# Church Numerals

## Encoding natural numbers as lambda-terms

$$
\begin{aligned}
\text{zero} &= \lambda f.\lambda x.x \\
\text{one} &= \lambda f.\lambda x.f\ x \\
\text{two} &= \lambda f.\lambda x.f\,(f\ x) \\
\text{succ} &= \lambda n.\lambda f.\lambda x.f\,(n\ f\ x) \\
\text{plus} &= \lambda n.\lambda m.n\ \text{succ}\ m \\
\text{times} &= \lambda n.\lambda m.n\ (\text{plus}\ m)\ \text{zero}
\end{aligned}
$$

**Idea: “apply $f$ to $x$, $n$ times”**

# Church Numerals to machine integers

```
; uscheme or possibly uhaskell
-> (val add1 ((curry +) 1))
-> (define to-int (n)
             ((n add1) 0))
-> (to-int three)
3
-> (to-int ((times three) four))
12
```

# Church Numerals in $\lambda$

```
<0>   = \f.\x.x;
succ  = \n.\f.\x.f (n f x);
plus  = \n.\m.n succ m;
times = \n.\m.n (plus m) <0>;
 ...
-> <4>;
\f.\x.f (f (f (f x)))
-> <3>;
\f.\x.f (f (f x))
-> times <4> <3>;
\f.\x.f (f (f (f (f (f (f (f (f (f (f (f x)))))))))))
```

# Reduction rules

**Central rules: substitution and optimization:**

$$\frac{}{(\lambda x.M)N \xrightarrow{\beta} M[x \mapsto N]} \text{ (Beta)} \qquad \frac{x \text{ not free in } M}{(\lambda x.Mx) \xrightarrow{\eta} M} \text{ (Eta)}$$

**Structural rules: Reduce anywhere, any time**

$$\frac{M \to M'}{MN \to M'N} \text{ (Nu)} \quad \frac{N \to N'}{MN \to MN'} \text{ (Mu)} \quad \frac{M \to M'}{\lambda x.M \to \lambda x.M'} \text{ (Xi)}$$

**(Good for both $\beta$ and $\eta$.)**

# Idea: normal form

A term is a normal form if

It cannot be reduced

What do you suppose it means to say

- A term has no normal form?
- A term has a normal form?

# Idea: normal form

A term **is** a normal form if

It cannot be reduced

A term **has** a normal form if

There **exists** a sequence of reductions that terminates (in a normal form)

A term **has no normal form** if

It always reduces forever

(This term **diverges**)

# Normal forms code for values

**Corollary of Church-Rosser:**

**if $A \to^* B$, $B$ in normal form, and**

**$A \to^* C$, $C$ in normal form**

**then $B$ and $C$ are identical**

**(up to renaming of bound variables)**