

hw7

Zhitm

May 2023

1 Parser

program ::= variable | let-expr | variable program | let-expr program
let-expr ::= "let" variable "=" term
term ::= atom | application | abstraction
params ::= variable | variable params
variable ::= letter | variable letter | variable digit
letter ::= "a" | "b" | ... "Z"
digit ::= "1" | ... "9" | "0"
application ::= atom | application atom
atom ::= parenthesized-term | variable
parenthesized-term ::= "(" term ")"
abstraction ::= "λ" params "." term

Exaples:

let K = λx y.x
program → let-expr
— let-expr → "let" variable "=" term → variable
—— variable → "K"
— let-expr → "let" variable "=" term → term
—— term → abstraction
——— abstraction → "λ" params "." term → params
———— params → variable params → variable
————— variable → x
————— params → variable params → params → variable
————— variable → y
——— abstraction → "λ" params "." term → term
——— term → atom → variable
———— variable → "x"