Q: Grammar which generates strings of length 2.

$\Sigma = \{a, b\}$

Way 1: $L = \{aa, ab, ba, bb\}$ → finite

$S \to aa \mid ab \mid ba \mid bb$

Way 2: RE for language

either a or b

$\underbrace{(a+b)}_{A}\underbrace{(a+b)}_{A}$

$$\boxed{\begin{aligned} S &\to AA \\ A &\to a \mid b \end{aligned}}$$

Q: $a^n \mid n \geq 0$

$L = \{a^0, a^1, a^2, a^3, \dots\}$

$L = \{\varepsilon, a, aa, aaa, \dots\}$

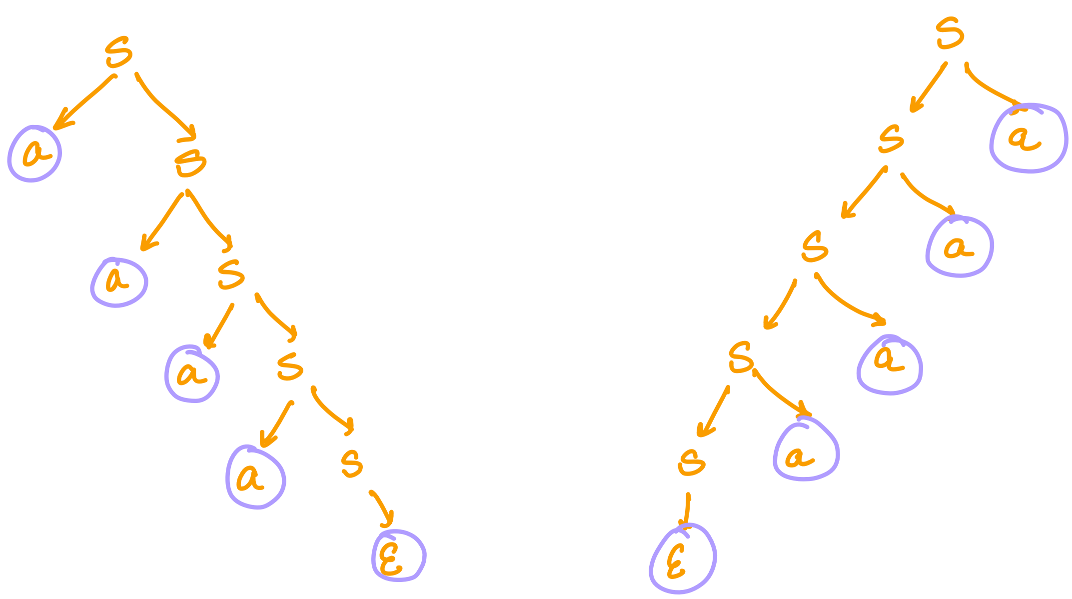
RE: a^*

$\boxed{S \to aS \mid \varepsilon}$ OR $\boxed{S \to Sa \mid \varepsilon}$

Parse Tree

aaaa :

aaaa



Code:

```
int main()
{
    int a,b;
    a=10;
    b=20;
    ⋮
}
```

$(a+b)(a+b)$

Expression

↑ Represent

Regular language

Lang. Strings of length 2

Generator

Regular Grammar

$S \to AA$

$A \to a \mid b$

fit?

