Arithmetic on Church numerals

Jules Jacobs

April 17, 2021

Church naturals allow us to represent numbers in pure lambda calculus. In this short note I'll explain how to define addition, multiplication, and power on Church nats. As a bonus, I'll show how to define fast growing functions.

Church repesents a natural number n as a higher order function, which I'll denote $[n]$. The function $[n]$ takes another function f and composes f with itself n times:

$$[n]\, f = \underbrace{f \circ f \cdots \circ f}_{n \text{ times}} = f^n$$

We can convert a Church nat a back to an ordinary nat by applying it to the successor function $S\ n = n+1$: if $a = [n]$ then $a\ s\ 0$ gives us back ordinary natural number n because $a\ s\ 0$ is the n-fold application of the successor function to the number 0, which just increments it n times.

The first few Church natural numbers are:

$$\begin{aligned}
[0] &= \lambda f.\lambda z.z \\
[1] &= \lambda f.\lambda z.fz \\
[2] &= \lambda f.\lambda z.f(fz) \\
[3] &= \lambda f.\lambda z.f(f(fz))
\end{aligned}$$

Many descriptions of Church nats will view them in that way: as a function that takes *two* arguments f and z that computes $f(f(\ldots(fz)\ldots))$, but point of view gets incredibly confusing when you try to define arithmetic on them, particularly multiplication and power. So think about $[n]f = f^n$ as performing n-fold function composition.

Let's first define the successor function on Church nats:

$$[n+1]\, f = f^{n+1} = f \circ f^n = f \circ ([n]\, f)$$

So if a is a Church nat, then the successor is defined as

$$s\ a = \lambda f.f \circ (af) = \lambda f.\lambda z.f(afz)$$

Addition is also fairly easy:

$$[n+m]\, f = f^{n+m} = f^n \circ f^m = ([n]\, f) \circ ([m]\, f)$$

So if a, b are Church nats, then addition is defined as

$$a + b = \lambda f.(af) \circ (bf) = \lambda f.\lambda z.a(bf)$$

Multiplication is not much harder:

$$[n \cdot m]f = f^{n \cdot m} = (f^n)^m = [m]\,([n]\, f)$$

So if a, b are Church nats, then multiplication is defined as

$$a \cdot b = \lambda f.a(bf)$$

Power is a bit trickier:

$$[n^m]f = f^{(n^m)} = f^{\overbrace{n \cdot n \cdots n}^{m \text{ times}}} = (((f^n)^n)^n \cdots)^n = [n]\ ([n]\ (\cdots [n]\ f)) = ([m]\ [n])\ f$$

So if a, b are Church nats, then power is defined as

$$a^b = \lambda f.(a\ b)f = a\ b$$

Nice! If that explanation was confusing, here's another one. If we apply $b\ f^k$ we get $f^{b \cdot k}$, because the Church nat b composes f^k with itself b times. Therefore $b\ (b\ f^k) = f^{b^2 \cdot k}$, and so on. Therefore, $b\ (b\ \cdots (bf)) = f^{(b^a)}$. But applying the function b an a number of times, is precisely what the action of a as a Church nat is. So $(a\ b)f = f^{(a^b)}$ performs power, so $a^b = a\ b$.

Given any function $g : N \to N$ we can define a series of ever faster growing functions as follows:

$$f_0(n) = g(n)$$
$$f_{k+1}(n) = f_k^n(n)$$

We can define this function using Church naturals:

$$f_k = k\ (\lambda f.\lambda n.nf n)\ g$$

If we take $g = s$ then,

$$f_0(n) = n + 1$$
$$f_1(n) = 2n$$
$$f_2(n) = 2^n \cdot n$$

The function $A(n) = f_n(n)$ grows pretty quickly. We can play the same game again, by putting $g = A$, obtaining a sequence:

$$h_0(n) = A(n)$$
$$h_{k+1}(n) = h_k^n(n)$$

To get a feeling for how fast this grows, consider h_1:

$$\begin{aligned} h_1(n) &= h_0^n(n) \\ &= A(A(A(\ldots A(A(n))))) \\ &= A(A(A(\ldots A(f_n(n))))) \\ &= A(A(A(\ldots f_{f_n(n)}(f_n(n))))) \end{aligned}$$

An expression like $h_3(3)$ gives us a relatively short lambda term that will normalise to a huge term. We might as well start with $g(n) = n^n$ since that's even easier to write using Church naturals:

$$
\begin{aligned}
g &= \lambda a.a\ a \\
A &= \lambda k.k\ (\lambda f.\lambda n.nf\,n)\ g \\
h &= \lambda k.k\ (\lambda f.\lambda n.nf\,n)\,A \\
3 &= \lambda f.\lambda z.f(f(f\ z)) \\
X &= h\ 3\ 3
\end{aligned}
$$

You can't write down anything close to the number X even if you were to write a hundred pages of towers of exponentials. Of course, we can continue this game, and define a sequence

$$
\begin{aligned}
g_0 &= \lambda a.\lambda b.a\ b \\
g_1 &= \lambda k.k\ (\lambda f.\lambda n.nf\,n)\ g_0 \\
g_2 &= \lambda k.k\ (\lambda f.\lambda n.nf\,n)\ g_1 \\
&\ldots
\end{aligned}
$$

Which can be generalised as:

$$
\begin{aligned}
f(g) &= \lambda k.k\ (\lambda f.\lambda n.nf\,n)\ g \\
g_n &= f^n(g_0)
\end{aligned}
$$

So we get an even more compact, yet much larger number with:

$$
\begin{aligned}
f &= \lambda g.\lambda k.k\ (\lambda f.\lambda n.nf\,n)\ g \\
Y &= (\lambda n.nf\ \ (\lambda a.aa)\ n)(\lambda f.\lambda z.f(f(f\ z)))
\end{aligned}
$$

Of course, you can easily define much faster growing functions. But here's a challenge: what's the shortest lambda term that normalises, but takes more than the age of the universe to normalise? Or: what's the largest Church natural you can write down in less than 30 symbols?

Please let me know any mistakes. I haven't checked for mistakes at all :)