Programming language theory, 2020 spring.

Start date: 24.02.2019

Deadline: 29.02.2019, 9:00 (means before lesson)

Home work 3:

Exercise 1. using Church Booleans:

 $F=\lambda x.\lambda y.y, T=\lambda x.\lambda y.x$

 $XOR = \lambda a.\lambda b.a(bFT)(bTF)$

 $NOT = \lambda a.aFT$

 $AND = \lambda a.\lambda b.abF$

Exercise 2. using Churh Numerals

 $pred = \lambda n.\lambda y.\lambda x.n(\lambda w.\lambda z.z(w y))(\lambda g.x)(\lambda g.g)$

 $minus = \lambda p.\lambda q.(q pred)p$

 $isequal = \lambda p.\lambda q.AND(GreaterOREqual(p,q)|GreaterOREqual(q,p))$

due to GreaterOREqual = $\lambda p.\lambda q.isZero(p pred q)$, $iszero = \lambda p.p(\lambda a.F)T$

Exercise 4. Implement factorial using

1) Y-combinator - $\lambda y.(\lambda x.y(x x))$

multiplication = $\lambda p.\lambda q.\lambda x(p(q x))$

factorial = Y-combinator ($\lambda p.\lambda q.p(multiplication q(pred q))F$)

2) Z-combinator - $\lambda p(\lambda q.p(\lambda r.(q q)r))(\lambda q.p(\lambda r.(q q)r))$

factorial = Z-combinator($\lambda p.\lambda q.p(\lambda x.(multiplication q pred q)F))$

I can't understand and check it by my own, so I tried to find more information about it in other sources.

$$Y(f) = (g \Rightarrow (x \Rightarrow g(x(x)))(x \Rightarrow g(x(x)))(f)$$

$$= (x \Rightarrow f(x(x)))(x \Rightarrow f(x(x)))$$

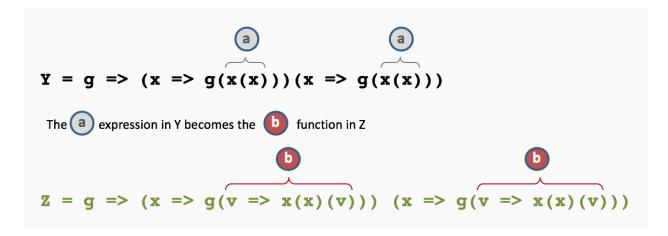
$$= f((x \Rightarrow f(x(x)))(x \Rightarrow f(x(x)))$$

$$= f(x \Rightarrow f(x(x)))(x \Rightarrow f(x(x)))$$

```
 Z(f) = (g \Rightarrow (x \Rightarrow g(v \Rightarrow x(x)(v))) (x \Rightarrow g(v \Rightarrow x(x)(v))))(f) 
 = (x \Rightarrow f(v \Rightarrow x(x)(v))) (x \Rightarrow f(v \Rightarrow x(x)(v))) 
 = f(v \Rightarrow (x \Rightarrow f(v \Rightarrow x(x)(v))) (x \Rightarrow f(v \Rightarrow x(x)(v)))(v)) 
 = f(v \Rightarrow (x \Rightarrow f(v \Rightarrow x(x)(v))) (x \Rightarrow f(v \Rightarrow x(x)(v)))(v)) 
 = f(v \Rightarrow z(f)(v)) 
 = f(v \Rightarrow z(f)(v))
```

 $\frac{https://medium.com/swlh/y-and-z-combinators-in-javascript-lambda-calculus-with-real-code-31f25be934ec}{}$

The difference is that it introduces a function in its definition instead of a direct calculation of an expression.



Comparison between Y and Z