$S \to \underline{DT}\ \underline{A}\ ;$

$DT \to int \mid float \mid char$

$A \to \underline{VN}, A$

Q: $(a+b)^*$ ⟶ set of all strings over a, b

length 1: $(a+b)$

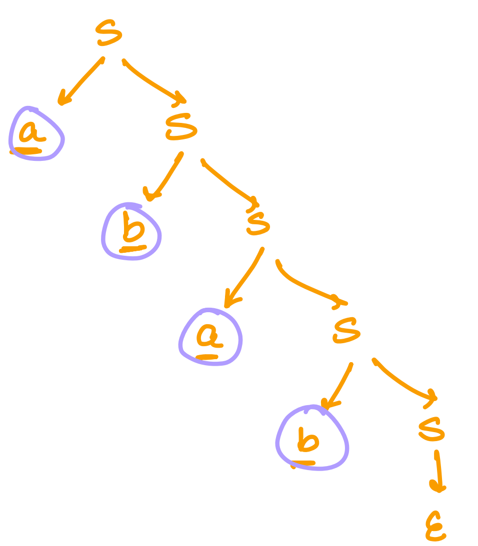
length 2: $(a+b)^2 = (a+b)(a+b)$

⋮

** $a^*: S \to aS \mid \varepsilon$

** $\boxed{S \to aS \mid bS \mid \varepsilon}$

Derive abab



Q: Strings of length atleast 2 language

RE: $\underbrace{(a+b)}_{A}\underbrace{(a+b)}_{A}\underbrace{(a+b)^*}_{B}$

$S \rightarrow AAB$

$A \rightarrow a \mid b \quad \longrightarrow (a+b)$

$B \rightarrow aB \mid bB \mid \varepsilon \quad \longrightarrow (a+b)^*$

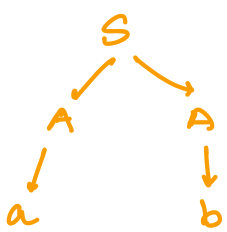
Q: length atmost 2

RE: $\underbrace{(a+b+\varepsilon)}_{A}\underbrace{(a+b+\varepsilon)}_{A}$

$S \rightarrow AA$

$A \rightarrow a \mid b \mid \varepsilon$

} can't give strings of length ≥ 3



Q: Start and ends with different symbol

$\Sigma = \{a, b\}$

RE: $a\underbrace{(a+b)^*}_{A}b + b\underbrace{(a+b)^*}_{A}a$

$S \rightarrow aAb \mid bAa$

$A \rightarrow aA \mid bA \mid \varepsilon$

Q: Starts and Ends with same symbol

① RE: $a\underbrace{(a+b)^*}_{A}a + b\underbrace{(a+b)^*}_{A}b + a + b$

② $S \rightarrow aAa \mid bAb \mid a \mid b$

$A \rightarrow aA \mid bA \mid \epsilon$

Q: $a^n b^n \mid n \geq 1$ } language

$\rightarrow$ ab, aabb, ...

5 a's 5 b's

6 a's 6 b's

RE:

$\underline{a,b}$ alphabets, +, ·, *

repeat

$((a+b)^* \cdot (a+b)^*)^*$

Regular language: FA or RE } Regular language

$a^n b^n \longrightarrow$ You can't write a RE

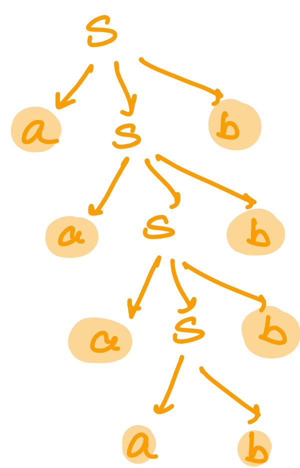
aaaa bbbb

① count

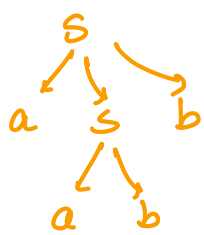
② first a's then b's

$S \longrightarrow aSb \mid ab$

a^4b^4



a^2b^2

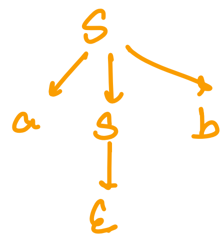


Q: $a^n b^n \mid n \geq 0$

ε, ab, aabb . . .

$S \to aSb \mid \varepsilon$

ab :



