Lambda Calculus

Now you can bring a computer to your tests!

Vincent Macri

William Lyon Mackenzie C.I. Math Club

Table of Contents

1 Introduction

2 One Argument

3 Booleans

4 Church Numerals

What is lambda calculus?

Introduction

- Created by Alonzo Church

What is lambda calculus?

Introduction

- Created by Alonzo Church
- A way of representing pure mathematical functions

What is lambda calculus?

Introduction

- Created by Alonzo Church
- A way of representing pure mathematical functions
- Can represent any computer program

What is lambda calculus?
Introduction

- Created by Alonzo Church
- A way of representing pure mathematical functions
- Can represent any computer program
- Equivalent to Turing machines

In math class, we would define a function that accepts an argument x and outputs $x + 1$ as so:

$$f(x)$$

In math class, we would define a function that accepts an argument x and outputs $x + 1$ as so:

$$f(x) = x + 1$$

In math class, we would define a function that accepts an argument x and outputs $x + 1$ as so:

$$f(x) = x + 1$$

In lambda calculus, we do it like this:

$$\lambda$$

In math class, we would define a function that accepts an argument x and outputs $x + 1$ as so:

$$f(x) = x + 1$$

In lambda calculus, we do it like this:

$$\lambda x$$

In math class, we would define a function that accepts an argument x and outputs $x + 1$ as so:

$$f(x) = x + 1$$

In lambda calculus, we do it like this:

$$\lambda x.$$

In math class, we would define a function that accepts an argument x and outputs $x + 1$ as so:

$$f(x) = x + 1$$

In lambda calculus, we do it like this:

$$\lambda x.x + 1$$

In math class, we would define a function that accepts an argument x and outputs $x + 1$ as so:

$$f(x) = x + 1$$

In lambda calculus, we do it like this:

$$\lambda x.x + 1$$

If we wanted to find $4 + 1$, we could do this:

$$f(4) = 4 + 1 = 5$$

In math class, we would define a function that accepts an argument x and outputs $x + 1$ as so:

$$f(x) = x + 1$$

In lambda calculus, we do it like this:

$$\lambda x.x + 1$$

If we wanted to find $4 + 1$, we could do this:

$$f(4) = 4 + 1 = 5$$

In lambda calculus, we apply a value to a function like this:

$$(\lambda x.x + 1)4 = 4 + 1 = 5$$

In math class, we would define a function that accepts an argument x and outputs $x + 1$ as so:

$$f(x) = x + 1$$

In lambda calculus, we do it like this:

$$\lambda x.x + 1$$

If we wanted to find $4 + 1$, we could do this:

$$f(4) = 4 + 1 = 5$$

In lambda calculus, we apply a value to a function like this:

$$(\lambda x.x + 1)4 = 4 + 1 = 5$$

You can think of λ as f, and . as $=$.

Table of Contents

1 Introduction

2 One Argument

3 Booleans

4 Church Numerals

Currying
One Argument

How would we define a function that outputs $x + y$ in math class?

How would we define a function that outputs $x + y$ in math class?

$$f(x, y) = x + y$$

Currying
One Argument

How would we define a function that outputs $x + y$ in math class?

$$f(x, y) = x + y$$

In lambda calculus, functions are only allowed to have one parameter.

Currying
One Argument

How would we define a function that outputs $x + y$ in math class?

$$f(x, y) = x + y$$

In lambda calculus, functions are only allowed to have one parameter.

So, to add two numbers, we have a function output another function, like this:

$$\lambda x.\lambda y.x + y$$

Currying
One Argument

How would we define a function that outputs $x + y$ in math class?

$$f(x, y) = x + y$$

In lambda calculus, functions are only allowed to have one parameter.

So, to add two numbers, we have a function output another function, like this:

$$\lambda x.\lambda y.x + y$$

And we use it like this:

$$(\lambda x.\lambda y.x + y)(2 \quad 3)$$

Currying
One Argument

How would we define a function that outputs $x + y$ in math class?

$$f(x, y) = x + y$$

In lambda calculus, functions are only allowed to have one parameter.

