Seminar 7

CONTEX-FREE GRAMMARS (CFG)

- Productions of the form $A \to \alpha$, $A \in N$ and $\alpha \in (N \cup \Sigma)^*$

SYNTAX TREE (PARSE TREE)

→ result of parsing (syntactic analysis)
→ S is the root, nodes $\in N \cup \Sigma$

! In a CFG, a word $w \in L(G) \iff \exists$ a syntax tree with frontier w

EX 1 Given the CFG below, give the leftmost and rightmost derivations for w and draw the syntax tree.

a) $G = (\{S, A, B\}, \{0, 1\},$
$\{S \to 0B^{1} \mid 1A^{2}$
$A \to 0^{3} \mid 0S^{4} \mid 1AA^{5}$
$B \to 1^{6} \mid 1S^{7} \mid 0BB^{8}\})$

$w = 000110\,1110$

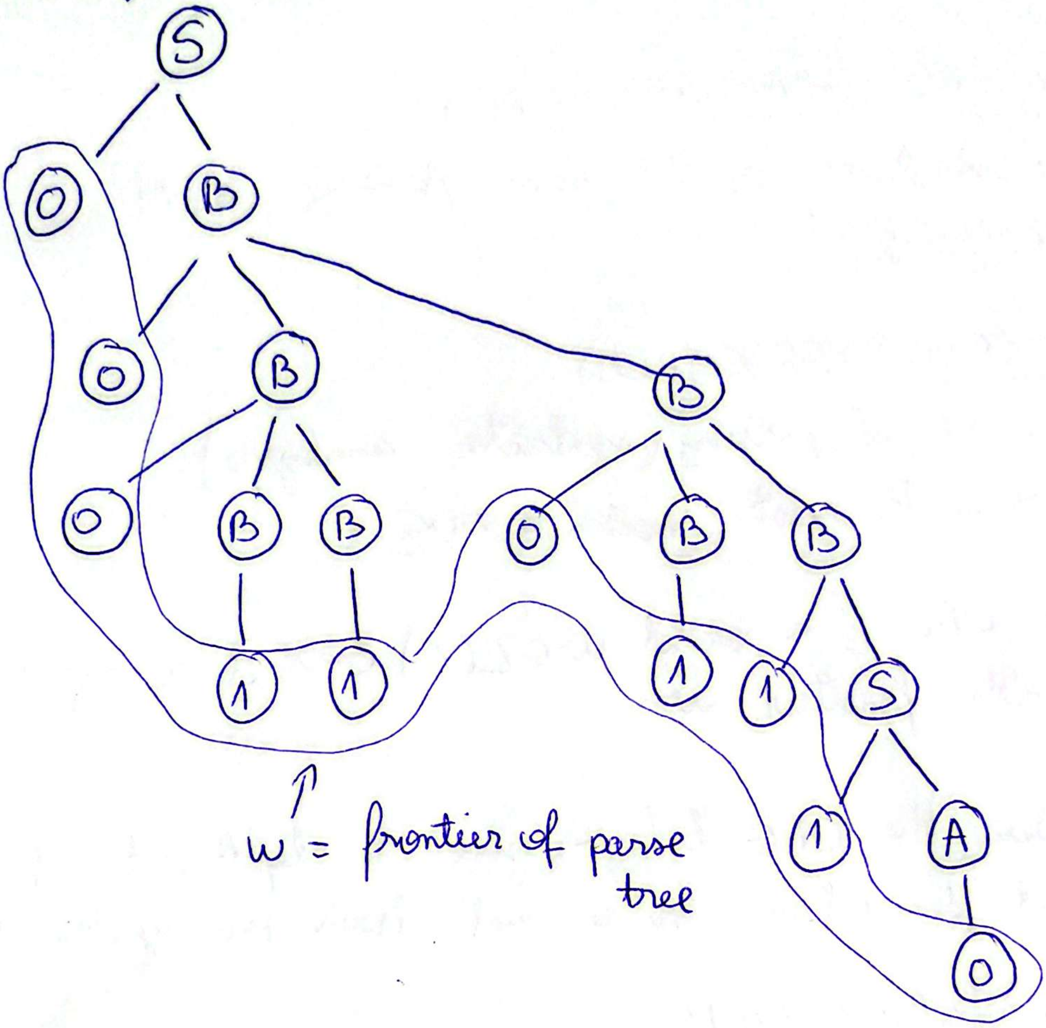
- leftmost: 1886686723

$S \xrightarrow{1} 0\underline{B} \xrightarrow{8} 00\underline{B}B \xrightarrow{8} 000\underline{B}BB \xrightarrow{6} 0001\underline{B}B \xrightarrow{6} 00011\underline{B} \to$
$\xrightarrow{8} 000110\underline{B}B \xrightarrow{6} 0001101\underline{B} \xrightarrow{7} 00011011\underline{S} \xrightarrow{2} 000110111A$
$\xrightarrow{3} 0001101110$

- rightmost: 1887236866

$S \xrightarrow{1} 0\underline{B} \xrightarrow{8} 00B\underline{B} \xrightarrow{8} 00B0B\underline{B} \xrightarrow{7} 00B0B1\underline{S} \xrightarrow{2} 00B0B11\underline{A}$
