COMP 330

Name: Yuhao Wu
ID Number: 260711365

2018-10-15

October 15th Lecture 14

Context-Free Grammar(CFG) Definition: A CFG contains a set of symbols:

- a set of symbols Σ *or* T called terminals.
- another set of symbols V called non-terminals $\quad V \cap T = \emptyset$
- A special $S \in V$, which is the start symbol
- A finite set of rules/productions of the form:

$$A \in V \qquad A \to \alpha \qquad \alpha \in (V \cup T)^*$$

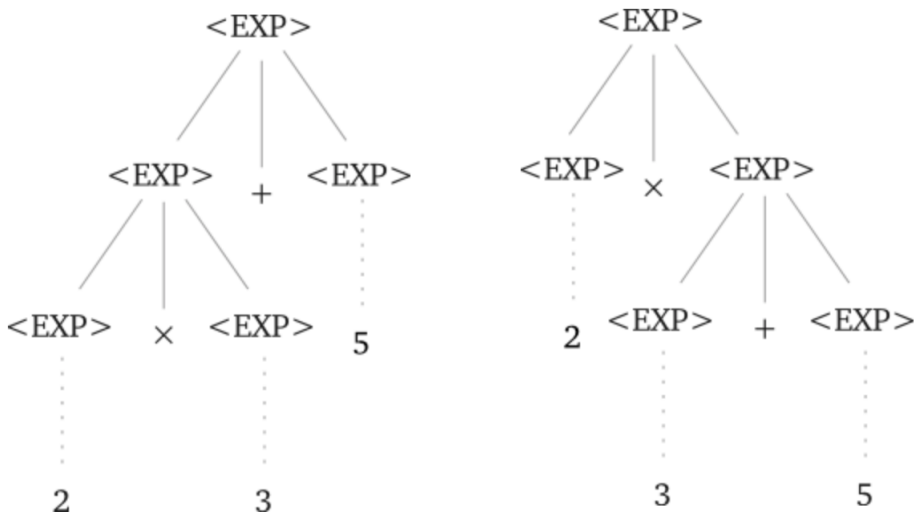
Example:
$T = \{ a, b \}$ $\quad V = \{ S \}$ $\quad L = \{a^n\ b^n \mid n \in \mathbb{N} \}$
Rule: $(1) S \to \varepsilon$ $\quad (2) S \to aSb$

REMARK: Context-free language may not be regular language.
All regular languages are context-free.

Language with arithmetic expressions:

- $T = \{ 0, 1, 2, \ldots, 9, +, \times, (,) \}$
- $V = \{ \langle EXP \rangle, \langle NUM \rangle, \langle NZ \rangle, \langle N \rangle \}$ start symbol: $\langle EXP \rangle$
- $\langle EXP \rangle \to \langle EXP \rangle + \langle EXP \rangle \mid \langle EXP \rangle \times \langle EXP \rangle \mid (\langle EXP \rangle) \mid \langle NUM \rangle$
- $\langle NUM \rangle \to 0 \mid \langle NZ \rangle$
- $\langle NZ \rangle \to 1\langle N \rangle \mid 2\langle N \rangle \mid \ldots \mid 9\langle N \rangle \mid \varepsilon$
- $\langle N \rangle \to 1\langle N \rangle \mid 2\langle N \rangle \mid \ldots \mid 9\langle N \rangle \mid \varepsilon$

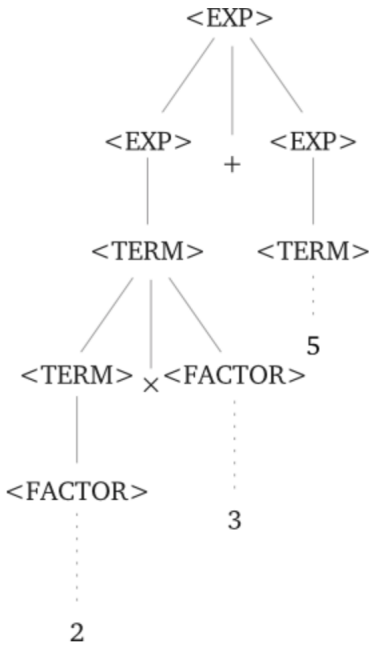
As we can see, there are two possible trees there.



