Comp 330 - Lecture 15 - Oct 24th

IE : 100 niente uccisero l'asino
100 nothings killed the donkey
Don't take on too much work/
more than you can
handle

A4 is out

Properties of & Decision Problems for CFGs/CFLs

Equivalent definitions to $L(G)$ — G is CFG

$$L(G) = \{ w \in T^* : S \overset{*}{\underset{G}{\Rightarrow}} w \}$$

Leftmost & rightmost derivations

Ex G: $S \to SS \mid aSb \mid bSa \mid \varepsilon$

Rightmost derivation of abba

$$S \to SS \to SbSa \to SbEa \to aSbba \to abba$$

$S \overset{*}{\underset{G\,rm}{\Rightarrow}} w \to$ rm = rightmost

Leftmost

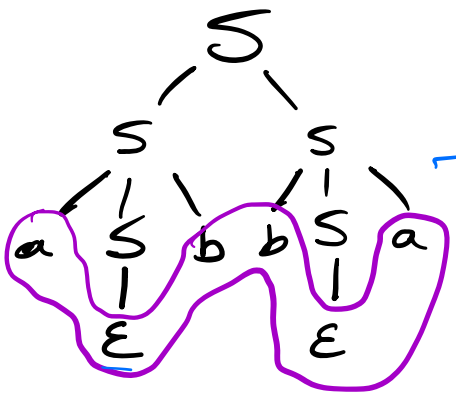
$$S \to SS \to aSbS \to aEbS \to abbSa \to abba$$

$S \overset{*}{\underset{G\,lm}{\Rightarrow}} w$

Parse trees

$G: S \to SS \mid aSb \mid bSa \mid \varepsilon$

The parse tree of G for $w = abba$



$\longrightarrow S \to bSa$

yield of the parse tree of G.

Who cares?

Thm (All these representations are equivalent)

Given a CFG $G = (V, S, T, P)$, $w \in T^*$

The following statements are equivalent

1) $S \overset{*}{\underset{G}{\to}} w$

2) $S \overset{*}{\underset{rm}{\to}} w$

3) $S \overset{*}{\underset{lm}{\Rightarrow}} w$

4) w is the yield of a parse tree of G.

Implications : 2, 3, 4 allow us to state new properties / lemmas for CFLs / CFGs.

Ambiguity

Ex $L = \{ w \in \{0, 1, \ldots, 9, +, \times, (,)\}^* : w$ is a valid arithmetic expression of single digits $\}$

Σ^*

1. L is NOT REG → Closure properties

2. L is CF. Here is a CFG

$G = (V = \{E, N\}, S = E, T = \Sigma, P)$

P

$E \to E \times E \mid E + E \mid (E) \mid N$

$N \to 0 \mid 1 \mid \ldots \mid 19.$

$1+1 \times 2$

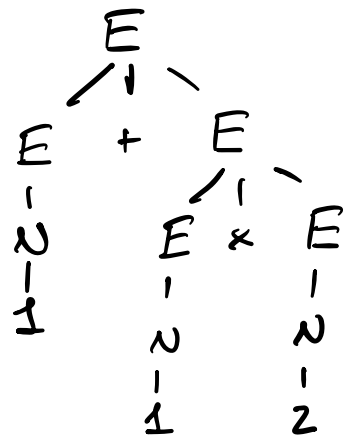
$E \to E+E \to N+E \to 1+E \to 1+E \times E$

$\overset{*}{\Rightarrow} 1+N \times N$

$\overset{*}{\Rightarrow} 1+1 \times 2.$

Consider a first parse tree of G for the string $1+(1\times 2)$

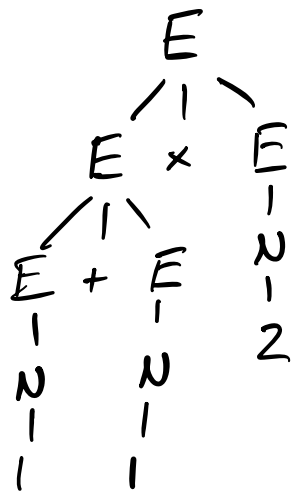
$1 + 2 = 3$



This is the correct interpretation because propagating up the result is 3.

