Explaining **Y**

John Peloquin

In the lambda calculus, given a term F, how can we find a *fixed point* for F—that is, a term x such that $Fx = x$? Well, if infinite terms were allowed, we could just take

$$x = F(F(\cdots))$$

with F repeated infinitely many times. Because then we would have

$$Fx = F(F(F(\cdots))) = x$$

Sadly, infinite terms are not allowed. However, we can achieve the desired goal by constructing a term which "generates" infinitely many F's. First consider

$$\omega = \lambda x.xx$$

and

$$\Omega = \omega\omega$$

Then Ω β-reduces to itself infinitely since

$$\begin{aligned}
\Omega &= \omega\omega && \text{by definition of } \Omega \\
&= (\lambda x.xx)\omega && \text{by definition of } \omega \\
&= \omega\omega && \text{by } \beta\text{-reduction} \\
&= \Omega \\
&= \cdots
\end{aligned}$$

If we introduce F into ω, we can generate an F at each β-reduction step of Ω. Let $\omega_F = \lambda x.F(xx)$ and $\Omega_F = \omega_F\omega_F$. Then

$$\Omega_F = \omega_F\omega_F = (\lambda x.F(xx))\omega_F = F(\omega_F\omega_F) = F(\Omega_F)$$

In other words, $x = \Omega_F$ is a fixed point of F.

We can generalize this and define the fixed-point combinator

$$\mathbf{Y} = \lambda f.\,\Omega_f = \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$$

So $\mathbf{Y}F = F(\mathbf{Y}F)$ is a fixed point of F for any F. This is the $\mathbf{Y}$ combinator.

The $\mathbf{Y}$ combinator is important because it lets us define *recursive* functions in the lambda calculus. For example, if we want G such that $GX = X(XG)$ for all X, we would have it if

$$G = \lambda x.x(xG) = (\lambda gx.x(xg))G$$

So we can just take $G = \mathbf{Y}(\lambda gx.x(xg))$.

The $\mathbf{Y}$ combinator is not the only fixed-point combinator. Above, we might have interchanged the order of λf and λx to obtain

$$A = \lambda xf.f(xxf)$$

and

$$\Theta = AA$$

Then

$$\Theta F = AAF = (\lambda f.f(AAf))F = F(AAF) = F(\Theta F)$$

This is Turing's fixed-point combinator, which has certain advantages over the $\mathbf{Y}$ combinator. There are infinitely many more.

References

[1] Barendregt, H. and E. Barendsen. *Introduction to Lambda Calculus.* 2000.