Better Numbers

Infinite Radix Tree Encoding of Integers

August 1, 2021

The Problem

Current systems for encoding integers are hopelessly ancient. Modern times call for modern solutions. Hindu-Arabic numerals date back to at least the 5th Century AD. Pentadactyly dates back to the first tetrapod Circa 340 Million years BC. You need to be really flexible to use hexadecimal.

The Problem

Worse still even breaking away from base ten, you still need to denote the base you want to use.
Whatever you do, you will just be using base 10.

The solution

Primes are the solution, obviously.
Each number can be encoded as a list of indices of prime factors.
The primes in the system are differentiated only my position in the list.
Through recursion, this system lets us do away with numerals altogether.

The solution

Each tree is represented by a list of children enclosed in angle brackets. The base case is the most irregular, 0 is represented by a tree with only a root $<>$.
I don't care for it but it is really important as it turns out. Only non zero children are listed "by convention" and this saves an infinite amount of space.

Problems Solved

The string "Base 10" is hopelessly ambiguous. In base 10, 10 is written 10.

Examples

0 : <>
1 : < <> >
2 : < < <> > >
3 : < <>,< <> > >
4 : < < < <> > > >
5 : < <>,<>,< <> > >

< < < <> > > > + < <>,<>,< <> > >
= < <>,< < <> > > > < < <> > > * < <>,< <> > > =
< < <> >, < <> > >

Drawbacks

$$<<<<>>>> + <<>, <>, <<>>>>$$

$$=<<>, <<<>>>>$$

$$2^{2^{2^0}} + 2^0 * 3^0 * 5^{2^0} = 2^0 * 3^{2^{2^0}}$$

$$4 + 5 = 9$$

$$<<<>>> * <<>, <<>>>>$$

$$=<<<>>>, <<>>>>$$

It's actually not very easy to use. Adding numbers is easiest by converting out of this format and factorising the sum.

Drawbacks

Multiplication is easy.

Drawbacks

Multiplication is easy. You just add the matching pairs of children.

Drawbacks

Multiplication is easy. You just add the matching pairs of children.
Was it addition that I just said was hard?

Getting Defensive

Okay some things are really easy in this format, describing arbitrary primes? Permuting the indices of the prime factors of a number? The binary operations I've devised so far are really hard to justify but I'm sure they'd get added to some itteration of x86.