Факторизованная грамматика для интерпретатора лямбда-выражений

1 Грамматика

Терминальные символы Σ = {\, . , (,) , [a-zA-Z], [0-9], _ , let , = , \n , ϵ }
ϵ – пустая строка

Нетерминальные символы N = {program, definition, term, term', lambda-abstraction, variable, letter, digit }

Стартовый символ S = {programm}

1.1 Правила

$\langle program \rangle ::= \langle definition \rangle \{ \backslash n \langle program \rangle \} \mid \langle term \rangle \{ \backslash n \langle program \rangle \} \mid \epsilon$
$\langle definition \rangle ::= let\ \langle variable \rangle = \langle term \rangle$
$\langle term \rangle ::= \langle lambda - abstraction \rangle \mid \langle variable \rangle \mid \langle term' \rangle$
$\langle term' \rangle ::= \langle term \rangle\ \langle term \rangle \mid \langle term \rangle \langle term \rangle \mid (\langle term \rangle) \mid \langle term \rangle\ (\langle term \rangle) \mid (\langle term \rangle)\ (\langle term \rangle) \mid$
$(\langle term \rangle)\ \langle term \rangle \mid \langle term \rangle (\langle term \rangle) \mid (\langle term \rangle)(\langle term \rangle) \mid (\langle term \rangle) \langle term \rangle \mid$
$\langle lambda - abstraction \rangle ::= \backslash \langle variable \rangle \{\ \langle variable \rangle \}. \langle term \rangle$
$\langle variable \rangle ::= \langle letter \rangle \{ \langle letter \rangle \langle digit \rangle \}$
$\langle letter \rangle ::= a|...|z|A|...|Z$
$\langle digit \rangle ::= 0|1|2|3|4|5|6|7|8|9$

2 Примеры

let S = \x y z.x z (y z)
let K = \x y.x
Вывод: Для K
program $\rightarrow$ $definition \rightarrow let\ \langle variable \rangle = \langle term \rangle \rightarrow let\ K = \langle term \rangle \rightarrow$
$let\ K = \langle lambda - abstraction \rangle \rightarrow let\ K = \backslash \langle variable \rangle \{\ \langle variable \rangle \}. \langle term \rangle \rightarrow$
$let\ K = \backslash x\ y. \langle term \rangle \rightarrow let\ K = \backslash x\ y. \langle variable \rangle \rightarrow let\ K = \backslash x\ y.x$
S
$program \rightarrow definition \rightarrow let\ \langle variable \rangle = \langle term \rangle \rightarrow let\ \langle S \rangle = \langle term \rangle \rightarrow$
$let\ \langle S \rangle = \langle lambda - abstraction \rangle \rightarrow let\ S = \backslash \langle variable \rangle \{\ \langle variable \rangle \}. \langle term \rangle \rightarrow$

$let\ S = \backslash xyz.\langle term\rangle \rightarrow let\ S = \backslash xyz.\langle term'\rangle \rightarrow let\ S = \backslash xyz.\langle term\rangle\ \langle term\rangle \rightarrow$
$let\ S = \backslash xyz.\langle variable\rangle\ \langle term\rangle \rightarrow let\ S = \backslash xyz.x\ \langle term\rangle \rightarrow let\ S = \backslash xyz.x\ \langle term'\rangle \rightarrow$
$let\ S = \backslash xyz.x\ \langle term\rangle\ (\langle term\rangle) \rightarrow let\ S = \backslash xyz.x\ \langle variable\rangle\ (\langle term\rangle) \rightarrow let\ S =$
$\backslash xyz.x\ y\ (\langle term\rangle) \rightarrow let\ S = \backslash xyz.x\ y\ (\langle term'\rangle) \rightarrow let\ S = \backslash xyz.x\ y\ (\langle term\rangle\ \langle term\rangle) \rightarrow$
$let\ S = \backslash xyz.x\ y\ (\langle variable\rangle\ \langle term\rangle) \rightarrow let\ S = \backslash xyz.x\ y\ (y\ \langle term\rangle) \rightarrow let\ S =$
$\backslash xyz.x\ y\ (y\ \langle variable\rangle) \rightarrow let\ S = \backslash xyz.x\ y\ (y\ z)$