

hw7

Zhitm

May 2023

1 Parser

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

Exaples:

let K = λx y.x

K

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

“let” “K” “=” “λ” “x” “y” “.” letter term →
 “let” “K” “=” “λ” “x” “y” “.” “x” term →
 “let” “K” “=” “λ” “x” “y” “.” “x” atom →
 “let” “K” “=” “λ” “x” “y” “.” “x” variable →
 “let” “K” “=” “λ” “x” “y” “.” “x” letter →
 “let” “K” “=” “λ” “x” “y” “.” “x” “K”