Towards Subtraction

Pairs

Pairs of Numbers

How do we encode data structures using LC expression?

Let's look at pairs: (a, b).

Pair(a, b) = \z. z a b

Think of z as a selector function that can be used to pick a or b.

Let p = \z. z a b.

Then observe:

- p True = a
- p False = b

Iterating over pairs of (n-1, n)

- Start with **Pair(0, 0)**.
- We want to have a generator of the following:
  (0, 0) →(0, 1) →(1, 2)→(2, 3)→... →(n-1, n)

```
h = \p.
   IfZero p[1] Then
    Pair(0, 1)
   Else
    Pair(Succ(p[0]), Succ(p[1]))
```

h = \p. **IfZero** (p **False**) (\z. z 0 1) (\z. z (**Succ** p **True**) (**Succ** p **False**))

Extracting predecessor from (n-1, n)

So, we can compute (n-1, n) with the help of h:

\n. n **h** (\z. z 0 0)

Now, we can extract the first component which is (n-1):

Pred = \n. (n **h** (\z. z 0 0) **True**)