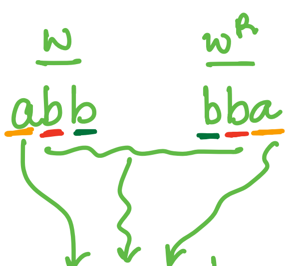
Q: Palindrome

even length

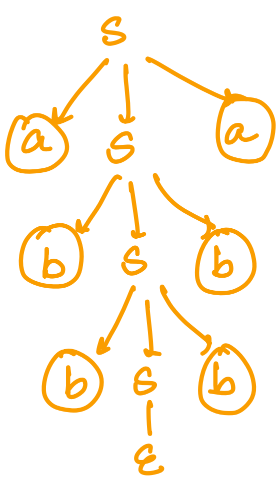
ww^R

3 3 = 6

aab baa



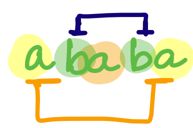
$S \to aSa \mid bSb \mid \varepsilon$

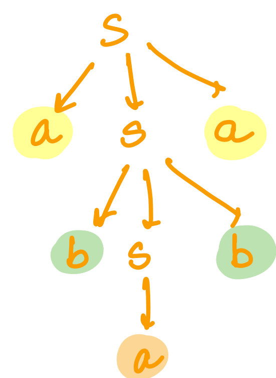


odd length

waw^R OR wbw^R

$S \to aSa \mid bSb \mid a \mid b$





Palindrome: $S \to aSa \mid bSb \mid a \mid b \mid \varepsilon$

Q: Even length strings

RE: $(\underbrace{\underbrace{(a+b)}_{A}\underbrace{(a+b)}_{A}}_{B})^*$

B^*

$S \to BS \mid \varepsilon$

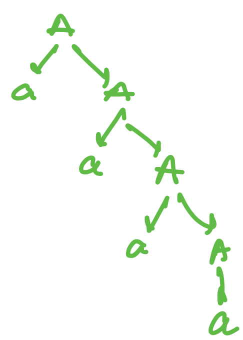
$S \to BS \mid \varepsilon$
$B \to AA$
$A \to a \mid b$

Q: $a^n b^m \mid n, m \geq 1$

RE: $a^+ b^+$

$\underbrace{aa^*}_{A}\ \underbrace{bb^*}_{B}$

$S \to AB$
$A \to aA \mid a$
$B \to bB \mid b$



$a^+ : aa^*$

$S \to aS \mid a$

a^* $S \to aS \mid \varepsilon$

Q: $\underbrace{a^n b^n}_{A}\ \underbrace{c^m}_{B} \mid n, m \geq 1$

$S \to AB$

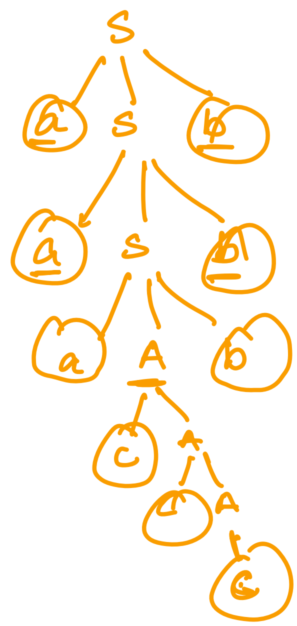
$A \to aAb \mid ab$ $\to a^n b^n$
$B \to cB \mid c$ $\to c^m$

