

# Lambda Grammar

Ksenchik Nikita

May 28, 2023

## 1 Grammar 1

$\langle \text{program} \rangle ::= \text{let } \langle \text{variable} \rangle = \langle \text{term} \rangle \mid \langle \text{term} \rangle$   
 $\langle \text{term} \rangle ::= \langle \text{appliqueSeq} \rangle \mid \langle \text{term}' \rangle$   
 $\langle \text{term}' \rangle ::= \langle \text{variable} \rangle \mid \langle \text{abstraction} \rangle \mid "(" \langle \text{term} \rangle ")"$   
 $\langle \text{appliqueSeq} \rangle ::= \langle \text{term}' \rangle \langle \text{term}' \rangle \{ \langle \text{term}' \rangle \}$   
 $\langle \text{abstraction} \rangle ::= \backslash \langle \text{variable} \rangle \{ \langle \text{variable} \rangle \}. \langle \text{term} \rangle$   
 $\langle \text{variable} \rangle ::= \langle \text{keyword} - \text{exception} \rangle \mid \langle \text{letter} \rangle \{ \langle \text{letter} \rangle \mid \langle \text{digit} \rangle \}$   
 $\langle \text{keyword} - \text{exception} \rangle ::= \text{let}$   
 $\langle \text{letter} \rangle ::= a \mid \dots \mid z \mid A \mid \dots \mid Z$   
 $\langle \text{digit} \rangle ::= 1 \mid \dots \mid 9$

## 2 Grammar 2

$\langle \text{program} \rangle ::= \text{let } \langle \text{variable} \rangle = \langle \text{term} \rangle \mid \langle \text{term} \rangle$   
 $\langle \text{term} \rangle ::= \langle \text{content} \rangle \langle \text{term}' \rangle$   
 $\langle \text{term}' \rangle ::= \langle \text{content} \rangle \langle \text{term}' \rangle \mid \langle \text{epsilon} \rangle$   
 $\langle \text{content} \rangle ::= \langle \text{variable} \rangle \mid \langle \text{abstraction} \rangle \mid "(" \langle \text{term} \rangle ")"$   
 $\langle \text{abstraction} \rangle ::= \backslash \langle \text{variable} \rangle \{ \langle \text{variable} \rangle \}. \langle \text{term} \rangle$   
 $\langle \text{variable} \rangle ::= \langle \text{keyword} - \text{exception} \rangle \mid \langle \text{letter} \rangle \{ \langle \text{letter} \rangle \mid \langle \text{digit} \rangle \}$   
 $\langle \text{keyword} - \text{exception} \rangle ::= \text{let}$   
 $\langle \text{letter} \rangle ::= a \mid \dots \mid z \mid A \mid \dots \mid Z$   
 $\langle \text{digit} \rangle ::= 1 \mid \dots \mid 9$   
 $\langle \text{epsilon} \rangle ::= ""$

