# Lecture 10

- HW2 out (due Monday Oct 16)

This week
- Today: CFG's and Pumping Lemma for CFL's
- Wednesday: Pumping Lemma, Practice Problems

Next week
- Mon: Equivalence Between PDA and CFGs
- Wed: Review for Test 1

10/23
- Mon: Test 1 (in class)

## Example 4

$\mathcal{G} = ( V = \{E\}, \Sigma = \{a, b, +, *, (, )\}, R, E )$

$R:$ $E \to E+E \mid E*E \mid (E) \mid a \mid b$

**Derivation for $a+b*a \in \mathcal{L}(\mathcal{G})$:**

$E \to E+E \to E+E*E \to a+E*E \to a+b*E \to a+b*a$

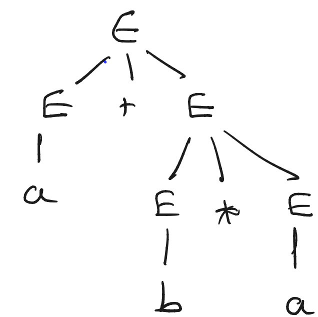
# Example 4

$\mathcal{G} = ( V = \{E\}, \ \Sigma = \{a, b, +, *, (, )\}, \ R, \ E \}$

$R:$ $\quad E \rightarrow E + E \mid E * E \mid (E) \mid a \mid b$

## Derivation # 1 for $a + b * a \in \mathcal{L}(\mathcal{G})$:

$E \rightarrow E + E \rightarrow E + E * E \rightarrow a + E * E \rightarrow a + b * E \rightarrow a + b * a$

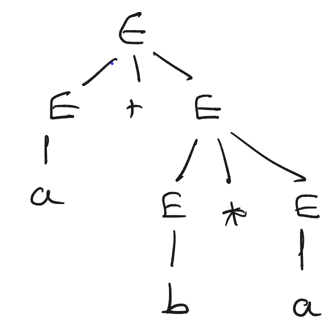


Derivation tree

## Example 4

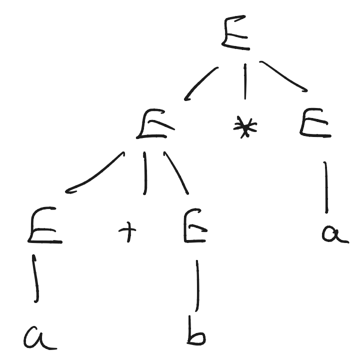
$G = (V = \{E\}, \Sigma = \{a, b, +, *, (, )\}, R, E)$

$R: \quad E \to E+E \mid E*E \mid (E) \mid a \mid b$



**Derivation #1 for $a+b*a$:**

$E \to E+E \to E+E*E \to a+E*E \to a+E*a \to a+b*a$



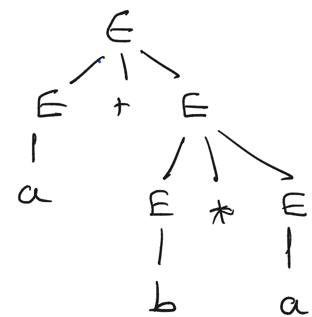
**Derivation #2 for $a+b*a$:**

$E \to E*E \to E*a \to E+E*a \to a+E*a \to a+b*a$

**Defn.** A leftmost derivation is a derivation where at each step, we replace the leftmost variable

⇒ Derivations #1 and #2 were not Leftmost.
The corresponding Leftmost derivations are:

**Derivation #1** $(E \to E+E \mid E*E \mid (E) \mid a \mid b)$



$E \to E+E \to a+E \to a+E*E \to a+b*E \to a+b*a$

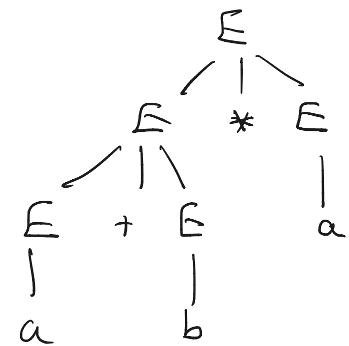
**Corresponding Leftmost:**

$E \to E+E \to E+E*E \to a+E*E \to a+E*a \to a+b*a$

**Defn.** A leftmost derivation is a derivation where at each step, we replace the leftmost variable

**Derivation #2**

$E \to E * E \to E * a \to E + E * a \to a + E * a \to a + b * a$




**Corresponding Leftmost:**

$E \to E * E \to E + E * E \to a + E * E \to a + b * E \to a + b * a$

**Claim** There is a 1-1 correspondence between a derivation tree and a leftmost derivation

# Ambiguous vs UnAmbiguous grammars

Defn A CFG $G$ is ambiguous if there exists some $w \in \mathcal{L}(G)$ such that $w$ has more than one different derivation trees ($\equiv$ more than one leftmost derivation)

Example 4 is ambiguous since we just saw that $w = a+b*a$ has 2 different derivation trees

# Ambiguous vs UnAmbiguous grammars

**Defn** A CFG $\mathcal{G}$ is ambiguous if there exists some $w \in \mathcal{L}(\mathcal{G})$ such that $w$ has more than one different derivation trees ($\equiv$ more than one leftmost derivation)

**Example 4** $\mathcal{G}$: $E \to E+E \mid E*E \mid (E) \mid a \mid b$ is ambiguous since $w = a+b*a$ has 2 different derivation trees

**Define** : $\mathcal{G}'$ :

$$E \to E+F \mid F$$
$$F \to F*\mathcal{G} \mid \mathcal{G}$$
$$\mathcal{G} \to (E) \mid a \mid b$$

