Informal notes on the Y combinator

Deian Stefan

February 19, 2017

Suppose we want to implement the factorial function in λ calculus. This function is recursive and thus far, we have not defined any recursive functions in the λ-calculus. Indeed, this is not immediately clear how to do—since we don't have a way to name the "current" function in λ-calculus, we don't have way to call it recursively. So, how can we do this?

Well, from the previous exercises we know that there are λ-terms that can reduce indefinitely. Ω, for example, always reduces to itself—i.e., $\Omega =_\beta \Omega$. The other, more interesting term is $\mathbf{Y}$. $\mathbf{Y}$ has the property that $\mathbf{Y}\ f =_\beta f\ (\mathbf{Y}\ f)$ for any f, i.e., $\mathbf{Y}\ f$ is the *fixed point* of f. It is precisely this property that we need to define recursive recursive functions in λ-calculus!

To start, let:

$$f \triangleq \lambda \text{fac}.\lambda n.\text{if } n \leq 1 \text{ then } 1 \text{ else } n * (\text{fac } (n-1))$$

Note that f is *not* the factorial function—f is a high-order function that takes a function fac and returns a function that itself takes a value n as an argument and, depending on n either returns 1 or returns the multiplication of n with the result from calling the fac function with $n - 1$.

We're now going to use the $\mathbf{Y}$ combinator to define the factorial function as:

$$\text{factorial} \triangleq \mathbf{Y}\ f$$

How do we know that this is actually the factorial function? By equational reasoning:

$$\begin{aligned}
\text{factorial} &= \mathbf{Y}\ f \\
&=_\beta f\ (\mathbf{Y}\ f) \\
&= f\ \text{factorial} \\
&= (\lambda \text{fac}.\lambda n.\text{if } n \leq 1 \text{ then } 1 \text{ else } n * (\text{fac } (n-1)))\ \text{factorial} \\
&=_\beta \lambda n.\text{if } n \leq 1 \text{ then } 1 \text{ else } n * (\text{factorial } (n-1))
\end{aligned}$$

Naturally, you can use the above definition to actually do calculation, say, compute the factorial of 2:

$$
\begin{aligned}
\text{factorial } 2 &= (\mathbf{Y}\ f)\ 2 \\
&=_\beta (f\ (\mathbf{Y} f))\ 2 \\
&= ((\lambda \text{fac}.\lambda n.\text{if } n \leq 1 \text{ then } 1 \text{ else } n * (\text{fac}\ (n-1)))\ (\mathbf{Y}\ f))\ 2 \\
&=_\beta (\lambda n.\text{if } n \leq 1 \text{ then } 1 \text{ else } n * ((\mathbf{Y}\ f)\ (n-1)))\ 2 \\
&=_\beta \text{if } 2 \leq 1 \text{ then } 1 \text{ else } 2 * ((\mathbf{Y}\ f)\ (2-1)) \\
&=_\beta \text{if } 2 \leq 1 \text{ then } 1 \text{ else } 2 * ((\mathbf{Y}\ f)\ 1) \\
&=_\beta 2 * ((\mathbf{Y}\ f)\ 1) \\
&=_\beta 2 * ((f\ (\mathbf{Y}\ f))\ 1) \\
&= 2 * (((\lambda \text{fac}.\lambda n.\text{if } n \leq 1 \text{ then } 1 \text{ else } n * (\text{fac}\ (n-1)))\ (\mathbf{Y}\ f))\ 1) \\
&=_\beta 2 * ((\lambda n.\text{if } n \leq 1 \text{ then } 1 \text{ else } n * ((\mathbf{Y}\ f)\ (n-1)))\ 1) \\
&=_\beta 2 * (\text{if } 1 \leq 1 \text{ then } 1 \text{ else } 1 * ((\mathbf{Y}\ f)\ (1-1))) \\
&=_\beta 2 * 1 \\
&=_\beta 2
\end{aligned}
$$

There is a deeper meaning to all of this, but we will not explore it in these notes. I recommend you re-read section 4.2 from the textbook and lookup the meaning of fix points.