Church's λ-calculus

1 Syntax

Assume we have a countable set of (variable) names, which we shall denote by (possibly indexed) small letters - $a, b, c, ..., x, y, z, a_0, a_1, a_2, ...$

Definition: $\lambda - term$

```
<term> := <name>
        | (λ <name>.<term>)
        | (<term><term>)
```

Conventions

1. Function application (3rd line) is left-associative, so $(((A_1A_2)A_3)...A_k)$
2. Nested abstractions (2nd line) $(\lambda x_1.(\lambda x_2.(...\lambda x_k.A)...))$ can be abbreviated as $\lambda x_1x_2...x_k.A$

2 Free and Bound Variables

Free variables:

1. $< name >$ is free in $< name >$
2. $< name >$ is free in $\lambda < name' > . < term >$ if $< name > \neq < name' >$ and $< name >$ is free in $< term >$
3. $< name >$ is free in $< term' >< term'' >$ if $< name >$ is free in $< term' >$ or $< name >$ is free in $< term'' >$

Bound variables

1. $< name >$ is bound in $\lambda < name' > . < term >$ if $< name > = < name' >$ or $< name >$ is bound in $< term >$
2. $< name >$ is bound in $< term' >< term'' >$ if $< name >$ is bound in $< term' >$ or $< name >$ is bound in $< term'' >$

3 Reductions

Denote by $T[x := R]$ the term in which for every free occurrence of x is replaced by E

α**-conversion**: Bound variables can be renamed: $\lambda x.F \equiv \lambda t.F[x := t]$ provided t is not free in F and t is not bound by λ in F whenever it replaces an x. Example: $\lambda x.yx(\lambda x.xx)zx \equiv \lambda t.yt(\lambda x.xx)zt$.

β**-reduction**: Applying a function to the argument $(\lambda x.F)A \equiv F[x := A]$ provided all free occurrences in A remain free in $F[x := A]$

Definition: Normal Form

A λ-term is in normal form if no β reduction can be applied to it

Theorem If a λ-term has a normal form then the formal for is unique (up to renaming of bound variables)

Computing the normal form: Keep on replacing the leftmost bound variable by the actual argument until no further reduction is possible. This does not terminate iff the initial term has no normal form.

4 Church Numerals

The Church Numerals $C_0, C_1, C_2, ...$ are defined as follows

$$\begin{aligned} C_0 &\equiv \lambda sz.z \\ C_1 &\equiv \lambda sz.s(z) \\ C_2 &\equiv \lambda sz.s(s(z)) \\ \dots & \quad \dots \end{aligned}$$

The successor can be defined as

$$S = \lambda uvw.v(uvw)$$

Lemma. For every two terms in F and A, $C_n FA = F^{(n)}A$
Corollary. Doing addition in λ-calculus: $C_n S C_m = C_{n+m}$

5 Predecessor is hard

Define true and false by

$$\begin{aligned} T &\equiv \lambda xy.x \\ F &\equiv \lambda xy.y \end{aligned}$$

and represent a pair (a, b) by $\lambda z.zab$ Define

$$\Phi \equiv \lambda pz.z(S(pT))(pT)$$

And finally the predecessor is defined as

$$P \equiv \lambda n.n\Phi\,(\lambda z.zC_0C_0)\,F$$