$\xrightarrow{3} 00B0\underline{B}110 \xrightarrow{6} 00\underline{B}01110 \xrightarrow{8} 000B\underline{B}01110 \xrightarrow{6} 000\underline{B}101110$
$\xrightarrow{6} 0001101110$

parse / syntax tree



b) $G = (\{E, T, F\}, \{a, +, *, (,)\},$

$\{E \to E+T \mid T$

$T \to T*F \mid F$

$F \to (E) \mid a\})$

$w = a*(a+a)$ Homework

Ambigous grammars

Def: A CFG is ambigous if $\exists$ at least one word $w \in L(G)$ that admits two distinct syntax trees.

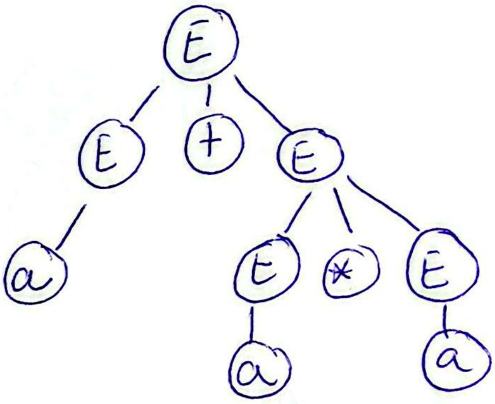
Ambigous language = language generated by an ambigous CFG

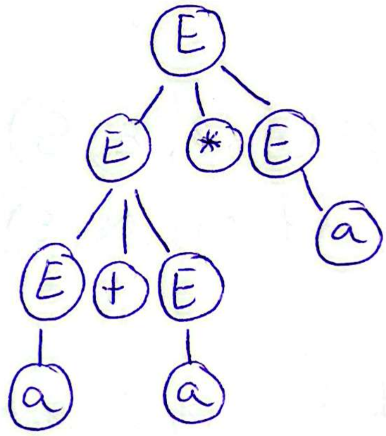
EX 2 Prove that the following grammars are ambigous

~~HW~~ a) $G_1 = (\{S, B, C\}, \{a, b, c\}, \{S \to abC \mid aB,\ B \to bC,\ C \to c\}, S)$

b) $G_2 = (\{E\}, \{a, +, *, (,)\}, \{E \to E+E \mid E*E \mid (E) \mid a\})$

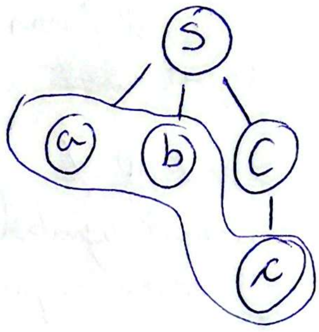
b) $w = a*a+a$



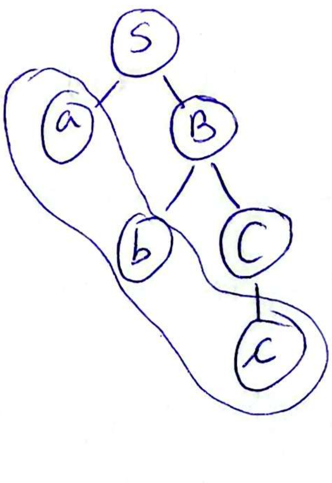


a) $w = abc$

$S \to abC \to abc$



$S \to aB \to abC \to abc$



Scanat cu CamScanner

c) $G_3 = (\{S\}, \{if, then, else, a, b\},$
$\{S \to if\ b\ then\ S \mid if\ b\ then\ S\ else\ S \mid a\}, S)$ HW

