Lambda Calculus

May 7, 2024

Ji, Yong-hyeon

λ`input.output`

```
function (input) {
        return output
}
```

$\lambda a.\lambda b.$

```
def add (a, b):
        return a + b

add(5,7)

def add (a):
        def adda(b):
                return a + b
        return adda

add(5)(7)
```

$\lambda a.(\lambda b.(a+b))(5)(7)$

$\mathsf{T} = \lambda a.\lambda b.a$

$\mathsf{F} = \lambda a.\lambda b.b$

λ`bool.bool`$(t)(f)$

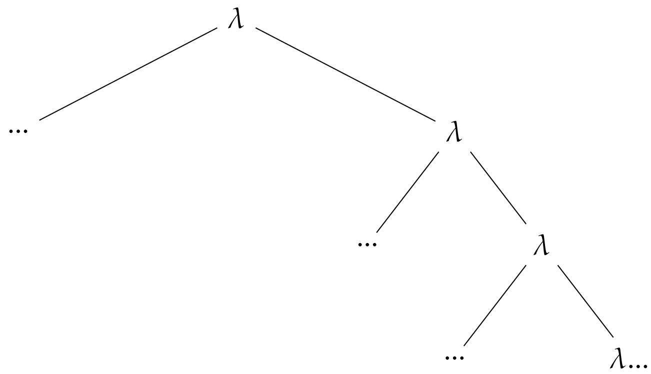
- returned if bool is true

- returned if bool is false

Logic Gates

- $\neg = \lambda b.b(\mathsf{F})(\mathsf{T})$
  - returned if b is true
  - returned if b is false
- $\vee = \lambda a.\lambda b.a(\mathsf{T})(b)$
  - returned if a is true
  - returned if b is false

- $\wedge = \lambda a.\lambda b.a(b)(\mathsf{F})$
  - returned if a is true
  - returned if b is false



$$
\begin{aligned}
1 &= \lambda f.\lambda a.f(a) \\
2 &= \lambda f.\lambda a.f(f(a)) \\
3 &= \lambda f.\lambda a.f(f(f(a))) \\
4 &= \lambda f.\lambda a.f(f(f(f(a))))
\end{aligned}
$$

$$3(f)(a) = f(f(f(a)))$$

$$
\begin{aligned}
+1 &= \lambda n.\lambda f.\lambda a.f(n(f)(a)) \\
+ &= \lambda x.\lambda y.x(+1)(y) \\
* &= \lambda x.\lambda y.y(+(x))(0) \\
** &= \lambda x.\lambda y.y(*(x))(1)
\end{aligned}
$$

Department of Information Security, Cryptology and Mathematics
College of Science and Technology
Kookmin University
Seoul, Republic of Korea

References

[1] "Why functions are turing complete (Lambda Calculus)" YouTube, uploaded by A Byte of Code, 4 Sep 2022, https://www.youtube.com/watch?v=m32kbFXBRR0