History

λ-calculus was introduced as a foundation for mathematics by Alonzo Church, in the 1930s.

It was proven to be logically inconsistent.

The subject was kept alive because of its computational properties.

$\boxed{\text{Thm: } \lambda\text{-definable functions} \Leftrightarrow \text{recursive functions}}$

There was later a weaker system called simply-typed λ-calculus introduced. It is more categorical.

The (untyped) Lambda Calculus

See Henk Barendregt - The Lambda Calculus: Its Syntax & Semantics

Electronic Copy At Library

You'll find a nice historic overview.

λ-calculus is (for one thing) a notation for specifying functions

<u>Informal Presentation</u>

$$f(x) = x^2+3$$

We'll write

$$\lambda x . x^2+3$$

This says " the operation of sending x to x^2+3.

Functions can be applied to terms.

$$(\lambda x. x^2+3)\,5 \xrightarrow[\beta]{} 5^2+3.$$

This is called β-reduction.

More generally denote an expression like x^2+3 as $t(x)$

$$(\lambda x. t(x))\,5 \xrightarrow[\beta]{} t(5)$$

In "x^2+3", x is a free variable.

(Formalize this later.)

In general, β-reduction is written like this:

If N is a term, M is a term

$$(\lambda x. N)\,M \xrightarrow[\beta]{} \underbrace{N[x:=M]}$$

substitute M for every occurence of x.

There is a restriction on when you can do this. We'll ignore for now

There can be more than one variable

$$(\lambda x. xy)7 \leadsto xy[x:=7]$$
$$= 7y$$

λ's can be iterated.

$$(\lambda x.\lambda y.xy)7 \leadsto (\lambda y.xy)7$$
$$= \lambda y.7y$$

You'll also see the notation

$$[7/x]M$$

as a way of denoting

$$M[x:=7]$$

Lots of exercises & examples in Barendregt.

~~Practice~~

Consider the term $\lambda x.xx$

This has the property of doubling everything we apply to it.

$$(\lambda x.xx)N \xrightarrow{\beta} xx[x:=N]$$
$$= NN$$

Here's the weird(?), awesome(?) thing about λ-calculus.

> Any term can be applied to any term. In particular, any term can be applied to itself.

$$(\lambda x.xx)\lambda x.xx \xrightarrow{\beta}$$

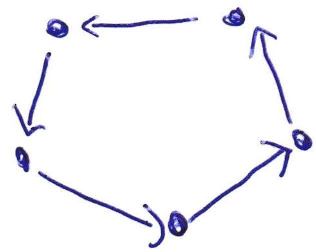
$$xx[x:=\lambda x.xx] = (\lambda x.xx)\lambda x.xx$$

So this term β-reduces to itself

This fact is represented by the β-graph

You'll find exercises in Barendregt like:

Find a λ-term whose β-graph is



and ~~th~~ even stranger examples. See p. 73.

$\boxed{\text{MORAL: The } \beta\text{-reduction process doesn't necessarily terminate}}$

Can we talk about functions of several variables?

$$f(x,y) = x^2y + 3xy^3$$

$$\lambda\langle x,y\rangle.\ x^2y + 3xy^3$$

we could write expressions like this.

Note: This works in any cartesian closed category.

Consider the following in the category of sets & functions

Let X and Y be sets. Let $X \Rightarrow Y$ be the set of all functions from X to Y.

Then:

$$(X \times Y) \Rightarrow Z \cong X \Rightarrow (Y \Rightarrow Z)$$

isomorphism of sets

Let $f \in (X \times Y) \Rightarrow Z$. Construct an element of $X \Rightarrow (Y \Rightarrow Z)$

$$\hat{f} : X \to Y \Rightarrow Z$$

$$\hat{f}(x) : Y \to Z$$

$$\underbrace{\hat{f}(x)(y) = f(x,y)}$$

You could write this as

$$\lambda x . \lambda y . f(x,y)$$

Exercise: Define the function

$$X \Rightarrow (Y \Rightarrow Z) \longrightarrow (X \times Y) \Rightarrow Z$$

and prove the two functions are inverses.

Preview of more difficult results.

In the untyped λ-calculus, everything is a term that can be plugged into a function and a function itself.

To model this mathematically, we would need a "set" such that

$$X \cong X \Rightarrow X$$

In ordinary sets, this is impossible except for the one-point set.

One must look at other cartesian closed categories.

Nontrivial example first described by Dana Scott.

· "continuous lattices"

Looking for more examples is a large industry.

Next time: Develop this formally.

Then look at <u>typed</u> λ-calculus.

A function $f: X \to Y$ has type $X \Rightarrow Y$

The only thing that f can be applied to are terms of type X.

$$
\begin{tikzcd}
\text{Typed } \lambda\text{-calculus} \arrow[rr, Leftrightarrow] \arrow[dr, Leftrightarrow] & & \text{Cartesian closed category} \\
& \text{intuitionistic logic for } \wedge, \Rightarrow \arrow[ur, Rightarrow] &
\end{tikzcd}
$$