RECURSIVE DESCENT PARSER

Configuration

working stack

(s, i, α, β)

parsing state:

input stack
part of the tree to be built

q = normal
b = back
f = final
e = error

position of current symbol in the sequence

- initial config: $(q, 1, \varepsilon, S)$
- final config: $(f, n+1, \alpha, \varepsilon)$

Moves

- EXPAND if head of input stack is nonterminal

$$(q, i, \alpha, A\beta) \vdash (q, i, \alpha A_1, \gamma_1 \beta)$$

where $A \to \gamma_1 | \gamma_2 | \dots$

- ADVANCE if head of input stack is terminal = current symbol from input ~~stack~~

$$(q, i, \alpha, a_i \beta) \vdash (q, i+1, \alpha a_i, \beta)$$

- MOMENTARY INSUCCESS if head of input stack is terminal ≠ current symbol from input

$$(q, i, \alpha, a_i \beta) \vdash (b, i, \alpha, a_i \beta)$$

- BACK if head of working stack is a terminal

$(b, i, \alpha a, \beta) \vdash (b, i-1, \alpha, a\beta)$

- ANOTHER TRY if head of working stack is a non-terminal

$(b, i, \alpha A_j, \gamma_j \beta) \vdash (q, i, \alpha A_{j+1}, \gamma_{j+1} \beta)$ if $\exists A \to \gamma_{+1}$

$\vdash (b, i, \alpha, A\beta)$ otherwise

- SUCCESS

$(q, n+1, \alpha, \varepsilon) \vdash (f, n+1, \alpha, \varepsilon)$

What you have to know:

- from state q you can
  - expand
  - advance
  - momentary insuccess → will chage the state to b

- from state b you can
  - another try
    - ↳ if you still have productions for that non terminal it will change the state back to q
    - ↳ if not state b remains
  - back

Scanat cu CamScanner

Recursive descent parser example

$G = (N, \Sigma, P, S)$, $N = \{S\}$, $\Sigma = \{a, b, c\}$

$$P: \quad S \to aSbS \ (1)$$
$$S \to aS \ (2)$$
$$S \to c \ (3)$$

$w = aacbc$

$$(q, 1, \varepsilon, S) \xmapsto{\text{expand}} (q, 1, S_1, \underline{aSbS}) \xmapsto{\text{advance}} (q, 2, S_1 a, SbS)$$

$$\xmapsto{\text{expand}} (q, 2, S_1 a S_1, \underline{aSbS}\, bS) \xmapsto{\text{advance}} (q, 3, S_1 a S_1 a, SbSbS)$$

$$\xmapsto{\text{expand}} (q, 3, S_1 a S_1 a S_1, \underbrace{aSbS}_{\neq c}\, bSbS) \xmapsto[\text{insuccess}]{\text{momentary}} (b, 3, S_1 a S_1 a S_1, aSbSbSbS)$$

$$\xmapsto[\text{try}]{\text{another}} (q, 3, S_1 a S_1 a S_2, \underbrace{aS}_{\neq c}\, bSbS) \xmapsto[\text{insuccess}]{\text{momentary}}$$

$$(b, 3, S_1 a S_1 a S_2, aSbSbS) \xmapsto[\text{try}]{\text{another}} (q, 3, S_1 a S_1 a S_3, \underline{c}\, bSbS)$$

$$\xmapsto{\text{advance}} (q, 4, S_1 a S_1 a S_3 c, bSbS) \xmapsto{\text{advance}} (q, 5, S_1 a S_1 a S_3 cb, SbS)$$