However there is an alternative parse tree

$1+1\times 2$



This is the incorrect interpretation of the string

Def A CFG $G = (V, S, T, P)$ is ambiguous if $\exists\ w \in T^*$ with 2 or more distinct parse trees with yield w.

↳ (Not isomorphic)

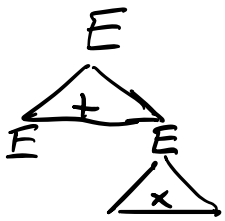
In this case, it is possible to disambiguate the grammar G

$$E \to E+E \mid T$$
$$T \to T \times T \mid F$$
$$F \to (F) \mid N$$
$$N \to 0 \mid 1 \mid 2 \mid \dots \mid 9$$

Heuristic:
lowest precedence first
highest precedence last.

Ⓐ There is a problem with this grammar. How do generate (1+1)×2?

$1+1\times 2$



Converting a CFG G to an unambiguous grammar G is not always possible.

Def $\Sigma \neq \emptyset$, $L \subseteq \Sigma^*$, L is inherently ambiguous if $\forall$ CFG G s.t. $L(G) = L$, G is ambiguous.

L_1 = $\{a^n b^n c^m d^m : n, m \geq 1\}$ ∪ $\{a^n b^m c^m d^n : n, m \geq 1\}$ = L_2

Ex $L = \{a^n b^n c^m d^m : n, m \geq 1\} \cup \{a^n b^m c^m d^n : n, m \geq 1\}$

- L is CF
- L is inherently ambiguous.

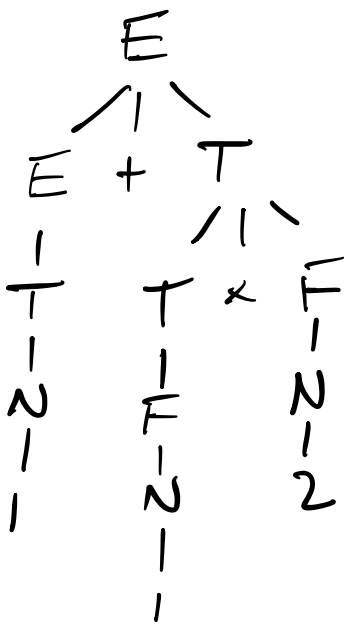
There must always be 2 distinct

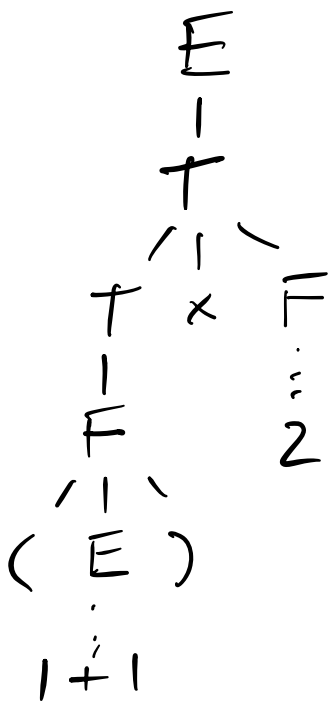
Fix this by allowing to go back up the tree but only from the highest precedence

$E \to E + T \mid T$
$T \to T \times F \mid F$
$F \to (E) \mid N$
$N \to 0 \mid 1 \mid 2 \ldots 19$

} Allows for same strings as G.

$1+1\times2$ & $(1+1)\times2$





parse trees for $w = a^n b^n c^n d^n \quad n \geq 1$
for any correct CFG G s.t. $L(G) = L$.
↳ Extremely challenging exercise.

Thm A grammar G is ambiguous
This is an iff. → iff $\exists\ w \in T^*$ with at least 2 distinct leftmost (rightmost) derivations.

(*) 1 leftmost & 1 rightmost $\not\Rightarrow$ ambiguous

Pf.

