Lambda Calculus

Alonzo Church

- Computability, Logic
- Lambda Calculus

- Why Lambda?

Only 3 "Interesting Parts" of Any PL

- Variables:
  - `x, y, z`
- Function Definitions:
  - `λx.x+3 == (lambda (x) (+ x 3))`
- Function Application:
  - `(λx.x+3) 6 → 6+3`
  - `(λa.(λb.a*b)) 3 4 → (λb.3*b) 4`
    `→ 3*4`

Really, That's It?

```
int f(int x) { return x + 10; }
f(5);

let f = λx.x+10  in f(5)


(λf.f(5))(λx.x+10)
```

```
(λf.λx.f(f x))


(defun twice (f)
   (lambda (x) (f (f x)))
```

Paren Rules

- `A B C = (A B) C`
- `λx.xy = λx.(xy)`

$$
\begin{aligned}
& \underline{(\lambda f.\lambda x.f(f\ x))}\ \underline{(\lambda y.y+1)}\ \underline{2} \\
= \;& (\underline{(\lambda f.\lambda x.f(f\ x))}\ \underline{(\lambda y.y+1)})\ 2 \\
\rightarrow \;& (\lambda x.(\lambda y.y+1)(\underline{(\lambda y.y+1)}\ \underline{x}))\ 2 \\
\rightarrow \;& (\lambda x.\underline{(\lambda y.y+1)}\ \underline{(x+1)})\ 2 \\
\rightarrow \;& \underline{(\lambda x.(x+1)+1)}\ \underline{2} \\
\rightarrow \;& (2+1)+1
\end{aligned}
$$

```
(λf.λx.f(f x))(λz.x+z)2 =
(λf.λa.f(f a))(λz.x+z)2 ->     (rename x)
([(λz.x+z)/f]λa.f(f a))2 =     (reduce f)
(λa.(λz.x+z)((λz.x+z)a))2 =    (substitution)
(λa.(λb.x+b)((λz.x+z)a))2 ->   (rename z)
(λa.(λb.x+b)([a/z](x+z))2 =    (reduce z)
(λa.(λb.x+b)(x+a))2 ->         (substitution)
(λa.[(x+a)/b](x+b))2 =         (reduce b)
(λa.x+(x+a))2 ->               (substitution)
[2/a](x+(x+a))=                (reduce a)
x+x+2                          (substitution)
```

```
(λf.λx.f(f x))(λz.x+z)2 =
(λf.λa.f(f a))(λz.x+z)2 ->     (rename x)
([(λz.x+z)/f]λa.f(f a))2 =     (reduce f)
(λa.(λz.x+z)((λz.x+z)a))2 =    (substitution)
(λa.(λb.x+b)((λz.x+z)a))2 ->   (rename z)
```

Try a different step.
What is the final result?

```
(λf.λx.f(f x))(λz.x+z)2 =
(λf.λa.f(f a))(λz.x+z)2 ->     (rename x)
([(λz.x+z)/f]λa.f(f a))2 =     (reduce f)
(λa.(λz.x+z)((λz.x+z)a))2 =    (substitution)
(λa.(λb.x+b)((λz.x+z)a))2 ->
(λa.(λb.x+b)(x+a))2 ->
(λa.x+x+a)2 ->
 x+x+2
```

```
(λf.λx.f(f x))(λz.x+z)2 =
(λf.λa.f(f a))(λz.x+z)2 ->     (rename x)
([(λz.x+z)/f]λa.f(f a))2 =     (reduce f)
(λa.(λz.x+z)((λz.x+z)a))2 =    (substitution)
(λa.(λb.x+b)((λz.x+z)a))2 ->
(λa.x+((λz.x+z)a))2 ->
(λa.x+(x+a))2 ->
 x+x+2
```

```
(λf.λx.f(f x))(λz.x+z)2 =
(λf.λa.f(f a))(λz.x+z)2 ->     (rename x)
([(λz.x+z)/f]λa.f(f a))2 =     (reduce f)
(λa.(λz.x+z)((λz.x+z)a))2 =    (substitution)
(λa.(λb.x+b)((λz.x+z)a))2 ->
(λb.x+b)((λz.x+z)2)->
(λb.x+b)(λz.x+2)->
 x+x+2
```

Confluence

$$\begin{tikzcd}
 & (\lambda y.y+1)((\lambda x.x+x)2) \arrow[dl] \arrow[dr] & \\
((\lambda x.x+x)2)+1 \arrow[dr] & & (\lambda y.y+1)(2+2) \arrow[dl] \\
 & 2+2+1 &
\end{tikzcd}$$

Confluence

$$\begin{tikzcd}
 & M \arrow[dl] \arrow[dr] & \\
M_1 \arrow[dr, dashed] & & M_2 \arrow[dl, dashed] \\
 & N &
\end{tikzcd}$$

If $M \rightarrow M_1$ and $M \rightarrow M_2$ then $M_1 \rightarrow^* N$ and $M_2 \rightarrow^* N$ for some N

Is Functional Programming Better?

Pros?
Cons?