Set}$ has sets as objects and functions as morphisms. Identities and composition are defined as expected.
2. The category $\boldsymbol{Rel}$ has sets as objects and relations as morphisms. Identity for an obj. A is $\mathrm{id}_A = \{(a,a) \mid a \in A\}$. Composition of $R \in \boldsymbol{Rel}(B,C)$, $S \in \boldsymbol{Rel}(A,B)$ is:

$$R \circ S = \{(a,c) \in A \times C \mid \exists\, b \in B : (a,b) \in S,\ (b,c) \in R\}$$

A category $\boldsymbol{C}$ is given by:

- a collection of *objects*;
- for each pair of objects A and B, a collection $\boldsymbol{C}(A, B)$ of *morphisms* (aka *arrows*);
- for each triple A, B, C of objects, a *composition* operation $\circ \colon \boldsymbol{C}(B, C) \times \boldsymbol{C}(A, B) \to \boldsymbol{C}(A, C)$ satisfying *associativity*, i.e.
$$f \circ (g \circ h) = (f \circ g) \circ h \qquad \text{for all } f \in \boldsymbol{C}(C, D),\ g \in \boldsymbol{C}(B, C),\ h \in \boldsymbol{C}(A, B);$$
- for every object A, a morphism $\mathrm{id}_A \in \boldsymbol{C}(A, A)$, called *identity* on A, such that
$$\mathrm{id}_B \circ f = f = f \circ \mathrm{id}_A \qquad \text{for all } f \in \boldsymbol{C}(A, B).$$

Examples

1. The category $\boldsymbol{Set}$ has sets as objects and functions as morphisms. Identities and composition are defined as expected.
2. The category $\boldsymbol{Rel}$ has sets as objects and relations as morphisms. Identity for an obj. A is $\mathrm{id}_A = \{(a, a) \mid a \in A\}$. Composition of $R \in \boldsymbol{Rel}(B, C)$, $S \in \boldsymbol{Rel}(A, B)$ is:
$$R \circ S = \{(a, c) \in A \times C \mid \exists\, b \in B : (a, b) \in S,\ (b, c) \in R\}$$
3. The category $\boldsymbol{Cpo}$ has cpo's as objects and Scott-continuous functions as morphisms. Identities and composition are defined as expected.

A partially ordered set (poset for short) is a set D with an order (that is, reflexive, transitive and antisymmetric) relation $\leq$ on D.

In a poset $(D, \leq)$, a subset $X \subseteq D$ is directed if $X \neq \emptyset$ and for every $x, y \in X$ there is an upper bound, that is, there is $z \in X$ such that $x \leq z$ and $y \leq z$.

In a poset $(D, \leq)$, the supremum or least upper bound (lub for short) $\bigvee X$ of $X \subseteq D$ is the smallest $z \in D$ such that $x \leq z$ for all $x \in X$.

A complete partial order (cpo for short) is a partially ordered set $(D, \leq)$ such that:

- there is a least element $\perp \in D$, that is, $\perp \leq x$ for all $x \in D$;
- for every directed $X \subseteq D$ there is a lub $\bigvee X \in D$.

Examples

1. The category $\boldsymbol{Set}$ has sets as objects and functions as morphisms. Identities and composition are defined as expected.
2. The category $\boldsymbol{Rel}$ has sets as objects and relations as morphisms. Identity for an obj. A is $\mathrm{id}_A = \{(a, a) \mid a \in A\}$. Composition of $R \in \boldsymbol{Rel}(B, C)$, $S \in \boldsymbol{Rel}(A, B)$ is:

$$R \circ S = \{(a, c) \in A \times C \mid \exists b \in B : (a, b) \in S,\ (b, c) \in R\}$$

3. The category $\boldsymbol{Cpo}$ has cpo's as objects and Scott-continuous functions as morphisms. Identities and composition are defined as expected.

In a category $\boldsymbol{C}$, a product of two objects A_1, A_2 is an object $A_1 \times A_2$ with two morphisms $\pi_i \colon A_1 \times A_2 \to A_i$ called *projections* (with $i \in \{1, 2\}$) such that, for every object C and morphisms $f_i \colon C \to A_i$ (with $i \in \{1, 2\}$), there is a unique morphism $\langle f_1, f_2\rangle \colon C \to A_1 \times A_2$ such that $\pi_i \circ \langle f_1, f_2\rangle = f_i$ (with $i \in \{1, 2\}$).

That is, the following diagrams commute

$$\begin{tikzcd} & C \arrow[dl, "f_1"'] \arrow[d, dashed, "\langle f_1, f_2\rangle"] \arrow[dr, "f_2"] & \\ A_1 & A_1 \times A_2 \arrow[l, "\pi_1"] \arrow[r, "\pi_2"'] & A_2 \end{tikzcd}$$

In a category $\boldsymbol{C}$, a product of two objects A_1, A_2 is an object $A_1 \times A_2$ with two morphisms $\pi_i \colon A_1 \times A_2 \to A_i$ called *projections* (with $i \in \{1, 2\}$) such that, for every object C and morphisms $f_i \colon C \to A_i$ (with $i \in \{1, 2\}$), there is a unique morphism $\langle f_1, f_2 \rangle \colon C \to A_1 \times A_2$ such that $\pi_i \circ \langle f_1, f_2 \rangle = f_i$ (with $i \in \{1, 2\}$).