Q: $a^n \underbrace{c^m}_{A} b^n \mid n, m \geq 1$

$S \rightarrow aSb \mid aAb$

$A \rightarrow cA \mid c$

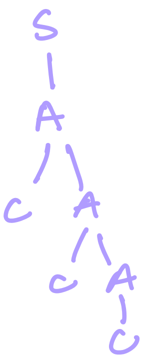
aa bb

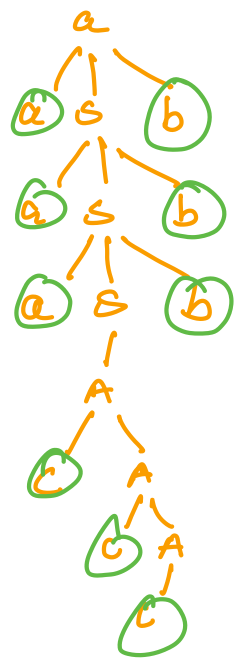


Problem

$S \rightarrow aSb \mid A$

$A \rightarrow cA \mid c$





ccc

not in the language

Q: $\underbrace{a^n b^n}_{A} \underbrace{c^m d^m}_{B} \mid n, m \geq 1$

$S \rightarrow AB$

$A \rightarrow aAb \mid ab$

$B \rightarrow cBd \mid cd$

Q: $a^n b^{2n} \mid n \geq 1$

$a^n (bb)^n$

$S \rightarrow aSbb \mid abb$

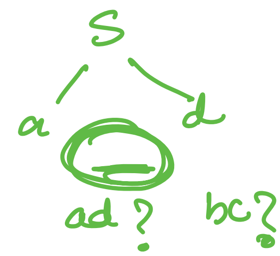
$a^n b^{2n} \mid n \geq 0$

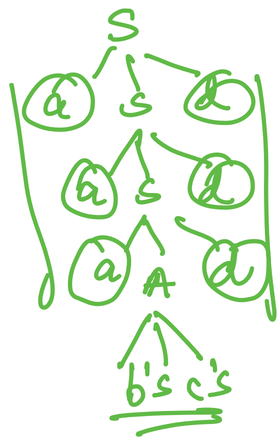
$S \rightarrow aSbb \mid \epsilon$

Q: $a^n \underbrace{b^m c^m}_{A} d^n \mid n,m \geq 1$

$S \longrightarrow aSd \mid aAd$

$A \longrightarrow bAc \mid bc$





Q: $a^{m+n} b^m c^n \mid n,m \geq 1$

$a^n \underbrace{a^m b^m}_{A} c^n$

$S \longrightarrow aSc \mid aAc$

$A \longrightarrow aAb \mid ab$

Q: $a^n b^{n+m} c^m \mid n,m \geq 1$

$a^n \underline{b^m b^n} c^m$

$\underbrace{a^n b^n}_{A} \underbrace{b^m c^m}_{B}$

$S \longrightarrow AB$

$A \longrightarrow aAb \mid ab \longrightarrow a^n b^n$

$B \longrightarrow bBc \mid bc \longrightarrow b^m c^m$

Q: $a^n b^m c^{n+m} \mid n,m \geq 1$

$a^n \underbrace{b^m c^m}_{A} c^n \mid n,m \geq 1$

$S \longrightarrow aSc \mid aAc$

$A \longrightarrow bAc \mid bc$

Classification of Grammar:

Chomsky divided the grammar into 4 types:

1. Type 3 (Regular Grammar)
2. Type 2 (Context free Grammar)
3. Type 1 (Context Sensitive Grammar)
4. Type 0 (Recursively Enumerable Grammar)

more relaxed ↓ | ↑ more restricted

Type 3: Regular Grammar

Grammar has all the productions of the form:

1st definition

$A \rightarrow \alpha B \mid \beta$

$A, B \in V$

$\alpha, \beta \in T^*$

$S \rightarrow \underset{\alpha}{a}\underset{B}{S} \mid \underset{\beta}{a} \rightarrow a^+$

aa^*

$\underset{L}{\alpha} \cdot \underset{R}{B}$

Right linear Grammar

2nd definition

$A \rightarrow B\alpha \mid \beta$

$A, B \in V$

$\alpha, \beta \in T^*$

$S \rightarrow \underset{B}{S}\underset{\alpha}{a} \mid a$

$\underset{L}{B} \cdot \underset{R}{\alpha}$

Left linear Grammar

eg:

$A \rightarrow aB \mid a$

$B \rightarrow aB \mid bB \mid a \mid b$

Right linear Grammar

$V \rightarrow \text{terminals} \; V$ (RLG)

OR

$V \rightarrow V \; \text{terminals}$ (LLG)

eg:

$A \rightarrow Ba \mid a$

$B \rightarrow Ba \mid Bb \mid a \mid b$

Left linear Grammar

Eg:

$A \to Ba \mid a$ ⟶ LLG

$B \to aB \mid a$ — RLG

Not a Type 3 Grammar
(bcz it is a combination of LLG & RLG)

Type 2

Productions are of the form:

