

4-1/

LC is simple
we can add pairs, bools, nats by hand

```

fac := (λ n.
  (((if (is zero n)
one
    (mult n) (fac n))))))
  
```

```

mult = λx. λy. λf. λz.
  (y (x f)) z
  
```

LC infinite loop

```

P → → → → P
P → → M → → → M
(λx. M) N ⇒ M[x ← N]
(λx. (L M)) N ⇒ (L M)[x ← N]
(λx. (L x)) N ⇒ (L x) xxx [x ← N]
  
```

```

m → m
(x y) → (x y)
(λx. (M) N) →
  (λ. m) N
  = m[x ← N]
  
```

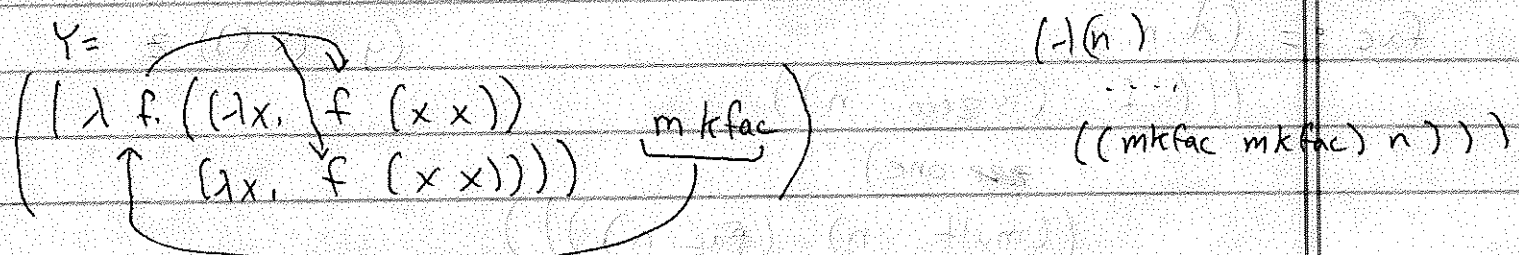
```

Ω = (w w) w = λx. x x
(λx. (x x))
  (λx. (x x))
  → (λx. (x x))
    (λx. (x x))
  = (w w) = Ω
  
```

Ω → Ω

λ -2 / $fac := (\lambda (n) \dots fac \dots)$ $mkfac := (\lambda (fac) (\lambda (n) \dots fac \dots))$

$F = mkfac F$ $(\lambda (mkfac)$



$(\lambda x. mkfac (x x)) (\lambda x. mkfac (x x)) = mkfac (\lambda x. mkfac (x x))$

$(Y mkfac) = mkfac (Y mkfac)$

Y is the fixed-point combinator

ISWIM $E = X \mid \lambda X. E \mid E E \mid E (E_1 \dots E_n) \mid b \mid o_i \mid E_1 \dots E_n \mid O_n \overline{E}^n$

$b =$ the constants $\in B = \{true, false, 0, 1, 2, \dots, 77, 108, \dots\}$

$o_i =$ primitive operations w/ i arguments

$o_1 = \{-\}$ $o_2 = \{+, -, *, /\}$

b and o_i do not contain variable references

$b [X \leftarrow N] = b$ $o_i E_0 E_1 [X \leftarrow N] = o_i E_0 [X \leftarrow N] E_1 [X \leftarrow N]$

operations do not bind variable

Δ -rule: $o_i b_0 \dots b_i \rightarrow a$ δ describes the real behavior of operation
 where $\delta(o_i, b_0 \dots b_i) = a$

$(\lambda x. x + y) 7 \delta \Rightarrow (7 + y) \delta \Rightarrow 7 + 8 \Rightarrow 15$