Normal forms

Def (Chomsky Normal Form) A CFG $G = (V, S, T, P)$ is in CNF if $\forall\ \alpha \to \beta \in P$
$\alpha \in V$
either $\beta \in V \cdot V$
$\beta \in T$

Ex

$S \to AB$ ✓
$S \to a$ ✓

$S \to aSb$ ×
$S \to aSS$ ×
$S \to aaa$ ×

Def (Greibach Normal Form) A CFG $G=(V,S,T,P)$ is in GNF if $\forall\ \alpha \to \beta \in P$

$\alpha \in V \quad \beta \in T \cdot V^*$

Ex

$S \to a$ ✓

$S \to aAB$ ✓

$S \to AB$ ✗ $\quad S \to Aa$ ✗

$S \Rightarrow aSb$ ✗

Thm Let $G=(V,S,T,P)$ be a CFG. Then $\exists$ grammars G' & G'' is CNF & GNF respectively s.t.

$$L(G') = L(G'') = L(G) - \{\varepsilon\}$$

Pf idea **Ex** Convert the following grammar G to CNF.

$S \to AB \mid ABCD$ ✓

$A \to aAa \mid a$ ✓

$B \to b$ ✓

$C \to ba$

$D \to a$ ✓

$\Rightarrow$

For ~~$C \to ba$~~

$T_a \to a \quad T_b \to b$

$C \to T_b T_a$

~~$A \to a A a$~~

$A \to T_a A T_a$

$V_1 \to T_a A$

$A \to V_1 T_a$

$S \to ABCD$

$V_1 \to AB \quad S \to V_1CD$

$V_2 \to V_1C \quad S \to V_2D$

What if $A \to \varepsilon$ or $A \to B$

$A \to BCD \mid \ldots \mid CD \mid BD \mid D$

$B \to \varepsilon \mid \ldots$

$C \to \varepsilon \mid \ldots$

$A \to B \mid \ldots \mid aA \mid bB \mid \ldots$

$B \to aA \mid bB \mid \ldots$

Decision problems & procedures for CFGs / CFLs

DP: Given a CFG G, is $w \in L(G)$?

↓ Decision problem

1. Brute force:
   - A. Convert CFG G to CNF
   - B. Create all of the parse trees

with yield length of w. → Catalan numbers ~ Exponential in |w|.

C. For each parse tree, check if the yield is w.

Itiroo Sakai 1961

2. CYK Algorithm - Cocke - Younger - Kasami

Relies on the form of production rules in CNF.

