Informal notes on the \( Y \) combinator

Deian Stefan

February 19, 2017

Suppose we want to implement the factorial function in \( \lambda \) calculus. This function is recursive and thus far, we have not defined any recursive functions in the \( \lambda \)-calculus. Indeed, this is not immediately clear how to do—since we don’t have a way to name the “current” function in \( \lambda \)-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 \( \lambda \)-terms that can reduce indefinitely. \( \Omega \), for example, always reduces to itself—i.e., \( \Omega =_{\beta} \Omega \). The other, more interesting term is \( Y \). \( Y \) has the property that \( Y f =_{\beta} f (Y f) \) for any \( f \), i.e., \( Y f \) is the fixed point of \( f \). It is precisely this property that we need to define recursive recursive functions in \( \lambda \)-calculus!

To start, let:

\[
f \triangleq \lambda f. \lambda n. \text{if } n \leq 1 \text{ then } 1 \text{ else } n \ast (f (n - 1))
\]

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 \( Y \) combinator to define the factorial function as:

\[
\text{factorial} \triangleq Y f
\]

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

\[
\begin{align*}
\text{factorial} & = Y f \\
& =_{\beta} f (Y f) \\
& = f \text{ factorial} \\
& = (\lambda f. \lambda n. \text{if } n \leq 1 \text{ then } 1 \text{ else } n \ast (f (n - 1))) \text{ factorial} \\
& =_{\beta} \lambda n. \text{if } n \leq 1 \text{ then } 1 \text{ else } n \ast (\text{factorial} (n - 1))
\end{align*}
\]

Naturally, you can use the above definition to actually do calculation, say, compute the factorial of 2:
factorial 2 = $(Y\ f)\ 2$

$$=_{\beta} (f\ (Yf))\ 2$$

$$= ((\lambda n.\ if\ n \leq 1\ then\ 1\ else\ n \ast (\text{fac}\ (n - 1)))\ (Y\ f))\ 2$$

$$=_{\beta} ((\lambda n.\ if\ n \leq 1\ then\ 1\ else\ n \ast ((Y\ f)\ (n - 1)))\ 2$$

$$=_{\beta} \text{if } 2 \leq 1\ then\ 1\ else\ 2 \ast ((Y\ f)\ (2 - 1))$$

$$=_{\beta} \text{if } 2 \leq 1\ then\ 1\ else\ 2 \ast ((Y\ f)\ 1)$$

$$=_{\beta} 2 \ast ((Y\ f)\ 1)$$

$$=_{\beta} 2 \ast ((f\ (Yf))\ 1)$$

$$= 2 \ast ((\lambda n.\ if\ n \leq 1\ then\ 1\ else\ n \ast (\text{fac}\ (n - 1)))\ (Y\ f))\ 1)$$

$$=_{\beta} 2 \ast ((\lambda n.\ if\ n \leq 1\ then\ 1\ else\ n \ast ((Y\ f)\ (n - 1)))\ 1)$$

$$=_{\beta} 2 \ast (\text{if } 1 \leq 1\ then\ 1\ else\ 1 \ast ((Y\ f)\ (1 - 1)))$$

$$=_{\beta} 2 \ast 1$$

$$=_{\beta} 2$$

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.