October 17th Lecture 15
So, we need to build a new grammar:

- V (non-terminals) $= \{ \langle EXP \rangle, \langle TERM \rangle, \langle FACTOR \rangle \}$
- Start Symbol $\langle EXP \rangle$
- $< EXP > \rightarrow < EXP > + < TERM > \mid < TERM >$
- $< TERM > \rightarrow < TERM > \times < FACTOR > \mid < FACTOR >$
- $< FACTOR > \rightarrow < NUM > \mid (< EXP >)$

Then, the only derivation tree for $2 \times 3 + 5$ is the following:



We couldn't generate a different tree for the expression if we tried - if we tried to group it the other way as we did above (by evaluating 3+5 first), then we would generate the expression $2 \times (3+5)$, which is not the same string.

REMARK: Tree structure is important for meaning.

EXAMPLE 1:
$L = \left\{ a^n b^{2n} \mid n \geq 0 \right\}$ $S \to aSbb \mid \varepsilon$

EXAMPLE 2:
$L = \{ X \mid X = X^{rev} \}$ $\Sigma = \{ a, b \}$ $S \to aSa \mid bSb \mid \varepsilon \mid a \mid b$

EXAMPLE 3:
$L = \{ X \in \Sigma^* \mid X$ has as many a's as b's $\}$ $\Sigma = \{ a, b \}$
We define a function $d(x) = \#b(x) - \#a(x)$, then $L = \{ x \mid d(x) = 0 \}$

Suppose that $u \in L$, then we can say u has the same number of a's as the number of b's.
And we assume that u starts with an a $u = aw$ $d(w) = 1$

Let x be the **shortest** prefix such that $d(x) = 0$, then we know that the last letter of x is b.
If the last letter of x is a, then we know that it can't be the shortest prefix such that $d(x) = 0$, there must exist shorter prefix.
Suppose $u = xy$, since for x, we have $d(x) = 0$, we also have $d(y) = 0$

x must be in this form $x = aVb$ then we have $u = xy = aVby$ $V, y \in L$

Thus, we have:

$$A: \quad S \to aSbS \mid bSaS \mid \varepsilon \; or$$
$$B: \quad S \to aSb \mid bSa \mid SS \mid \varepsilon$$

- Every word generated is in L
- Every word in L can be generated

For B:
Obviously, any word generated by B has the same number of a's and b's

Now, we need to show that any word with same number of a's and b's can be generated.
We prove by induction of the number of a's.

Base Case: ε, obviously, this is true.

Inductive Step: assume that the word contains n a's and n b's can be generated.
Consider string with $(n+1)$ a's and $(n+1)$ b's:

$$awb \implies S \to aSb \qquad bwa \implies S \to bSa$$

Suppose that w begins and ends with a $w = aw'a$, which means w' has 2 more b's than a's.
Thus, we can separate w' as $w' = w'_1 w'_2$, where w'_1 has one more b than a so is w'_2.
Thus, we can write w' as $w' = aw'_1 w'_2 a$ where aw'_1 has same number of a and b so is w'_2
According to our hypothesis, we can rewrite aw'_1 according to rule $S \to aSb$ we can rewrite $w'_2 a$ according to rule $S \to bSa$, which means w can be rewritten as $w = SS$

FACT: If you have a function behave like $d(x)$ defined before, which can only change by ± 1 at each step, it starts at negative and ends up at positive, then it must hit 0.

Chomsky Normal Form:

Every CFL has a grammar in which the rules are of a very restricted form, which is:

- $A \to a$
- $A \to BC$

Where A, B, C are non-terminal.
Note that $A \to \varepsilon$ is not allowed except when $A = S$.

