

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

$\Omega = \omega\omega$	by definition of Ω
$= (\lambda x.xx)\omega$	by definition of ω
$= \omega\omega$	by β -reduction
$= \Omega$	
$= \cdots$	

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 $\Omega_F = \omega_F\omega_F$ (where x does not occur free in 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 **Y** combinator.

The **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

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

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

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

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

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 **Y** combinator. There are infinitely many more.

References

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