That is, the following diagrams commute

$$\begin{tikzcd} & C \arrow[dl, "f_1"'] \arrow[d, dashed, "\langle f_1, f_2 \rangle"] \arrow[dr, "f_2"] & \\ A_1 & A_1 \times A_2 \arrow[l, "\pi_1"] \arrow[r, "\pi_2"'] & A_2 \end{tikzcd}$$

Lemma (Uniqueness of the product)

The product of two objects (if any) is unique up to isomorphism (i.e. given two products $(A_1 \times A_2, \pi_1, \pi_2)$ and $(A_1 \times' A_2, \pi'_1, \pi'_2)$, there are morphisms $f \colon A_1 \times A_2 \to A_1 \times' A_2$ and $g \colon A_1 \times' A_2 \to A_1 \times A_2$ such that $f \circ g = \mathrm{id}_{A_1 \times' A_2}$ and $g \circ f = \mathrm{id}_{A_1 \times A_2}$).

In a category $\boldsymbol{C}$, a product of two objects A_1, A_2 is an object $A_1 \times A_2$ with two morphisms $\pi_i \colon A_1 \times A_2 \to A_i$ called *projections* (with $i \in \{1, 2\}$) such that, for every object C and morphisms $f_i \colon C \to A_i$ (with $i \in \{1, 2\}$), there is a unique morphism $\langle f_1, f_2 \rangle \colon C \to A_1 \times A_2$ such that $\pi_i \circ \langle f_1, f_2 \rangle = f_i$ (with $i \in \{1, 2\}$).

That is, the following diagrams commute

$$\begin{tikzcd} & C \arrow[dl, "f_1"'] \arrow[d, dashed, "\langle f_1, f_2 \rangle"] \arrow[dr, "f_2"] & \\ A_1 & A_1 \times A_2 \arrow[l, "\pi_1"] \arrow[r, "\pi_2"'] & A_2 \end{tikzcd}$$

In a category $\boldsymbol{C}$, an object $\mathbf{1}$ is terminal if, for every object A, there is a unique morphism $!_A \in \boldsymbol{C}(A, \mathbf{1})$.

A category $\boldsymbol{C}$ is Cartesian if it has a terminal object $\mathbf{1}$ and every pair of objects A_1, A_2 has a product $(A_1 \times A_2, \pi_1, \pi_2)$.

In a category $\boldsymbol{C}$, a product of two objects A_1, A_2 is an object $A_1 \times A_2$ with two morphisms $\pi_i \colon A_1 \times A_2 \to A_i$ called *projections* (with $i \in \{1,2\}$) such that, for every object C and morphisms $f_i \colon C \to A_i$ (with $i \in \{1,2\}$), there is a unique morphism $\langle f_1, f_2 \rangle \colon C \to A_1 \times A_2$ such that $\pi_i \circ \langle f_1, f_2 \rangle = f_i$ (with $i \in \{1,2\}$).

That is, the following diagrams commute

$$\begin{tikzcd} & C \arrow[dl, "f_1"'] \arrow[d, dashed, "\langle f_1, f_2 \rangle"] \arrow[dr, "f_2"] & \\ A_1 & A_1 \times A_2 \arrow[l, "\pi_1"] \arrow[r, "\pi_2"'] & A_2 \end{tikzcd}$$

In a category $\boldsymbol{C}$, an object $\mathbf{1}$ is terminal if, for every object A, there is a unique morphism $!_A \in \boldsymbol{C}(A, \mathbf{1})$.

A category $\boldsymbol{C}$ is Cartesian if it has a terminal object $\mathbf{1}$ and every pair of objects A_1, A_2 has a product $(A_1 \times A_2, \pi_1, \pi_2)$.

Lemma (The terminal object is the neutral element of the product)

In a Cartesian category, for every object A, the objects $\mathbf{1} \times A$ and A and $A \times \mathbf{1}$ are isomorphic (that is, there are morphisms $f \colon \mathbf{1} \times A \to A$ and $g \colon A \to \mathbf{1} \times A$ such that $f \circ g = \mathrm{id}_A$ and $g \circ f = \mathrm{id}_{\mathbf{1} \times A}$, and similarly for A and $A \times \mathbf{1}$).

Category Theory in a Nutshell

In a category $\boldsymbol{C}$, a product of two objects A_1, A_2 is an object $A_1 \times A_2$ with two morphisms $\pi_i \colon A_1 \times A_2 \to A_i$ called *projections* (with $i \in \{1,2\}$) such that, for every object C and morphisms $f_i \colon C \to A_i$ (with $i \in \{1,2\}$), there is a unique morphism $\langle f_1, f_2 \rangle \colon C \to A_1 \times A_2$ such that $\pi_i \circ \langle f_1, f_2 \rangle = f_i$ (with $i \in \{1,2\}$).

That is, the following diagrams commute

$$\begin{tikzcd} & C \arrow[dl, "f_1"'] \arrow[d, dashed, "\langle f_1, f_2 \rangle"] \arrow[dr, "f_2"] & \\ A_1 & A_1 \times A_2 \arrow[l, "\pi_1"] \arrow[r, "\pi_2"'] & A_2 \end{tikzcd}$$

In a category $\boldsymbol{C}$, an object $\mathbf{1}$ is terminal if, for every object A, there is a unique morphism $!_A \in \boldsymbol{C}(A, \mathbf{1})$.

A category $\boldsymbol{C}$ is Cartesian if it has a terminal object $\mathbf{1}$ and every pair of objects A_1, A_2 has a product $(A_1 \times A_2, \pi_1, \pi_2)$.

Examples

1. The category $\boldsymbol{Set}$ is Cartesian. (Which product? Which terminal object?)

Category Theory in a Nutshell

In a category $\boldsymbol{C}$, a product of two objects A_1, A_2 is an object $A_1 \times A_2$ with two morphisms $\pi_i\colon A_1 \times A_2 \to A_i$ called *projections* (with $i \in \{1, 2\}$) such that, for every object C and morphisms $f_i\colon C \to A_i$ (with $i \in \{1, 2\}$), there is a unique morphism $\langle f_1, f_2\rangle\colon C \to A_1 \times A_2$ such that $\pi_i \circ \langle f_1, f_2\rangle = f_i$ (with $i \in \{1, 2\}$).