October 19th Lecture 16

Recognize a CFL

Push Down Automata PDA (stack machine) $\implies$ DFA + 1 stack
Recall: $\{a^n b^n \mid n \geq 0\}$, which is non-regular
CFL recognized by PDA:

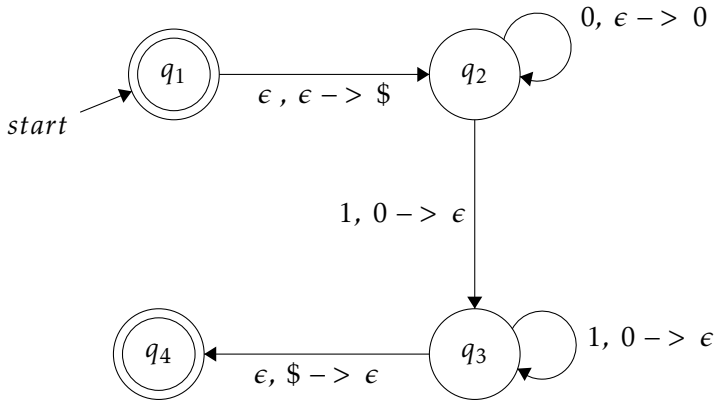
- Q: states Σ: alphabet(input) Γ: stack alphabet
- q_0: start state $F \subseteq Q$ accept
- $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$ $\Gamma_\varepsilon = \Gamma \cup \{\varepsilon\}$
- $\delta : Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon = \mathcal{P}_{finite}(Q \times \Gamma_\varepsilon)$ finite power set

Look at input and top of stack, then pop the stack, push a new symbol, change state.

We decorate arrows with $a, b \to c$
See a in the input, b on the top of the stack $\implies$ replace b with c on the stack

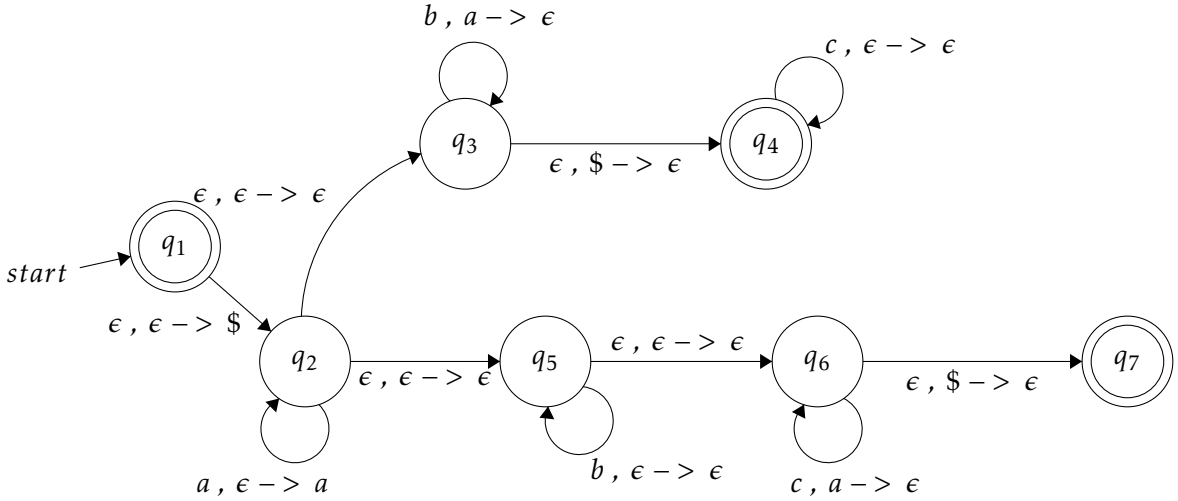
$a = \varepsilon$ don't look at input $b = \varepsilon$ just push c $c = \varepsilon$ just pop b

$L = \{ 0^n 1^n | n \geq 0 \}$



$: special symbol to indicate the end of stack

$L = \{ a^i b^j c^k \mid i,j,k \geq 0 \quad i = j \text{ or } i = k \}$



Theorem:

- Every CFL is recognized by some PDA
- Every language recognized by a PDA is a CFL
- the intersection of a CFL and a regular language = CFL
- the intersection of 2 CFL MAY NOT BE a context free.