1 Lecture

1.1 Syntax

$$\begin{aligned} Expr ::= {} & X \mid \\ & \lambda X.Expr \mid \\ & Expr\ Expr, \end{aligned} \tag{1}$$

where X is a predefined set of variable names.

1.2 Operational Semantics

$$(\lambda X.E)F \to E[X \mapsto F] \tag{beta-reduction}$$

Two expressions are beta-convertible ($=_\beta$) $\equiv$ they can be reduced to the same expression.

1.3 Auxiliaries

$$FV(X) = \{X\} \tag{2}$$
$$FV(\lambda X.E) = FV(E) \setminus \{X\} \tag{3}$$
$$FV(EF) = FV(E) \cup FV(F) \tag{4}$$

$$X[X \mapsto E] = E \tag{5}$$
$$Y[X \mapsto E] = Y \text{ if } Y \neq X \tag{6}$$
$$(\lambda X.E)[Y \mapsto F] = \lambda X.(E[Y \mapsto F]) \text{ if } Y \neq X \text{ and } X \notin FV(F) \tag{7}$$
$$(E\ F)[Y \mapsto G] = (E[Y \mapsto G])(F[Y \mapsto G]) \tag{8}$$

$$\lambda X.E = \lambda Y.(E[X \mapsto Y]) \text{ if } Y \notin FV(E) \tag{alpha-renaming}$$

1.4 Notable Combinators

An expression without any free variables is called combinator.

$$\Omega = (\lambda X.XX)(\lambda X.XX) \tag{divergent combinator}$$

$$Y = \lambda F.(\lambda X.F(XX))(\lambda X.F(XX)) \tag{Y combinator, fixpoint combinator}$$

For any F, $YF =_\beta F(YF)$.

$$I = \lambda X.X \tag{identity combinator}$$

$$K = \lambda X.\lambda Y.X \tag{K combinator}$$

$$S = \lambda X.\lambda Y.\lambda Z.(XZ)(YZ) \tag{S combinator}$$

2 Seminar

1. Implement booleans and their operations.
2. Implement natural numbers and their operations.
3. Use fixpoint combinator to implement function $\lambda n. \sum_{i=1}^{n} i$.