Consider the CFG G in CNF

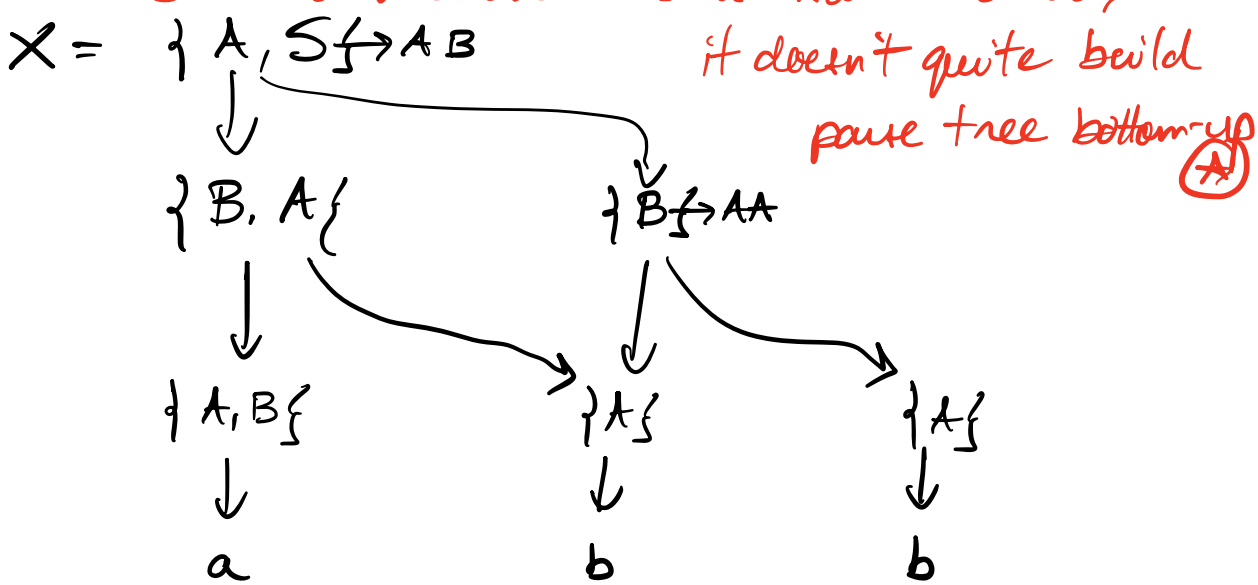
$S \to AB$

$A \to BA \mid BB \mid a \mid b$

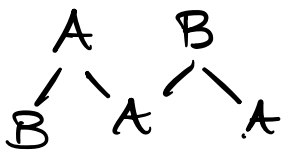
$B \to AA \mid a$

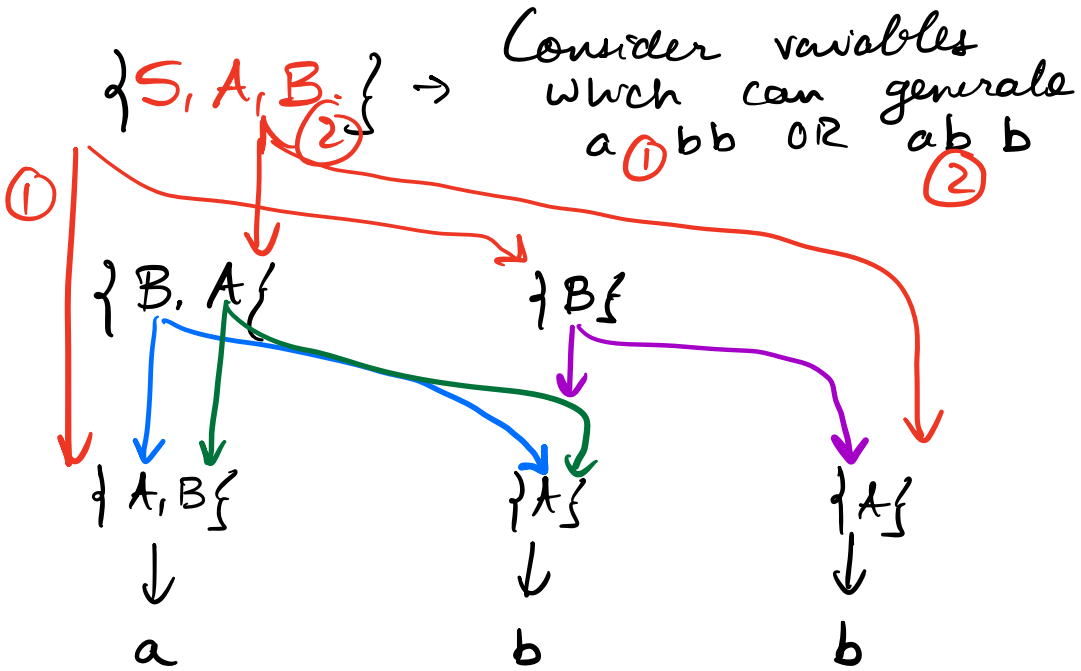
Using CYK, check whether $abb \in L(G)$.

(*) This version has a subtle error, it doesn't quite build parse tree bottom-up (*)



You can't have a parse tree with





① Consider {A, B{ · {B{ = {AB, BB{
S, A

② Consider {B, A{ · {A{ = {BA, AA{
B, A

Since $S \in X$, abb can be generated by G.

Can show that this alg is $O(|w|^3)$

DP: Deciding whether $L(G) = \phi$.

1. Brute force
   A. Convert G to a grammar in CNF G'
   B. for $w \in T^*$ → This loops forever!
      call CYK on (G', w)
      if $w \in L(G')$ return false
   return true

2. Check if S is a generating variable
   i.e. if $\exists\ w \in T^*$ s.t. $S \overset{*}{\Rightarrow} w$
   1. GEN ← T
   2. Repeat until no change in GEN
      for each $X \to \alpha$ do
         if every symbol in α is in GEN then
            add X to GEN
   3. Check if S is in GEN

Most questions about CFG & CFL are undecidable.

Given G_1, G_2, is $L(G_1) = L(G_2)$? } VAL-COMPS

G_1, is $L(G_1) = \Sigma^*$? }

G_1, is G_1 ambiguous ?

Exercise Are the analogous L.D.P. for REG decidable ?