1 Denotational Semantics

1.1 Denotational Semantics of Lambda Calculus

1.1.1 Syntax

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

1.1.2 Semantics

Semantic domains: $env = string \to function$, $fcn = fcn \to fcn$; notational conventions $e \in env, f, f' \in fcn, E, E' \in Expr$

$$[\![x]\!] = \lambda e.e(x) \tag{2}$$

$$[\![\lambda x.E]\!] = \lambda e.\lambda p.[\![E]\!](e[x \mapsto p]) \tag{3}$$

$$[\![E_1 \ E_2]\!] = \lambda e.([\![E_1]\!](e))([\![E_2]\!](e)) \tag{4}$$

2 Seminar

1. What is the denotation of $\lambda x.x$?
2. Define a denotational semantics of the language of expressions with variables.
3. Define the denotational semantics of an imperative language with the following syntax:

$$\begin{aligned} Expr ::= & \ Num \mid \\ & \triangle Expr \mid \\ & Expr \square Expr \mid \\ & \star VarName \\ Statement ::= & \ VarName \texttt{ = } Expr \\ Program ::= & \ Statement\texttt{; } Program \mid \\ & \ \epsilon \end{aligned} \tag{5}$$

4. Implement the denotational semantics from previous task in *Mathematica*.