$$\xmapsto{\text{expand}} (q, 5, S_1 a S_1 a S_3 cb S_1, \underbrace{aSbS}_{a \neq c}\, bS) \xmapsto{\text{m.i}}$$

$$(b, 5, S_1 a S_1 a S_3 cb S_1, aSbSbS) \xmapsto[\text{try}]{\text{another}}$$

$$(q, 5, S_1 a S_1 a S_3 cb S_2, \underbrace{aS}_{a \neq c}\, bS) \xmapsto{\text{m.i}} (b, 5, S_1 a S_1 a S_3 cb S_2, aSbS)$$

$$\xmapsto[\text{try}]{\text{another}} (q, 5, S_1 a S_1 a S_3 cb S_3, \underline{c}\, bS) \xmapsto{\text{advance}} (q, 6, S_1 a S_1 a S_3 cb S_3 c, bS)$$

we parsed the entire word and we still have elements in β

$\Rightarrow$ we have to go back.

! β has to be empty at the end

$\xrightarrow{mi} (b, 6, S_1aS_1aS_3cbS_3\cancel{c}, bS)$

$\xrightarrow{back} (b, 5, S_1aS_1aS_3cbS_3, \cancel{c}bS)$

because here we have a non-terminal we have to do another try, but S_3 is the last production of S (we don't have $S_4, S_5 \dots$, so we will move S_3 back to β and delete its production. This operation is also called another try)

$\xrightarrow{another\ try} (b, 5, S_1aS_1aS_3cb, SbS)$

↳ because another try did not used other productions of S and put S_3 in β, the state stays as b

$\xrightarrow{back} (b, 4, S_1aS_1aS_3c, bSbS)$

when we are in state b and at end of α is a terminal the only possible move is back

$\xrightarrow{back} (b, 3, S_1aS_1aS_3, \cancel{c}bSbS)$

this is non-terminal and we are still in state b $\Rightarrow$ the only move we can do is another try

$\xrightarrow{another\ try} (b, 3, S_1aS_1a, SbSbS)$

$\xrightarrow{back} (b, 2, S_1aS_1, \underbrace{aSbS}bS)$

$\xrightarrow{another\ try} (q, 2, S_1aS_2, \underbrace{aS}bS)$

↳ state changed to q again because we had S_2 to use

$\xrightarrow{advance} (q, 3, S_1aS_2a, SbS) \xrightarrow{expand} (q, 3, S_1aS_2aS_1, \underbrace{aSbS}_{a \neq c}bS)$

$\xrightarrow{m.i} (b, 3, S_1aS_2aS_1, aSbSbS)$

$\xrightarrow{another\ try} (q, 3, S_1aS_2aS_2, \underbrace{aSbS}_{a \neq c}) \xrightarrow{mi} (b, 3, S_1aS_2aS_2, \underbrace{aS}bS)$

$\vdash^{\text{another}}_{\text{try}} (q, 3, S_1 a S_2 a S_3, cbS)$

$\vdash^{\text{advance}} (q, 4, S_1 a S_2 a S_3 c, bS)$

$\vdash^{\text{advance}} (q, 5, S_1 a S_2 a S_3 cb, S)$

$\vdash^{\text{expand}} (q, 5, S_1 a S_2 a S_3 cb S_1, aSbS)$

$\vdash^{\text{mi}} (b, 5, S_1 a S_2 a S_3 cb S_1, aSbS)$

$\vdash^{\text{another}}_{\text{try}} (q, 5, S_1 a S_2 a S_3 cb S_2, aS)$

$\vdash^{\text{mi}} (b, 5, S_1 a S_2 a S_3 cb S_2, aS)$

$\vdash^{\text{another}}_{\text{try}} (q, 5, S_1 a S_2 a S_3 cb S_3, c)$

$\vdash^{\text{advance}} (q, 6, S_1 a S_2 a S_3 cb S_3 c, \varepsilon)$

$\vdash^{\text{success}} (f, 6, S_1 a S_2 a S_3 cb S_3 c, \varepsilon)$

$\Rightarrow$ w is syntactically correct

parse tree: $S_1 S_2 S_3 S_3$