That is, the following diagrams commute

$$
\begin{tikzcd}
& C \arrow[dl, "f_1"'] \arrow[d, dashed, "\langle f_1, f_2\rangle"] \arrow[dr, "f_2"] & \\
A_1 & A_1 \times A_2 \arrow[l, "\pi_1"] \arrow[r, "\pi_2"'] & A_2
\end{tikzcd}
$$

In a category $\boldsymbol{C}$, an object $\mathbf{1}$ is terminal if, for every object A, there is a unique morphism $!_A \in \boldsymbol{C}(A, \mathbf{1})$.

A category $\boldsymbol{C}$ is Cartesian if it has a terminal object $\mathbf{1}$ and every pair of objects A_1, A_2 has a product $(A_1 \times A_2, \pi_1, \pi_2)$.

Examples

1. The category ***Set*** is Cartesian. (Which product? Which terminal object?)
2. The category ***Rel*** is Cartesian (Which product? Which terminal object?)

In a category $\boldsymbol{C}$, a product of two objects A_1, A_2 is an object $A_1 \times A_2$ with two morphisms $\pi_i \colon A_1 \times A_2 \to A_i$ called *projections* (with $i \in \{1,2\}$) such that, for every object C and morphisms $f_i \colon C \to A_i$ (with $i \in \{1,2\}$), there is a unique morphism $\langle f_1, f_2 \rangle \colon C \to A_1 \times A_2$ such that $\pi_i \circ \langle f_1, f_2 \rangle = f_i$ (with $i \in \{1,2\}$).

That is, the following diagrams commute

$$\begin{tikzcd} & C \arrow[dl, "f_1"'] \arrow[d, dashed, "\langle f_1, f_2 \rangle"] \arrow[dr, "f_2"] & \\ A_1 & A_1 \times A_2 \arrow[l, "\pi_1"] \arrow[r, "\pi_2"'] & A_2 \end{tikzcd}$$

In a category $\boldsymbol{C}$, an object $\mathbf{1}$ is terminal if, for every object A, there is a unique morphism $!_A \in \boldsymbol{C}(A, \mathbf{1})$.

A category $\boldsymbol{C}$ is Cartesian if it has a terminal object $\mathbf{1}$ and every pair of objects A_1, A_2 has a product $(A_1 \times A_2, \pi_1, \pi_2)$.

Examples

1. The category $\boldsymbol{Set}$ is Cartesian. (Which product? Which terminal object?)
2. The category $\boldsymbol{Rel}$ is Cartesian (Which product? Which terminal object?)
3. The category $\boldsymbol{Cpo}$ is Cartesian (Which product? Which terminal object?)

Category Theory in a Nutshell

In a category $\boldsymbol{C}$, a product of two objects A_1, A_2 is an object $A_1 \times A_2$ with two morphisms $\pi_i \colon A_1 \times A_2 \to A_i$ called *projections* (with $i \in \{1, 2\}$) such that, for every object C and morphisms $f_i \colon C \to A_i$ (with $i \in \{1, 2\}$), there is a unique morphism $\langle f_1, f_2 \rangle \colon C \to A_1 \times A_2$ such that $\pi_i \circ \langle f_1, f_2 \rangle = f_i$ (with $i \in \{1, 2\}$).

That is, the following diagrams commute

$$\begin{tikzcd} & C \arrow[dl, "f_1"'] \arrow[d, dashed, "\langle f_1, f_2 \rangle"] \arrow[dr, "f_2"] & \\ A_1 & A_1 \times A_2 \arrow[l, "\pi_1"] \arrow[r, "\pi_2"'] & A_2 \end{tikzcd}$$

In a category $\boldsymbol{C}$, an object $\mathbf{1}$ is terminal if, for every object A, there is a unique morphism $!_A \in \boldsymbol{C}(A, \mathbf{1})$.

A category $\boldsymbol{C}$ is Cartesian if it has a terminal object $\mathbf{1}$ and every pair of objects A_1, A_2 has a product $(A_1 \times A_2, \pi_1, \pi_2)$.

Example

1. For every $n \in \mathbb{N}$, define a notion of n-product $(A_1 \times \cdots \times A_n, \pi_1, \ldots, \pi_n)$ of the objects $A_1, \ldots, A_n$, by generalizing the definition of product and terminal object.
2. Prove that a Cartesian category has the n-product of any n objects, for all $n \in \mathbb{N}$.

Notation: In a Cartesian category $\boldsymbol{C}$, for every morphisms $f_i \in \boldsymbol{C}(A_i, B_i)$ with $i \in \{1, 2\}$, the *product map* is $f_1 \times f_2 = \langle f_1 \circ \pi_1, f_2 \circ \pi_2 \rangle \in \boldsymbol{C}(A_1 \times A_2, B_1 \times B_2)$.

A Cartesian category $\boldsymbol{C}$ is closed (CCC for short) if for every objects A, B there is an object $A \Rightarrow B$, called *exponent*, and a morphism $\mathrm{ev}_{A,B} \in \boldsymbol{C}(A \Rightarrow B \times A,\ B)$, called *evaluation*, such that, for every $f \in \boldsymbol{C}(C \times A,\ B)$ there is a unique $\mathrm{curry}(f) \in \boldsymbol{C}(C,\ A \Rightarrow B)$ such that $\mathrm{ev}_{A,B} \circ (\mathrm{curry}(f) \times \mathrm{id}_A) = f$.