So, to add two numbers, we have a function output another function, like this:

$$\lambda x.\lambda y.x + y$$

And we use it like this:

$$(\lambda x.\lambda y.x + y)(2 \quad 3) = (\lambda y.2 + y)3$$

How would we define a function that outputs $x + y$ in math class?

$$f(x, y) = x + y$$

In lambda calculus, functions are only allowed to have one parameter.

So, to add two numbers, we have a function output another function, like this:

$$\lambda x.\lambda y.x + y$$

And we use it like this:

$$(\lambda x.\lambda y.x + y)(2 \quad 3) = (\lambda y.2 + y)3 = 2 + 3 = 5$$

- Very simple

Why one argument?

One Argument

- Very simple
- Very powerful

Why one argument?

One Argument

- Very simple
- Very powerful
- Functions can only have one variable

But I'm lazy
One Argument

Me too!

But I'm lazy
One Argument

Me too!

We have some shortcuts to help us write down lambda calculus expressions, but it's important to remember what they represent, without the shortcuts.

But I'm lazy
One Argument

Me too!

We have some shortcuts to help us write down lambda calculus expressions, but it's important to remember what they represent, without the shortcuts.

$$\lambda x.\lambda y.\lambda z.A$$

Can be abbreviated as:

$$\lambda xyz.A$$

But I'm lazy

One Argument

Me too!

We have some shortcuts to help us write down lambda calculus expressions, but it's important to remember what they represent, without the shortcuts.

$$\lambda x.\lambda y.\lambda z.A$$

Can be abbreviated as:

$$\lambda xyz.A$$

Also, we assume that we evaluate a function with "multiple" arguments starting with the leftmost parameter.

This is stupid. It just makes everything harder.

One Argument

you're right!

For those examples, you're right!

This is stupid. It just makes everything harder.

One Argument

For those examples, you're right!

Let's get to the fun stuff now!

Table of Contents

1 Introduction

2 One Argument

3 Booleans

4 Church Numerals

Quote

"Any program can be written in lambda calculus."

Quote

"Any program can be written in lambda calculus."

— Me, 5 minutes ago

Quote

"Any program can be written in lambda calculus."

— Me, 5 minutes ago

So, let's bring on the Booleans!

We use the Church Booleans.

We use the Church Booleans.

Definition (TRUE)

$$\mathrm{TRUE} = \lambda xy.x$$

Definition (FALSE)

$$\mathrm{FALSE} = \lambda xy.y$$

We use the Church Booleans.

Definition (TRUE)

$$\mathrm{TRUE} = \lambda xy.x$$

Definition (FALSE)

$$\mathrm{FALSE} = \lambda xy.y$$

So TRUE returns the first value, and FALSE returns the second.

We use the Church Booleans.

Definition (TRUE)

$$\mathrm{TRUE} = \lambda xy.x$$

Definition (FALSE)

$$\mathrm{FALSE} = \lambda xy.y$$

So TRUE returns the first value, and FALSE returns the second.

We will use TRUE and FALSE as shorthand for these definitions.

Definition (NOT)

$$\mathrm{NOT} = \lambda b.b(\mathrm{FALSE}\ \mathrm{TRUE})$$

Definition (NOT)

$$\mathrm{NOT} = \lambda b.b(\mathrm{FALSE}\ \mathrm{TRUE})$$

NOT TRUE

$$(\lambda b.b(\mathrm{FALSE}\ \mathrm{TRUE}))\ \mathrm{TRUE} =$$

Definition (NOT)

$$\text{NOT} = \lambda b.b(\text{FALSE TRUE})$$

NOT TRUE

$$(\lambda b.b(\text{FALSE TRUE}))\ \text{TRUE} = \text{TRUE}(\text{FALSE TRUE})$$

NOT
Booleans

Definition (NOT)

$$\text{NOT} = \lambda b.b(\text{FALSE TRUE})$$

NOT TRUE

$$\begin{aligned}(\lambda b.b(\text{FALSE TRUE}))\ \text{TRUE} &= \text{TRUE}(\text{FALSE TRUE}) \\ &= \lambda xy.x(\text{FALSE TRUE})\end{aligned}$$