## 3 Examplpe 1, Gr. 1

$\text{let } K = \backslash x y.x \implies$   
 $\langle \text{program} \rangle$   
 $\text{let } \langle \text{variable} \rangle = \langle \text{term} \rangle$   
 $\text{let } \langle \text{letter} \rangle \{ \langle \text{letter} \rangle \mid \langle \text{digit} \rangle \} = \langle \text{term} \rangle$   
 $\text{let } K \{ \langle \text{letter} \rangle \mid \langle \text{digit} \rangle \} = \langle \text{term} \rangle$   
 $\text{let } K = \langle \text{term} \rangle$   
 $\text{let } K = \langle \text{term}' \rangle$   
 $\text{let } K = \langle \text{abstraction} \rangle$   
 $\text{let } K = \backslash \langle \text{variable} \rangle \{ \langle \text{variable} \rangle \}. \langle \text{term} \rangle$   
 $\text{let } K = \backslash \langle \text{letter} \rangle \{ \langle \text{letter} \rangle \mid \langle \text{digit} \rangle \} \{ \langle \text{variable} \rangle \}. \langle \text{term} \rangle$   
 $\text{let } K = \backslash \langle \text{letter} \rangle \{ \langle \text{variable} \rangle \}. \langle \text{term} \rangle$   
 $\text{let } K = \backslash x \{ \langle \text{variable} \rangle \}. \langle \text{term} \rangle$   
 $\text{let } K = \backslash x \langle \text{variable} \rangle \{ \langle \text{variable} \rangle \}. \langle \text{term} \rangle$   
 $\text{let } K = \backslash x \langle \text{letter} \rangle \{ \langle \text{letter} \rangle \mid \langle \text{digit} \rangle \} \{ \langle \text{variable} \rangle \}. \langle \text{term} \rangle$   
 $\text{let } K = \backslash x y \{ \langle \text{letter} \rangle \mid \langle \text{digit} \rangle \} \{ \langle \text{variable} \rangle \}. \langle \text{term} \rangle$   
 $\text{let } K = \backslash x y \{ \langle \text{variable} \rangle \}. \langle \text{term} \rangle$   
 $\text{let } K = \backslash x y. \langle \text{term} \rangle$

$$\begin{aligned} \text{let } K &= \backslash x y. \langle \text{term}' \rangle \\ \text{let } K &= \backslash x y. \langle \text{variable} \rangle \\ \text{let } K &= \backslash x y. \langle \text{letter} \rangle \{ \langle \text{letter} \rangle \mid \langle \text{digit} \rangle \} \\ \text{let } K &= \backslash x y.x \{ \langle \text{letter} \rangle \mid \langle \text{digit} \rangle \} \\ \text{let } K &= \backslash x y.x \end{aligned}$$

#### 4 Example 2, Gr 1

```

let gg = (\l o.l) e z ==>
< program >
let < letter > {< letter > | < digit >} = < term >
let g {< letter > | < digit >} = < term >
let g < letter > {< letter > | < digit >} = < term >
let gg {< letter > | < digit >} = < term >
let gg = < term >
let gg = < applique >
let gg = < term' > < term' > {< term' >}
let gg = (< term >) < term' > {< term' >}
let gg = (< term' >) < term' > {< term' >}
let gg = (< abstraction >) < term' > {< term' >}
let gg = (\ < variable > {< variable >}. < term >) < term' > {< term' >}
let gg = (\ < letter > {< letter > | < digit >} {< variable >}. < term >) < term' > {<
term' >}
let gg = (\l {< letter > | < digit >} {< variable >}. < term >) < term' > {< term' >}
let gg = (\l {< variable >}. < term >) < term' > {< term' >}
let gg = (\l < variable > {< variable >}. < term >) < term' > {< term' >}
let gg = (\l < letter > {< letter > | < variable >} {< variable >}. < term >) < term' > {<
term' >}
let gg = (\l o {< letter > | < variable >} {< variable >}. < term >) < term' > {< term' >}
let gg = (\l o {< variable >}. < term >) < term' > {< term' >}
let gg = (\l o. < term >) < term' > {< term' >}
let gg = (\l o. < term' >) < term' > {< term' >}
let gg = (\l o. < variable >) < term' > {< term' >}
let gg = (\l o. < letter > {< letter > | < digit >}) < term' > {< term' >}
let gg = (\l o.l {< letter > | < digit >}) < term' > {< term' >}
let gg = (\l o.l) < term' > {< term' >}
let gg = (\l o.l) < variable > {< term' >}
let gg = (\l o.l) < letter > {< letter > | < digit >} {< term' >}
let gg = (\l o.l) e {< letter > | < digit >} {< term' >}
let gg = (\l o.l) e {< term' >}
let gg = (\l o.l) e < term' > {< term' >}
let gg = (\l o.l) e < variable > {< term' >}
let gg = (\l o.l) e < letter > {< letter | < digit >} {< term' >}
let gg = (\l o.l) e z {< letter | < digit >} {< term' >}
let gg = (\l o.l) e z {< term' >}
let gg = (\l o.l) e z

```