That is, the following diagram commutes

$$\begin{tikzcd} C \times A \arrow[r, "f"] \arrow[d, dashed, "\mathrm{curry}(f) \times \mathrm{id}_A"'] & B \\ A \Rightarrow B \times A \arrow[ur, "\mathrm{ev}_{A,B}"'] & \end{tikzcd}$$

Notation: In a Cartesian category $\boldsymbol{C}$, for every morphisms $f_i \in \boldsymbol{C}(A_i, B_i)$ with $i \in \{1, 2\}$, the *product map* is $f_1 \times f_2 = \langle f_1 \circ \pi_1, f_2 \circ \pi_2 \rangle \in \boldsymbol{C}(A_1 \times A_2, B_1 \times B_2)$.

A Cartesian category $\boldsymbol{C}$ is closed (CCC for short) if for every objects A, B there is an object $A \Rightarrow B$, called *exponent*, and a morphism $\mathrm{ev}_{A,B} \in \boldsymbol{C}(A \Rightarrow B \times A,\ B)$, called *evaluation*, such that, for every $f \in \boldsymbol{C}(C \times A,\ B)$ there is a unique $\mathrm{curry}(f) \in \boldsymbol{C}(C,\ A \Rightarrow B)$ such that $\mathrm{ev}_{A,B} \circ (\mathrm{curry}(f) \times \mathrm{id}_A) = f$.

That is, the following diagram commutes

$$\begin{tikzcd}
C \times A \arrow[r, "f"] \arrow[d, dashed, "\mathrm{curry}(f) \times \mathrm{id}_A"'] & B \\
A \Rightarrow B \times A \arrow[ur, "\mathrm{ev}_{A,B}"'] &
\end{tikzcd}$$

Idea: The object $A \Rightarrow B$ "represents/internalizes" the hom-set $\boldsymbol{C}(A, B)$.

Notation: In a Cartesian category $\boldsymbol{C}$, for every morphisms $f_i \in \boldsymbol{C}(A_i, B_i)$ with $i \in \{1, 2\}$, the *product map* is $f_1 \times f_2 = \langle f_1 \circ \pi_1, f_2 \circ \pi_2 \rangle \in \boldsymbol{C}(A_1 \times A_2, B_1 \times B_2)$.

A Cartesian category $\boldsymbol{C}$ is closed (CCC for short) if for every objects A, B there is an object $A \Rightarrow B$, called *exponent*, and a morphism $\mathrm{ev}_{A,B} \in \boldsymbol{C}(A \Rightarrow B \times A,\ B)$, called *evaluation*, such that, for every $f \in \boldsymbol{C}(C \times A,\ B)$ there is a unique $\mathrm{curry}(f) \in \boldsymbol{C}(C,\ A \Rightarrow B)$ such that $\mathrm{ev}_{A,B} \circ (\mathrm{curry}(f) \times \mathrm{id}_A) = f$.

That is, the following diagram commutes

$$\begin{tikzcd} C \times A \arrow[r, "f"] \arrow[d, dashed, "\mathrm{curry}(f) \times \mathrm{id}_A"'] & B \\ A \Rightarrow B \times A \arrow[ur, "\mathrm{ev}_{A,B}"'] & \end{tikzcd}$$

Examples

1. The category $\boldsymbol{Set}$ is Cartesian closed. (Which exponent? Which evaluation?)

Notation: In a Cartesian category $\boldsymbol{C}$, for every morphisms $f_i \in \boldsymbol{C}(A_i, B_i)$ with $i \in \{1, 2\}$, the *product map* is $f_1 \times f_2 = \langle f_1 \circ \pi_1, f_2 \circ \pi_2 \rangle \in \boldsymbol{C}(A_1 \times A_2, B_1 \times B_2)$.

A Cartesian category $\boldsymbol{C}$ is closed (CCC for short) if for every objects A, B there is an object $A \Rightarrow B$, called *exponent*, and a morphism $\mathrm{ev}_{A,B} \in \boldsymbol{C}(A \Rightarrow B \times A,\ B)$, called *evaluation*, such that, for every $f \in \boldsymbol{C}(C \times A,\ B)$ there is a unique $\mathrm{curry}(f) \in \boldsymbol{C}(C,\ A \Rightarrow B)$ such that $\mathrm{ev}_{A,B} \circ (\mathrm{curry}(f) \times \mathrm{id}_A) = f$.

That is, the following diagram commutes

$$\begin{tikzcd} C \times A \arrow[r, "f"] \arrow[d, dashed, "\mathrm{curry}(f) \times \mathrm{id}_A"'] & B \\ A \Rightarrow B \times A \arrow[ur, "\mathrm{ev}_{A,B}"'] & \end{tikzcd}$$

Examples

1. The category $\boldsymbol{Set}$ is Cartesian closed. (Which exponent? Which evaluation?)
2. The category $\boldsymbol{Rel}$ is not Cartesian closed. (How to prove it?)

Notation: In a Cartesian category $\boldsymbol{C}$, for every morphisms $f_i \in \boldsymbol{C}(A_i, B_i)$ with $i \in \{1,2\}$, the *product map* is $f_1 \times f_2 = \langle f_1 \circ \pi_1, f_2 \circ \pi_2 \rangle \in \boldsymbol{C}(A_1 \times A_2, B_1 \times B_2)$.

A Cartesian category $\boldsymbol{C}$ is closed (CCC for short) if for every objects A, B there is an object $A \Rightarrow B$, called *exponent*, and a morphism $\mathrm{ev}_{A,B} \in \boldsymbol{C}(A \Rightarrow B \times A,\ B)$, called *evaluation*, such that, for every $f \in \boldsymbol{C}(C \times A,\ B)$ there is a unique $\mathrm{curry}(f) \in \boldsymbol{C}(C,\ A \Rightarrow B)$ such that $\mathrm{ev}_{A,B} \circ (\mathrm{curry}(f) \times \mathrm{id}_A) = f$.