Definition (NOT)

$$\mathrm{NOT} = \lambda b.b(\mathrm{FALSE\ TRUE})$$

NOT TRUE

$$\begin{aligned}(\lambda b.b(\mathrm{FALSE\ TRUE}))\ \mathrm{TRUE} &= \mathrm{TRUE}(\mathrm{FALSE\ TRUE}) \\ &= \lambda xy.x(\mathrm{FALSE\ TRUE}) \\ &= \mathrm{FALSE}\end{aligned}$$

Definition (AND)

$$\text{AND} = (\lambda pq.p)(q\ p)$$

Definition (AND)

$$\text{AND} = (\lambda pq.p)(q\ p)$$

AND(TRUE FALSE)

$$((\lambda pq.p)(q\ p))\ (\text{TRUE FALSE})$$

Definition (AND)

$$\text{AND} = (\lambda pq.p)(q\ p)$$

AND(TRUE FALSE)

$$\begin{aligned} & ((\lambda pq.p)(q\ p))\ (\text{TRUE FALSE}) \\ = {} & \text{TRUE}(\text{FALSE TRUE}) \end{aligned}$$

Definition (AND)

$$\text{AND} = (\lambda pq.p)(q\ p)$$

AND(TRUE FALSE)

$$\begin{aligned} & ((\lambda pq.p)(q\ p))\ (\text{TRUE FALSE}) \\ = & \ \text{TRUE}(\text{FALSE TRUE}) \\ = & \ \text{FALSE} \end{aligned}$$

Definition (IFTHENELSE)

$$\text{IFTHENELSE} = (\lambda btf.b)(t\ \ f)$$

Where b is a Boolean, t is the value to return if $b = \text{TRUE}$, and f is the value to return if $b = \text{FALSE}$.

IFTHENELSE(TRUE "Math is great" "Math is kool")

$$((\lambda btf.b)(t\ \ f))\ (\text{TRUE "Math is kool" "Math is lit"})$$

Definition (IFTHENELSE)

$$\text{IFTHENELSE} = (\lambda btf.b)(t\ \ f)$$

Where b is a Boolean, t is the value to return if $b = \text{TRUE}$, and f is the value to return if $b = \text{FALSE}$.

IFTHENELSE(TRUE “Math is great” “Math is kool”)

$$((\lambda btf.b)(t\ \ f))\ (\text{TRUE “Math is kool” “Math is lit”})$$
$$= (\text{TRUE})(\text{“Math is kool” “Math is lit”})$$

Definition (IFTHENELSE)

$$\text{IFTHENELSE} = (\lambda btf.b)(t\ \ f)$$

Where b is a Boolean, t is the value to return if $b = \text{TRUE}$, and f is the value to return if $b = \text{FALSE}$.

IFTHENELSE(TRUE “Math is great” “Math is kool”)

$$\begin{aligned} &((\lambda btf.b)(t\ \ f))\ (\text{TRUE “Math is kool” “Math is lit”}) \\ &= (\text{TRUE})(\text{“Math is kool” “Math is lit”}) \\ &= \text{“Math is kool”} \end{aligned}$$

Table of Contents

1 Introduction

2 One Argument

3 Booleans

4 Church Numerals

We define the naturals, $\mathbb{N}$, using Church numerals.

We define the naturals, $\mathbb{N}$, using Church numerals.

The natural numbers are higher-order functions in lambda calculus.

We define the naturals, $\mathbb{N}$, using Church numerals.

The natural numbers are higher-order functions in lambda calculus.

Definition (Church numerals)

$$0 = \lambda f x.x$$

We define the naturals, $\mathbb{N}$, using Church numerals.

The natural numbers are higher-order functions in lambda calculus.

Definition (Church numerals)

$$
\begin{aligned}
0 &= \lambda f x . x \\
1 &= \lambda f x . f(x)
\end{aligned}
$$

We define the naturals, $\mathbb{N}$, using Church numerals.