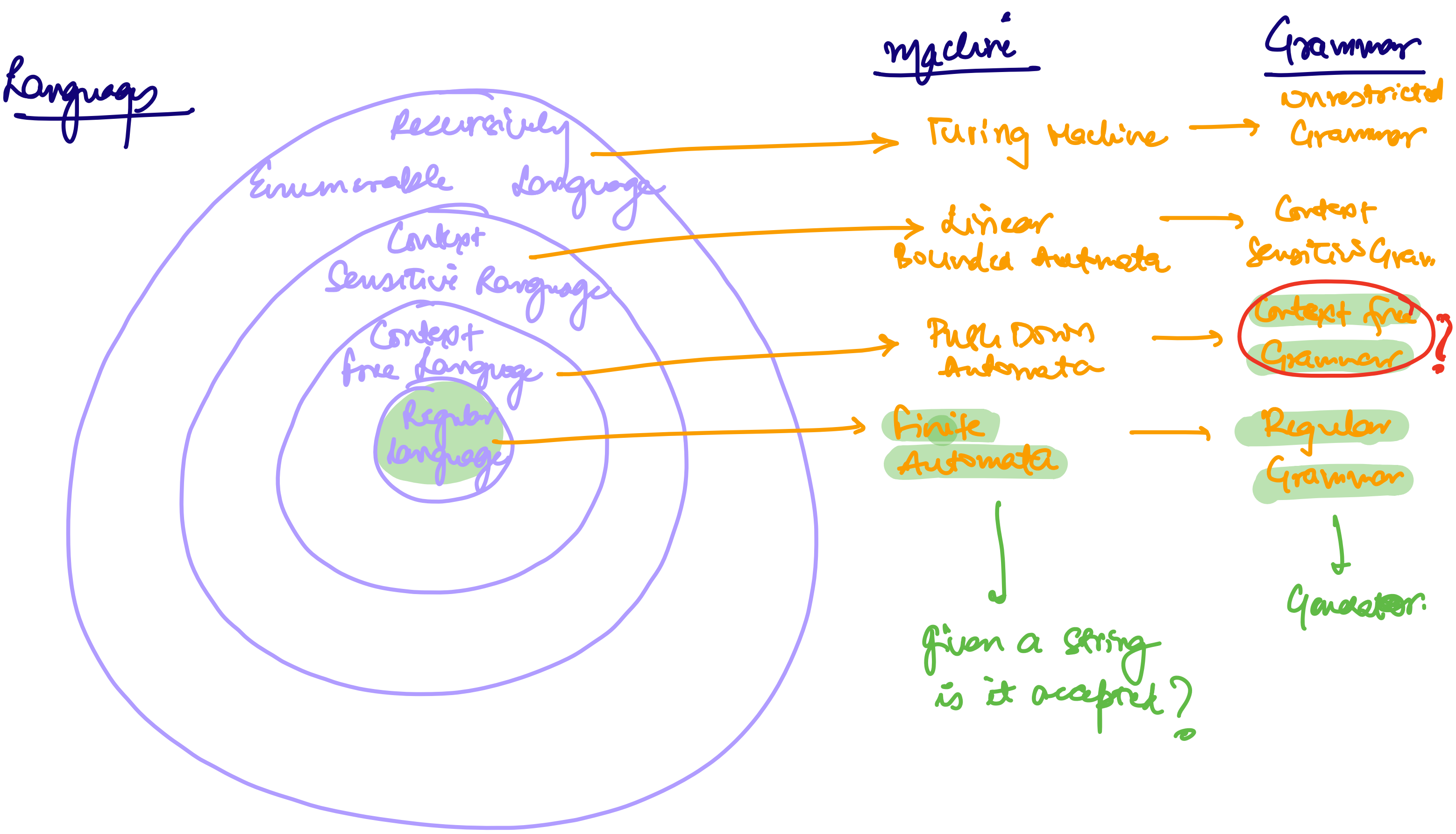
$A \to \alpha$

$A \in V$

$\alpha \in (V \cup T)^*$

↳ Context free grammar

Eg: CFG: $\underbrace{A}_{\text{1 Variable}} \to \underbrace{aAb \mid ab}_{\text{Variables, Terminals}}$



CFG can have ambiguity ?

AMBIGUITY:

$$E \rightarrow E+E \mid E*E \mid id$$

$V = \{E\}$
$T = \{+, *, id\}$
$S = \{E\}$

String: $id + id * id$

Left Most Derivation:

$$E \rightarrow E + E \rightarrow id + E \rightarrow id + E*E \rightarrow id + id*id$$

Right Most Derivation:

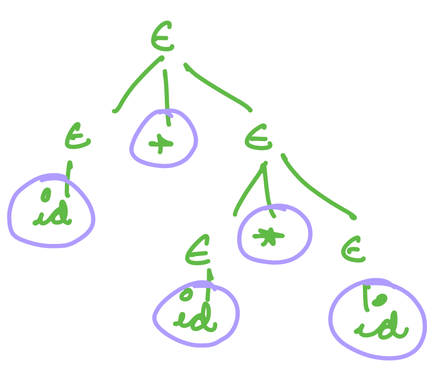
$$E \rightarrow E + E \rightarrow E + E*E \rightarrow E + E*id \rightarrow E + id*id \rightarrow id + id*id$$