That is, the following diagram commutes

$$\begin{tikzcd} C \times A \arrow[r, "f"] \arrow[d, dashed, "\mathrm{curry}(f) \times \mathrm{id}_A"'] & B \\ A \Rightarrow B \times A \arrow[ur, "\mathrm{ev}_{A,B}"'] & \end{tikzcd}$$

Examples

1. The category $\boldsymbol{Set}$ is Cartesian closed. (Which exponent? Which evaluation?)
2. The category $\boldsymbol{Rel}$ is not Cartesian closed. (How to prove it?)
3. The category $\boldsymbol{Cpo}$ is Cartesian closed. (Which exponent? Which evaluation?)

Notation: In a Cartesian category $\boldsymbol{C}$, for every morphisms $f_i \in \boldsymbol{C}(A_i, B_i)$ with $i \in \{1, 2\}$, the *product map* is $f_1 \times f_2 = \langle f_1 \circ \pi_1, f_2 \circ \pi_2 \rangle \in \boldsymbol{C}(A_1 \times A_2, B_1 \times B_2)$.

A Cartesian category $\boldsymbol{C}$ is closed (CCC for short) if for every objects A, B there is an object $A \Rightarrow B$, called *exponent*, and a morphism $\mathrm{ev}_{A,B} \in \boldsymbol{C}(A \Rightarrow B \times A,\ B)$, called *evaluation*, such that, for every $f \in \boldsymbol{C}(C \times A,\ B)$ there is a unique $\mathrm{curry}(f) \in \boldsymbol{C}(C,\ A \Rightarrow B)$ such that $\mathrm{ev}_{A,B} \circ (\mathrm{curry}(f) \times \mathrm{id}_A) = f$.

That is, the following diagram commutes

$$\begin{tikzcd} C \times A \arrow[r, "f"] \arrow[d, dashed, "\mathrm{curry}(f) \times \mathrm{id}_A"'] & B \\ A \Rightarrow B \times A \arrow[ur, "\mathrm{ev}_{A,B}"'] & \end{tikzcd}$$

Lemma

In a CCC, for every $f : C \times A \to B$, $g : C \to B$, $h : D \to C$ and $k : C \to A$, we have:

$$\langle k, g \rangle \circ h = \langle k \circ h, g \circ h \rangle \quad : D \to A \times B \qquad \text{(pair)}$$

$$\mathrm{ev}_{A,B} \circ \langle \mathrm{curry}(f), g \rangle = f \circ \langle \mathrm{id}_C, g \rangle \quad : C \to B \qquad (\beta_e)$$

$$\mathrm{curry}(f) \circ h = \mathrm{curry}(f \circ (h \times \mathrm{id}_A)) \quad : D \to A \Rightarrow B \qquad \text{(curry)}$$

Proof. Exercise! □

1. What is Denotational Semantics for Programming Languages?
2. Category Theory in a Nutshell
3. Categorical Semantics for the (Untyped) λ-Calculus
4. Summary, Exercises, Bibliography

Categorical Semantics for the (Untyped) λ-Calculus

In a CCC $\boldsymbol{C}$, a reflexive object is a triple $(U, \lambda, \mathrm{fun})$ where λ, fun are a *retraction* on the object U, that is, $\lambda \in \boldsymbol{C}(U \Rightarrow U,\, U)$ and $\mathrm{fun} \in \boldsymbol{C}(U,\, U \Rightarrow U)$ with $\mathrm{fun} \circ \lambda = \mathrm{id}_{U \Rightarrow U}$.

Idea: A reflexive object $(U, \lambda, \mathrm{fun})$ is a non-set-theoretic way to say "$(U \Rightarrow U) \subseteq U$".

Rmk: It is possible to define $@ \in \boldsymbol{C}(U \times U,\, U)$ such that $\mathrm{fun} = \mathrm{curry}(@)$. The notation for these morphisms are consistent with what you have concretely seen in Days 1–2.

In a CCC $\boldsymbol{C}$, a reflexive object is a triple $(U, \lambda, \mathrm{fun})$ where λ, fun are a *retraction* on the object U, that is, $\lambda \in \boldsymbol{C}(U \Rightarrow U,\, U)$ and $\mathrm{fun} \in \boldsymbol{C}(U,\, U \Rightarrow U)$ with $\mathrm{fun} \circ \lambda = \mathrm{id}_{U \Rightarrow U}$.

Examples

1. In the CCC $\boldsymbol{Set}$, there is no reflexive object. (How to prove it?)

In a CCC $\boldsymbol{C}$, a reflexive object is a triple $(U, \lambda, \mathrm{fun})$ where λ, fun are a *retraction* on the object U, that is, $\lambda \in \boldsymbol{C}(U \Rightarrow U,\, U)$ and $\mathrm{fun} \in \boldsymbol{C}(U,\, U \Rightarrow U)$ with $\mathrm{fun} \circ \lambda = \mathrm{id}_{U \Rightarrow U}$.

Examples

1. In the CCC $\boldsymbol{Set}$, there is no reflexive object. (How to prove it?)
2. In the category $\boldsymbol{Rel}$, there is no reflexive object. (Why?)

In a CCC $\boldsymbol{C}$, a reflexive object is a triple $(U, \lambda, \mathrm{fun})$ where λ, fun are a *retraction* on the object U, that is, $\lambda \in \boldsymbol{C}(U \Rightarrow U,\, U)$ and $\mathrm{fun} \in \boldsymbol{C}(U,\, U \Rightarrow U)$ with $\mathrm{fun} \circ \lambda = \mathrm{id}_{U \Rightarrow U}$.