The natural numbers are higher-order functions in lambda calculus.

Definition (Church numerals)

$$
\begin{aligned}
0 &= \lambda fx.x \\
1 &= \lambda fx.f(x) \\
2 &= \lambda fx.f(f(x))
\end{aligned}
$$

We define the naturals, $\mathbb{N}$, using Church numerals.

The natural numbers are higher-order functions in lambda calculus.

Definition (Church numerals)

$$
\begin{aligned}
0 &= \lambda f x . x \\
1 &= \lambda f x . f(x) \\
2 &= \lambda f x . f(f(x)) \\
3 &= \lambda f x . f\left(f(f(x))\right)
\end{aligned}
$$

We define the naturals, $\mathbb{N}$, using Church numerals.

The natural numbers are higher-order functions in lambda calculus.

Definition (Church numerals)

$$
\begin{aligned}
0 &= \lambda f x . x \\
1 &= \lambda f x . f(x) \\
2 &= \lambda f x . f(f(x)) \\
3 &= \lambda f x . f\left(f(f(x))\right) \\
4 &= \lambda f x . f\left(f\left(f(f(x))\right)\right)
\end{aligned}
$$

We define the naturals, $\mathbb{N}$, using Church numerals.

The natural numbers are higher-order functions in lambda calculus.

Definition (Church numerals)

$$
\begin{aligned}
0 &= \lambda f x . x \\
1 &= \lambda f x . f(x) \\
2 &= \lambda f x . f(f(x)) \\
3 &= \lambda f x . f\left(f(f(x))\right) \\
4 &= \lambda f x . f\left(f\left(f(f(x))\right)\right) \\
n &= \lambda f x . f^{n}(x)
\end{aligned}
$$

We can add 1 to a Church numeral using the successor function:

Definition (Church numerals)

$$\mathrm{SUCC} = \lambda nfx.f(nfx)$$

$\mathrm{SUCC}\ 0$

$$(\lambda nfx.f(nfx))\ (\lambda fx.x)$$

Successor
Church Numerals

We can add 1 to a Church numeral using the successor function:

Definition (Church numerals)

$$\mathrm{SUCC} = \lambda n f x . f(n f x)$$

SUCC 0

$$\begin{aligned} & (\lambda n f x . f(n f x))\,(\lambda f x . x) \\ = & \;(\lambda f x . f((\lambda f x . x) f x)) \end{aligned}$$

Successor
Church Numerals

We can add 1 to a Church numeral using the successor function:

Definition (Church numerals)

$$\mathrm{SUCC} = \lambda n f x . f(n f x)$$

$\mathrm{SUCC}\ 0$

$$\begin{aligned} & (\lambda n f x . f(n f x))\ (\lambda f x . x) \\ = & (\lambda f x . f((\lambda f x . x) f x)) \\ = & (\lambda f x . f((\lambda x . x) x)) \end{aligned}$$

We can add 1 to a Church numeral using the successor function:

Definition (Church numerals)

$$\mathrm{SUCC} = \lambda nfx.f(nfx)$$

$\mathrm{SUCC}\ 0$

$$\begin{aligned}
&(\lambda nfx.f(nfx))\,(\lambda fx.x)\\
&= (\lambda fx.f((\lambda fx.x)fx))\\
&= (\lambda fx.f((\lambda x.x)x))\\
&= \lambda fx.f(x)
\end{aligned}$$

We can add 1 to a Church numeral using the successor function:

Definition (Church numerals)

$$\mathrm{SUCC} = \lambda n f x . f(n f x)$$

$\mathrm{SUCC}\ 0$

$$\begin{aligned} & (\lambda n f x . f(n f x))\ (\lambda f x . x) \\ = {} & (\lambda f x . f((\lambda f x . x) f x)) \\ = {} & (\lambda f x . f((\lambda x . x) x)) \\ = {} & \lambda f x . f(x) \\ = {} & 1 \end{aligned}$$