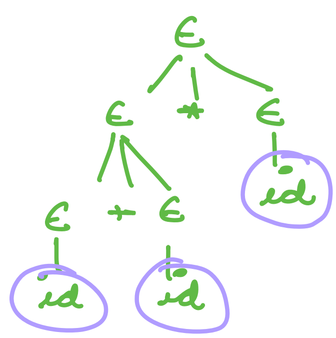
$$E \rightarrow E*E \rightarrow E+E*E \rightarrow id + E*E \rightarrow id + id*id$$

$$E \rightarrow E*E \rightarrow E*id \rightarrow E+E*id \rightarrow E+id*id \rightarrow id+id*id$$

PARSE TREE:

$id + id * id$



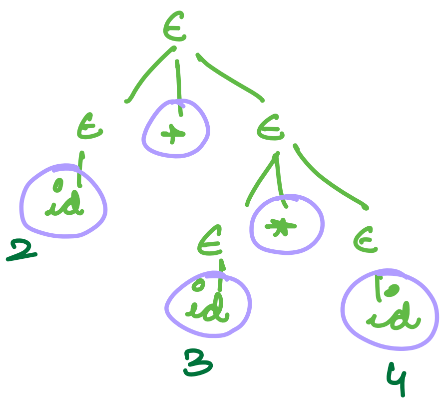


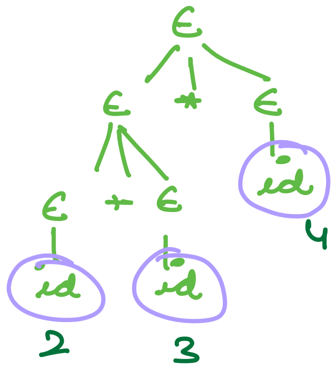
for a given string and a given grammar, you get more than 1 LMD, more than 1 RMD or more than 1 PT

$\Downarrow$

Grammar is ambiguous.