Examples

1. In the CCC $\boldsymbol{Set}$, there is no reflexive object. (How to prove it?)
2. In the category $\boldsymbol{Rel}$, there is no reflexive object. (Why?)
3. In the CCC $\boldsymbol{Cpo}$, there is a reflexive object. (Which one? Look at Day 1...)

In a CCC $\boldsymbol{C}$, a reflexive object is a triple $(U, \lambda, \mathrm{fun})$ where λ, fun are a *retraction* on the object U, that is, $\lambda \in \boldsymbol{C}(U \Rightarrow U,\, U)$ and $\mathrm{fun} \in \boldsymbol{C}(U,\, U \Rightarrow U)$ with $\mathrm{fun} \circ \lambda = \mathrm{id}_{U \Rightarrow U}$.

Examples

1. In the CCC $\boldsymbol{Set}$, there is no reflexive object. (How to prove it?)
2. In the category $\boldsymbol{Rel}$, there is no reflexive object. (Why?)
3. In the CCC $\boldsymbol{Cpo}$, there is a reflexive object. (Which one? The cpo $(\mathcal{P}(\mathbb{N}), \subseteq)$!)

In a CCC $\boldsymbol{C}$, a reflexive object is a triple $(U, \lambda, \mathrm{fun})$ where λ, fun are a *retraction* on the object U, that is, $\lambda \in \boldsymbol{C}(U \Rightarrow U, U)$ and $\mathrm{fun} \in \boldsymbol{C}(U, U \Rightarrow U)$ with $\mathrm{fun} \circ \lambda = \mathrm{id}_{U \Rightarrow U}$.

A sequence $\vec{x} = (x_1, \ldots, x_n)$ of variables is *adequate* for $\mathtt{M} \in \Lambda$ if the x_i's are pairwise distinct and $\mathsf{fv}(\mathtt{M}) \subseteq \{x_1, \ldots, x_n\}$. We write U^n for the n-product $U \times \overset{n}{\ldots} \times U$.

We are going to interpret λ-terms in *any* reflexive object of *any* CCC.

In a CCC $\boldsymbol{C}$, a reflexive object is a triple $(U, \lambda, \mathrm{fun})$ where λ, fun are a *retraction* on the object U, that is, $\lambda \in \boldsymbol{C}(U \Rightarrow U, U)$ and $\mathrm{fun} \in \boldsymbol{C}(U, U \Rightarrow U)$ with $\mathrm{fun} \circ \lambda = \mathrm{id}_{U \Rightarrow U}$.

A sequence $\vec{x} = (x_1, \ldots, x_n)$ of variables is *adequate* for $\mathtt{M} \in \Lambda$ if the x_i's are pairwise distinct and $\mathsf{fv}(\mathtt{M}) \subseteq \{x_1, \ldots, x_n\}$. We write U^n for the n-product $U \times \overset{n}{\ldots} \times U$.

We are going to interpret λ-terms in *any* reflexive object of *any* CCC.

Definition (Categorical semantics/interpretation of λ-terms)

Let $\vec{x} = (x_1, \ldots, x_n)$ be adequate for $\mathtt{M} \in \Lambda$. The categorical semantics of $\mathtt{M}$ wrt $\vec{x}$ in a reflexive object $(U, \lambda, \mathrm{fun})$ of a CCC is a morphism $[\![\mathtt{M}]\!]_{\vec{x}} \colon U^n \to U$ defined by:

$$
\begin{aligned}
[\![x_i]\!]_{\vec{x}} &= \pi_i && \text{where } i \in \{1, \ldots, n\} \\
[\![\mathtt{M}\,\mathtt{N}]\!]_{\vec{x}} &= \mathrm{ev}_{U,U} \circ \langle \mathrm{fun} \circ [\![\mathtt{M}]\!]_{\vec{x}},\ [\![\mathtt{N}]\!]_{\vec{x}} \rangle && \\
[\![\lambda y.\mathtt{N}]\!]_{\vec{x}} &= \lambda \circ \mathrm{curry}([\![\mathtt{N}]\!]_{\vec{x},y}) && \text{we assume wlog } y \notin \{x_1, \ldots, x_n\}
\end{aligned}
$$

In a CCC $\boldsymbol{C}$, a reflexive object is a triple $(U, \lambda, \mathrm{fun})$ where λ, fun are a *retraction* on the object U, that is, $\lambda \in \boldsymbol{C}(U \Rightarrow U,\, U)$ and $\mathrm{fun} \in \boldsymbol{C}(U,\, U \Rightarrow U)$ with $\mathrm{fun} \circ \lambda = \mathrm{id}_{U \Rightarrow U}$.

A sequence $\vec{x} = (x_1, \ldots, x_n)$ of variables is *adequate* for $\mathtt{M} \in \Lambda$ if the x_i's are pairwise distinct and $\mathsf{fv}(\mathtt{M}) \subseteq \{x_1, \ldots, x_n\}$. We write U^n for the n-product $U \times \overset{n}{\ldots} \times U$.

We are going to interpret λ-terms in *any* reflexive object of *any* CCC.

