Review: Church Encodings

```
true  = \x.\y.x;              // Booleans
false = \x.\y.y;

pair  = \x.\y.\f.f x y;       // pairs
fst   = \p.p (\x.\y.x);
snd   = \p.p (\x.\y.y);

noreduce bot   = (\x.x x)(\x.x x);  // divergence

                              // S-expressions
         nil   =          \n.\c.n;
         cons  = \y.\ys.\n.\c.c y ys;
         null? = \xs.xs true (\y.\ys.false);
noreduce car   = \xs.xs bot  (\y.\ys.y);
noreduce cdr   = \xs.xs bot  (\y.\ys.ys);
```

Review: Church Numerals

```
zero  = \f.\x.x;
succ  = \n.\f.\x.f (n f x);
plus  = \n.\m.n succ m;
times = \n.\m.n (plus m) zero;
 ...
-> four;
\f.\x.f (f (f (f x)))
-> three;
\f.\x.f (f (f x))
-> times four three;
\f.\x.f (f (f (f (f (f (f (f (f (f (f (f x))))))))))))
```

Reduction rules

Central rules: substitution and optimization:

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

Structural rules: Reduce anywhere, any time

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

(Good for both β and η.)

Free variables

$$\frac{}{x \text{ is free in } x} \qquad \frac{x \text{ is free in } M \qquad x \neq x'}{x \text{ is free in } \lambda x'.M}$$

$$\frac{x \text{ is free in } M}{x \text{ is free in } MN} \qquad \frac{x \text{ is free in } N}{x \text{ is free in } MN}$$

Your turn! Free Variables

What are the free variables in each expression?

```
\x.\y. y z
\x.x (\y.x)
\x.\y.\x.x y
\x.\y.x (\z.y w)
y (\x.z)
(\x.\y.x y) y
```

Your turn! Free Variables

What are the free variables in each expression?

```
\x.\y. y z                  - z
\x.x (\y.x)                 - nothing
\x.\y.\x.x y                - nothing
\x.\y.x (\z.y w)            - w
y (\x.z)                    - y z
(\x.\y.x y) y               - y
```

Capture-avoiding substitution

$$
\begin{aligned}
x[x \mapsto M] &= M \\
y[x \mapsto M] &= y \\
(YZ)[x \mapsto M] &= (Y[x \mapsto M])(Z[x \mapsto M]) \\
(\lambda x.Y)[x \mapsto M] &= \lambda x.Y \\
(\lambda y.Z)[x \mapsto M] &= \lambda y.Z[x \mapsto M] \\
&\quad \text{if } x \text{ not free in } Z \text{ or } y \text{ not free in } M \\
(\lambda y.Z)[x \mapsto M] &= \lambda w.(Z[y \mapsto w])[x \mapsto M] \\
&\quad \text{where } w \text{ not free in } Z \text{ or } M
\end{aligned}
$$

Last transformation is renaming of bound variables

Renaming of bound variables

So important it has its own Greek letter:

$$\frac{w \text{ not free in } Z}{\lambda y.Z \xrightarrow{\alpha} \lambda w.(Z[y \mapsto w])} \qquad \textbf{(ALPHA)}$$

Also has structural rules

Conversion and reduction

Alpha-conversion (rename bound variable)

$$\frac{y \text{ not free in } Z}{\lambda x.Z \xrightarrow{\alpha} \lambda y.Z[x \mapsto y]}$$

Beta-reduction (the serious evaluation rule)

$$(\lambda x.M)N \xrightarrow{\beta} M[x \mapsto N]$$

Eta-reduction:

$$\frac{x \text{ not free in } M}{\lambda x.Mx \xrightarrow{\eta} M}$$

All structural: Convert/reduce whole term *or* subterm

Church-Rosser Theorem

Equivalence of convertible terms:

if $A \rightarrow B$ **and** $A \rightarrow C$

there exists D **s.t.** $B \rightarrow^* D$ **and** $C \rightarrow^* D$

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 $A \rightarrow^* B$, B in normal form, and

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

then B and C are identical

(up to renaming of bound variables)

Y combinator can implement `fix`

Define Y such that, for any g, $Y\,g = g\,(Y\,g)$:

$$\begin{aligned} Y &= \lambda f.(\lambda x.f(xx))(\lambda x.f(xx)) \\ Y\,g &= (\lambda x.g(xx))(\lambda x.g(xx)) \end{aligned}$$

and by beta-conversion

$$\begin{aligned} Y\,g &= g\,((\lambda x.g(xx))(\lambda x.g(xx))) \\ Y\,g &= g\,(Y\,g) \end{aligned}$$

so

$Y\,g$ is a fixed point of g

Does $Y\,g$ have a normal form?

Normal-order reduction

(If a normal form exists, find it!)

Application offers up to three choices:

#1 $$\frac{}{(\lambda x.M)N \xrightarrow{\beta} M[x \mapsto N]}\ (\text{BETA}) \qquad \frac{x \text{ not free in } M}{(\lambda x.Mx) \xrightarrow{\eta} M}\ (\text{ETA})$$

#2 $$\frac{M \to M'}{MN \to M'N}\ (\text{NU}) \qquad \#3\ \frac{N \to N'}{MN \to MN'}\ (\text{MU}) \qquad \frac{M \to M'}{\lambda x.M \to \lambda x.M'}\ (\text{XI})$$

Slogan: “leftmost, outermost redex”

Normal-order illustration

Not every term has a normal form:

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

But

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

Think "bodies before arguments"

Applicative order does not terminate!