2+3*4





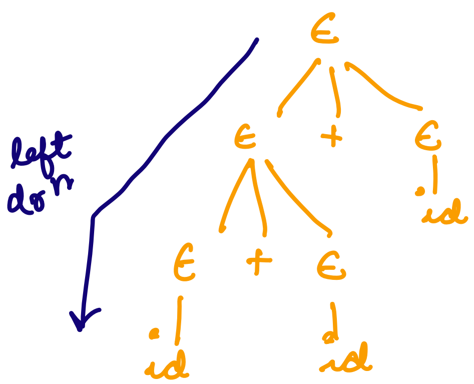
$\Downarrow$ 14 ✓ correct answer

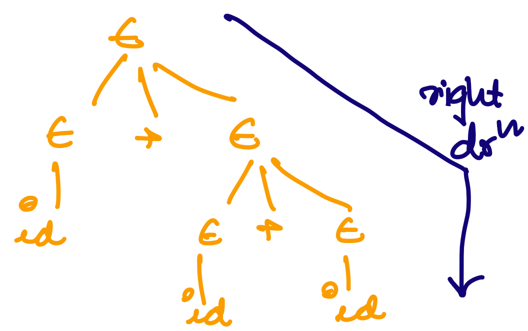
$\Downarrow$ 20

Disambiguous

$E \rightarrow E * E \mid E + E \mid id$

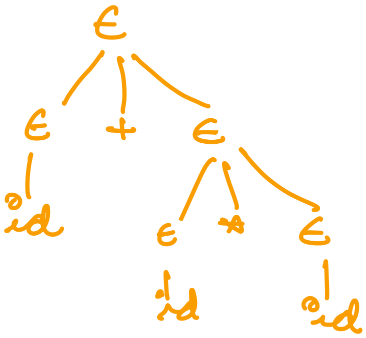
Case 1 String: id + id + id

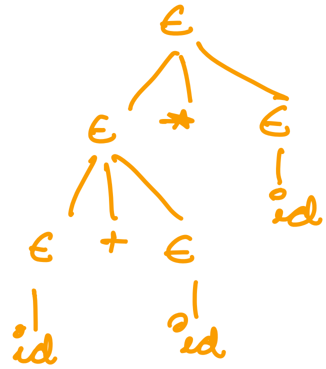




Case 2

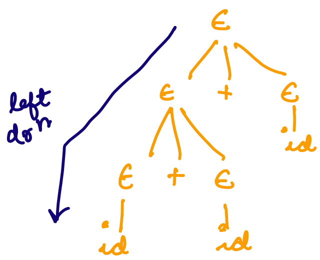
id + id * id

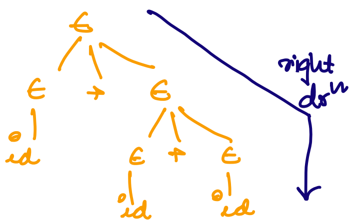


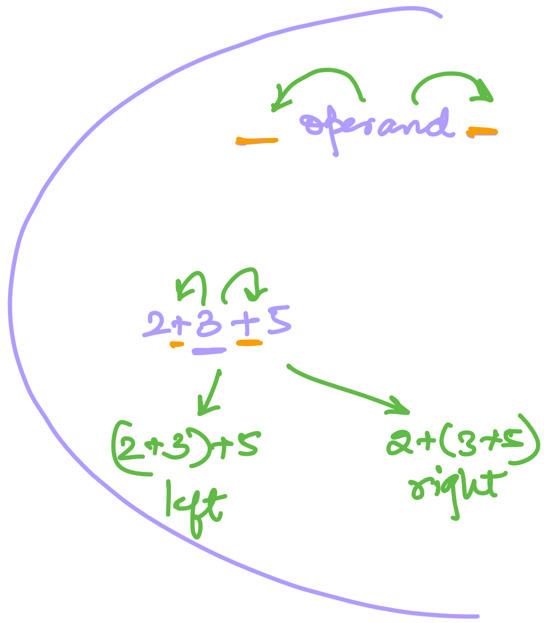


Case 1

Associativity Rules should be defined properly.







Left Associativity

Right Associativity

LHS → RHS

LHS = Leftmost symbol of RHS

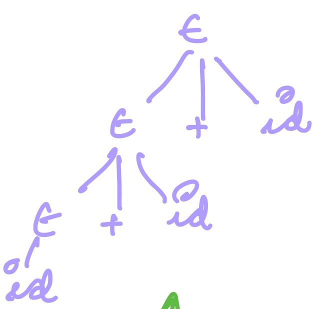
LHS = Rightmost symbol of RHS

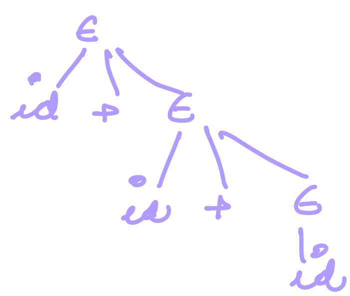
$E \rightarrow E + E \mid id$ Original Rule

$E \rightarrow E + id \mid id$

$E \rightarrow id + E \mid id$ → Right Recursive Grammar

Left Recursive Grammar





2+3+5

(2+3)+5

Left

2^3^5

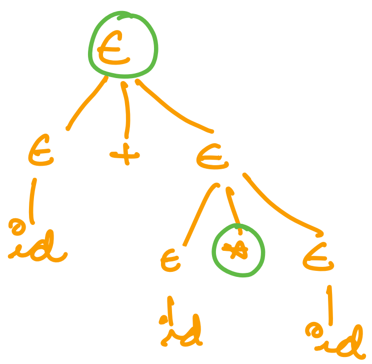
$2^{(3^5)}$

Right

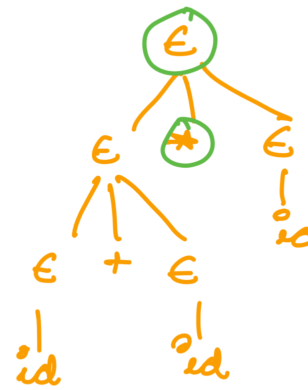
Case 2

Precedence

id + id * id



far away from start symbol

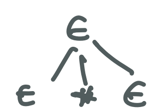


closer to start symbol

same level

$E \rightarrow E * E \mid E + E \mid id$

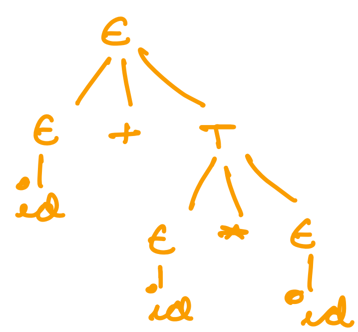
$\Downarrow$



$E \rightarrow E + T \mid T$

$T \rightarrow E * E$ → extra level

$E \rightarrow id$



Ambiguous Situation → Associativity → Left Recur. (Left Ass.)

→ Right Recursive (Right Ass.).

Ambiguous Situation → Precedence