Definition (Categorical semantics/interpretation of λ-terms)

Let $\vec{x} = (x_1, \ldots, x_n)$ be adequate for $\mathtt{M} \in \Lambda$. The categorical semantics of $\mathtt{M}$ wrt $\vec{x}$ in a reflexive object $(U, \lambda, \mathrm{fun})$ of a CCC is a morphism $[\![\mathtt{M}]\!]_{\vec{x}} \colon U^n \to U$ defined by:

$$
\begin{aligned}
[\![x_i]\!]_{\vec{x}} &= \pi_i && \text{where } i \in \{1, \ldots, n\} \\
[\![\mathtt{M}\,\mathtt{N}]\!]_{\vec{x}} &= \mathrm{ev}_{U,U} \circ \langle \mathrm{fun} \circ [\![\mathtt{M}]\!]_{\vec{x}},\ [\![\mathtt{N}]\!]_{\vec{x}} \rangle && \\
[\![\lambda y.\mathtt{N}]\!]_{\vec{x}} &= \lambda \circ \mathrm{curry}([\![\mathtt{N}]\!]_{\vec{x},y}) && \text{we assume wlog } y \notin \{x_1, \ldots, x_n\}
\end{aligned}
$$

Notation: When we write $[\![\mathtt{M}]\!]_{\vec{x}}$ we assume that $\vec{x}$ is adequate for $\mathtt{M}$, and we keep implicit the reflexive object $(U, \lambda, \mathrm{fun})$ where we interpret $\mathtt{M}$ (but it depends on it!).

Lemma (Substitution)

Let $\mathtt{M}, \mathtt{N} \in \Lambda$. *If* $x \notin \vec{y} = (y_1, \ldots, y_n)$ *then* $[\![\mathtt{M}\{x := \mathtt{N}\}]\!]_{\vec{y}} = [\![\mathtt{M}]\!]_{\vec{y},x} \circ \langle \mathrm{id}_{U^n}, [\![\mathtt{N}]\!]_{\vec{y}} \rangle$.

Proof. By induction on $\mathtt{M}$. Exercise! $\square$

Lemma (Substitution)

Let $\mathtt{M}, \mathtt{N} \in \Lambda$. *If* $x \notin \vec{y} = (y_1, \ldots, y_n)$ *then* $[\![\mathtt{M}\{x := \mathtt{N}\}]\!]_{\vec{y}} = [\![\mathtt{M}]\!]_{\vec{y},x} \circ \langle \mathrm{id}_{U^n}, [\![\mathtt{N}]\!]_{\vec{y}} \rangle$.

Proof. By induction on $\mathtt{M}$. Exercise! $\square$

Theorem (Invariance/Soundness)

Let $\mathtt{M}, \mathtt{N} \in \Lambda$. *If* $\mathtt{M} \to_\beta \mathtt{N}$ *then* $[\![\mathtt{M}]\!]_{\vec{x}} = [\![\mathtt{N}]\!]_{\vec{x}}$.

Lemma (Substitution)

Let $\mathtt{M}, \mathtt{N} \in \Lambda$. *If* $x \notin \vec{y} = (y_1, \ldots, y_n)$ *then* $[\![\mathtt{M}\{x := \mathtt{N}\}]\!]_{\vec{y}} = [\![\mathtt{M}]\!]_{\vec{y},x} \circ \langle \mathrm{id}_{U^n}, [\![\mathtt{N}]\!]_{\vec{y}} \rangle$.

Proof. By induction on $\mathtt{M}$. Exercise! □

Theorem (Invariance/Soundness)

Let $\mathtt{M}, \mathtt{N} \in \Lambda$. *If* $\mathtt{M} \to_\beta \mathtt{N}$ *then* $[\![\mathtt{M}]\!]_{\vec{x}} = [\![\mathtt{N}]\!]_{\vec{x}}$.

Proof. The key case is $\mathtt{M} = (\lambda y.\mathtt{M}_1)\mathtt{M}_2 \to_\beta \mathtt{M}_1\{y := \mathtt{M}_2\} = \mathtt{N}$. We assume wlog $y \notin \vec{x}$.

$$
\begin{aligned}
[\![\mathtt{M}]\!]_{\vec{x}} = [\![(\lambda y.\mathtt{M}_1)\mathtt{M}_2]\!]_{\vec{x}} &= \mathrm{ev}_{U,U} \circ \langle \mathrm{fun} \circ [\![\lambda y.\mathtt{M}_1]\!]_{\vec{x}},\ [\![\mathtt{M}_2]\!]_{\vec{x}} \rangle && (\text{def. of } [\![\cdot]\!]) \\
&= \mathrm{ev}_{U,U} \circ \langle \mathrm{fun} \circ \lambda \circ \mathrm{curry}([\![\mathtt{M}_1]\!]_{\vec{x},y}),\ [\![\mathtt{M}_2]\!]_{\vec{x}} \rangle && (\text{def. of } [\![\cdot]\!]) \\
&= \mathrm{ev}_{U,U} \circ \langle \mathrm{curry}([\![\mathtt{M}_1]\!]_{\vec{x},y}),\ [\![\mathtt{M}_2]\!]_{\vec{x}} \rangle && (\text{retraction}) \\
&= [\![\mathtt{M}_1]\!]_{\vec{x},y} \circ \langle \mathrm{id}_{U^n},\ [\![\mathtt{M}_2]\!]_{\vec{x}} \rangle && (\text{rule } \beta_e) \\
&= [\![\mathtt{M}_1\{x := \mathtt{M}_2\}]\!]_{\vec{y}} = [\![\mathtt{N}]\!]_{\vec{y}} && (\text{substitution})
\end{aligned}
$$

The other cases follow from the IH (proof by induction on the def. of $\mathtt{M} \to_\beta \mathtt{N}$). □

Lemma (Substitution)

Let $\mathtt{M}, \mathtt{N} \in \Lambda$. *If* $x \notin \vec{y} = (y_1, \ldots, y_n)$ *then* $[\![\mathtt{M}\{x := \mathtt{N}\}]\!]_{\vec{y}} = [\![\mathtt{M}]\!]_{\vec{y},x} \circ \langle \mathrm{id}_{U^n}, [\![\mathtt{N}]\!]_{\vec{y}} \rangle$.

