The Untyped Lambda-Calculus, cont.

Let C be a cartesian closed category with an object X such that

$$X = \hom(X,X)$$

(for example the free such category). We can "build a computer" in C. Any morphism $f: X \to X$ defines an "element"

$$\text{"}f\text{"}: 1 \longrightarrow \hom(X,X) = X.$$

Indeed, elements of X are in 1-1 correspondence with morphisms $f: X \to X$, since currying is an isomorphism. We'll describe such morphisms $f: X \to X$ using λ-calculus expressions, which are called <u>λ-terms</u>.

Last time we defined

$$T := \quad x \longmapsto (y \longmapsto x)$$
$$F := \quad x \longmapsto (y \longmapsto y)$$

These are morphisms from X to $\hom(X,X) = X$. We also defined

$$\& := \quad (x,y) \longmapsto x(y)(F)$$

which is a morphism from $X \times X$ to X, where things like $x(y)$ are defined using

$$X \times X = \hom(X,X) \times X \xrightarrow{ev} X$$

Warmup exercise: what's $T(a)$?

$$T(a) = (x \mapsto (y \mapsto x))(a) \quad \text{β-reduction}$$
$$= y \mapsto a$$

$$F(a) = (x \mapsto (y \mapsto y))(a)$$
$$= y \mapsto y$$

Now let's check that & works as it should by computing $\&(T,T)$, a.k.a. $T \,\&\, T$, etc.

$$T \,\&\, T = ((x,y) \mapsto x(y)(F))(T,T) \quad \text{β-reduction}$$
$$= T(T)(F)$$
$$= (y \mapsto T)(F)$$
$$= T \quad \checkmark$$

$$T \,\&\, F = T(F)(F)$$
$$= (y \mapsto F)(F)$$
$$= F \quad \checkmark$$

$$
\begin{aligned}
F \,\&\, T &= F(T)(F) \\
&= (y \mapsto y)(F) \quad \text{by the warmup exercise} \\
&= F \quad \checkmark
\end{aligned}
$$

$$
\begin{aligned}
F \,\&\, F &= F(F)(F) \\
&= (y \mapsto y)(F) \\
&= F \quad \checkmark
\end{aligned}
$$

Exercise: Concoct similar tricks to define the Boolean operations $\neg$, $\vee$, $\Rightarrow$.

Challenge: Understand these "tricks" and make them comprehensible, if possible.

Another operation that's very handy:

$$\text{if — then — else —}.$$

This is given by

$$(x, y, z) \longmapsto x(y)(z).$$

Let's check:

$$
\begin{aligned}
\text{if } T \text{ then } A \text{ else } B &= T(A)(B) = (y \mapsto A)(B) = A \quad \checkmark \\
\text{if } F \text{ then } A \text{ else } B &= F(A)(B) = (y \mapsto y)(B) = B. \quad \checkmark
\end{aligned}
$$

Note:

$$A \,\&\, B = \text{if } A \text{ then } B \text{ else } F$$

which makes sense.

Now let's define functions of Church numerals. If one goes far enough, one can write any partial recursive fn. $f : \mathbb{N} \xrightarrow{\circ} \mathbb{N}$ as a λ-term. Recall for any $n \in \mathbb{N}$ the Church numeral

$$\bar{n} : \hom(X, X) \longrightarrow \hom(X, X)$$

is "raising to the nth power":

$$\bar{n} = f \longmapsto f^n$$

or, in haute λ-calculusese,

$$\bar{n} = f \mapsto (x \mapsto \underbrace{f(f(f \cdots f}_{n}(x)))))$$

How do we define addition of Church numerals?

$$+ = (a, b) \longmapsto (f \mapsto a(f) \circ b(f))$$

where we defined $\circ$ earlier:

$$\circ = (f, g) \longmapsto (x \mapsto f(g(x)))$$

Check:

$$\bar{n} + \bar{m} = +(\bar{n},\bar{m}) = f \mapsto \bar{n}(f) \circ \bar{m}(f)$$
$$= f \mapsto f^n \circ f^m$$
$$= f \mapsto f^{n+m}$$
$$= \overline{n+m}$$

How about multiplication?

$$\cdot = (a,b) \longmapsto (f \mapsto a(b(f))$$
$$= (a,b) \longmapsto a \circ b$$

Check:

$$\bar{n} \cdot \bar{m} = \cdot(\bar{n},\bar{m}) = f \mapsto \bar{n}(\bar{m}(f))$$
$$= f \mapsto (f^m)^n$$
$$= f \mapsto f^{mn}$$
$$= \overline{mn}$$

Lots of fancier functions are defined recursively, like

$$n! = \begin{cases} 1 & \text{if } n=0 \\ n(n-1)! & \text{else} \end{cases}$$

So we'd like to say

$$\text{factorial} = (\text{if } n==0 \text{ then } 1 \text{ else } n \cdot \text{factorial}(n-1))$$

To get this to work we need to

1) Define an operation $x==0$ which takes values T or F.

2) Define an operation $x-1$ which does the right thing for church numerals $\bar{n}$ with $n>0$.

3) Figure out how to convert our recursive definition into a λ-term.

I'll leave 1) & 2) as exercises and do 3). For 3), we're given a definition of factorial that's a fixed-point equation: given a method f for turning fns into fns we seek a function a s.t. $a = f(a)$. For this we prove an astounding theorem.....

Fixed Point Theorem: Suppose f is an element of $\hom(X,X) = X$ defined by some λ-term. Then there exists a fixed point for f, namely an element a of X defined by some λ-term, with

$$f(a) = a.$$

Proof: We actually construct a from f. Let

$$A = \quad x \longmapsto f(x(x))$$

and let

$$a = A(A).$$

Let's check that it works:

$$a = A(A) = f(A(A)) = f(a) \qquad \blacksquare \quad (!)$$

This fiendish argument is probably due to Church & Turing. Applying this to

$$\text{factorial} = f(\text{factorial})$$

where

$$f(g) = (x \mapsto \text{if } x{=}{=}0 \text{ then } 1 \text{ else } x \cdot g(x-1))$$

the theorem gives an explicit λ-term for the factorial function.

Masochistic but useful exercise — use this to compute $3!$.