Proof. By induction on $\mathtt{M}$. Exercise! □

Theorem (Invariance/Soundness)

Let $\mathtt{M}, \mathtt{N} \in \Lambda$. *If* $\mathtt{M} \to_\beta \mathtt{N}$ *then* $[\![\mathtt{M}]\!]_{\vec{x}} = [\![\mathtt{N}]\!]_{\vec{x}}$.

Proof. The key case is $\mathtt{M} = (\lambda y.\mathtt{M}_1)\mathtt{M}_2 \to_\beta \mathtt{M}_1\{y := \mathtt{M}_2\} = \mathtt{N}$. We assume wlog $y \notin \vec{x}$.

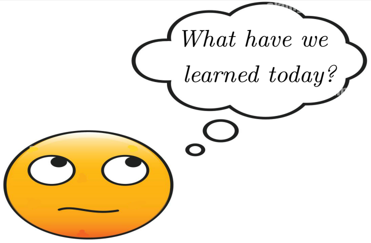
$$
\begin{aligned}
[\![\mathtt{M}]\!]_{\vec{x}} = [\![(\lambda y.\mathtt{M}_1)\mathtt{M}_2]\!]_{\vec{x}} &= \mathrm{ev}_{U,U} \circ \langle \mathrm{fun} \circ [\![\lambda y.\mathtt{M}_1]\!]_{\vec{x}},\ [\![\mathtt{M}_2]\!]_{\vec{x}} \rangle && \text{(def. of } [\![\cdot]\!]) \\
&= \mathrm{ev}_{U,U} \circ \langle \mathrm{fun} \circ \lambda \circ \mathrm{curry}([\![\mathtt{M}_1]\!]_{\vec{x},y}),\ [\![\mathtt{M}_2]\!]_{\vec{x}} \rangle && \text{(def. of } [\![\cdot]\!]) \\
&= \mathrm{ev}_{U,U} \circ \langle \mathrm{curry}([\![\mathtt{M}_1]\!]_{\vec{x},y}),\ [\![\mathtt{M}_2]\!]_{\vec{x}} \rangle && \text{(retraction)} \\
&= [\![\mathtt{M}_1]\!]_{\vec{x},y} \circ \langle \mathrm{id}_{U^n},\ [\![\mathtt{M}_2]\!]_{\vec{x}} \rangle && \text{(rule } \beta_e) \\
&= [\![\mathtt{M}_1\{x := \mathtt{M}_2\}]\!]_{\vec{y}} = [\![\mathtt{N}]\!]_{\vec{y}} && \text{(substitution)}
\end{aligned}
$$

The other cases follow from the IH (proof by induction on the def. of $\mathtt{M} \to_\beta \mathtt{N}$). □

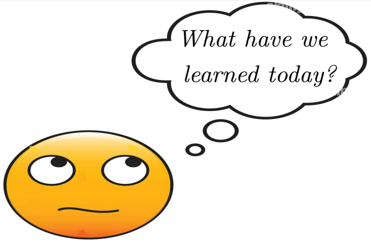
Even contextuality holds. Consistence depends on the specific reflexivity object.

1. What is Denotational Semantics for Programming Languages?
2. Category Theory in a Nutshell
3. Categorical Semantics for the (Untyped) λ-Calculus
4. Summary, Exercises, Bibliography

- What does denotational semantics is and is for.
- Some abstract properties that an algebraic structure has to fulfill to be a denotational semantics of the untyped λ-calculus.
- A taste of category theory.
- The notions of Cartesian closed category and reflexive object.
- How to interpret the untyped λ-calculus in a reflexive object of a Cartesian closed category.



- What does denotational semantics is and is for.
- Some abstract properties that an algebraic structure has to fulfill to be a denotational semantics of the untyped λ-calculus.
- A taste of category theory.
- The notions of Cartesian closed category and reflexive object.
- How to interpret the untyped λ-calculus in a reflexive object of a Cartesian closed category.



Rmk: Many of the notions and morphisms we have seen today are just an abstract version of what you have already seen instantiated more concretely in Days 1–2. The fact that we are using the same notations is not by chance...

Hint: Read again Days 1–2 slides keeping in mind the content of Day 3, and vice versa.

- Do the proofs of the statements on the slides.
- Look at our **notes** on the webpage of the course, there are plenty of **details**, **proofs** and **exercises**. Today's notes are under construction!
- The exercises will have **solutions** (but try to do them by yourself before looking at them!).
- Don't hesitate to **ask** us questions in person or on Discord about lectures, exercises, solutions, further reading.

- Chapters 1, 2, 3 and 9 of:
  Asperti A., Longo G.: Categories, Types and Structures, 1991,
  https://www.di.ens.fr/users/longo/files/CategTypesStructures/book.pdf

- Chapter 4 of:
  Amadio R., Curien P-L.: Domains and lambda-calculi, 1996,
  https://www.cambridge.org/core/books/domains-and-lambdacalculi/4C6AB6938E436CFA8D5A8533B76A7F23

- Chapters 1 and 5 of:
  Barendregt H. P.: The lambda-calculus, its syntax and semantics, 1984,
  https://www.sciencedirect.com/bookseries/studies-in-logic-and-the-foundations-of-mathematics/vol/103

- Chapter 1 of:
  Manzonetto G.: Models and theories of λ-calculus, PhD thesis, 2008,
  https://www.irif.fr/~gmanzone/ManzonettoPhdThesis.pdf

- To go (much) further:
  Barendregt H. P., Manzonetto G.: A Lambda Calculus Satellite, 2022,
  https://www.collegepublications.co.uk/logic/mlf/?00035