**Claim** $\mathcal{G}'$ is unambiguous, and $\mathcal{L}(\mathcal{G}) = \mathcal{L}(\mathcal{G}')$

# Ambiguous vs UnAmbiguous grammars

**Defn** A CFG $\mathcal{G}$ is ambiguous if there exists some $w \in \mathcal{L}(\mathcal{G})$ such that $w$ has more than one different derivation trees ($\equiv$ more than one leftmost derivation)

**Defn** A context free language $L$ is inherently ambiguous if every CFG that generates $L$ is ambiguous.

# Ambiguous vs UnAmbiguous grammars

**Defn** A context free Language $L$ is *inherently ambiguous* if every CFG that generates $L$ is ambiguous.

**Example** $L = \{a^n b^n c^m d^m \mid n, m \geq 0\} \cup \{a^n b^m c^n d^m \mid n, m \geq 0\}$ is inherently ambiguous

**Claim 1** $L$ is a CFL (Prove as an exercise)

**Claim 2** (idea): show that any $w$ of the form $a^n b^n c^n d^n$, $n \geq 2$ will always have at least 2 different derivation trees

# Context-Free Languages + PushDown Automata

Now we will define a larger class of languages that includes all regular languages plus new ones.

✓ 1 We will first define CFL's to be those languages accepted by PUSHDOWN AUTOMATA (PDA)

✓ 2 Then we give an alternative characterization of CFLs
Language/generation Model: Context Free grammars (CFGs)

will prove later → 2a We will prove these 2 characterizations are equivalent:

$\underbrace{\text{PUSHDOWN AUTOMATA (PDA)}}_{\text{Machine Model}} \equiv \underbrace{\text{Context Free grammars (CFGs)}}_{\text{Language/generation Model}}$

Next → 3 Pumping Lemma (for CFL's): used to prove that some languages are Not Context Free Languages

# Pumping lemma for CFLs

We will describe a property of any CFL, similar to how we explored the finite state property for regular languages.

However, for CFLs it is a bit easier to extract this property from the context-free-grammar. (Since Languages are context Free $\Leftrightarrow$ they have a CFG $\Leftrightarrow$ they have a PDA, it is fine to work in either model.)

Recall for DFAs: every DFA $M$ has a finite number of states:

Let $M$ be a $k$-state DFA. Then for every $w \in \Sigma^*$, if $|w| \geq k$, $M$ on $w$ will Loop. Therefore we can write $w = xyz$, $|y| \geq 1$, $|xy| \leq k$ such that for all $i \geq 0$ $w' = xy^iz$ will be accepted by $M$ if and only if $w$ is accepted by $M$.

---

For CFG's we will exploit a similar property: any CFG has a finite number of rules.

Let $G = (V, \Sigma, R, S)$ be a CFG, $|V| = k$.
Then for any $w \in \Sigma^*$ that is generated by $G$, if $|w| > k$
then every derivation of $w$ will repeat some variable.

# Pumping Lemma for Context Free Languages

**Lemma** Let $L$ be a CFL, $L \subseteq \Sigma^*$

Then there exists a number $p \geq 0$ such that for any string $w \in L$, $|w| \geq p$, $\exists$ strings $u, v, x, y, z \in \Sigma^*$ such that:

(0) $w = uvxyz$ and

(1) $|vxy| \leq p$ and $|vy| \geq 1$ and

(2) For every $i \geq 0$, $w^i = uv^i x y^i z$ is also in $L$.

---

Pumping Lemma for regular Languages: $\exists p \ldots$ $|w| \geq p$

can write $w = xyz$, $|xy| \leq p$, $|y| \geq 1$

such that $\forall i$ $w \in L$ iff $w^i = xy^i z \in L$

# Pumping Lemma for CFL's Proof

**Claim** Let $G = (V, \Sigma, R, S)$ be a CFG with $|V| = k$, and let $b$ be the max. length of any string $u \in (V \cup \Sigma)^*$ in a rule.
Then for any string $w$ that is generated by $G$;
if $|w| \geq b^{k+2}$, then any derivation of $w$ must repeat a variable.

we will call this value "p" the pumping length

**Proof** (see book)

$S \to \overbrace{aSb} \mid \overbrace{aRbb}$ $\quad b = 4$

$R \to \underbrace{aba}$

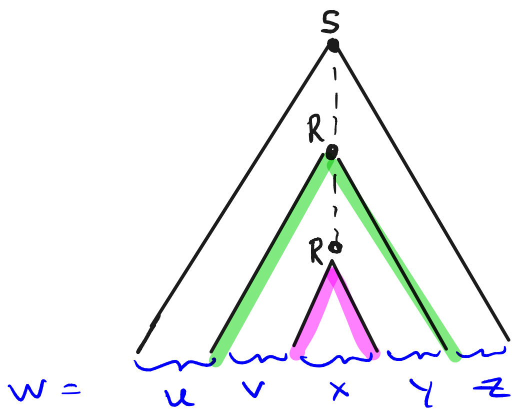
**PL:** $\exists p \geq 0\ \forall w \in \Sigma^*, w \in L$, if $|w| \geq p$ then can write $w = uvxyz$, $|vxy| \leq p$, $|vy| \geq 1$ such that $\forall i \geq 0\ w^i = uv^ixy^iz \in L$.

**Proof of PL:**

Let $L$ be a CFL and Let $G$ be a CFG generating $L$, where #variables in $G$ is $k$ and $b$ = max length of strings in rules.
Let $p = b^{k+2}$, and Let $w \in L$, $|w| \geq p$.
By claim, the derivation tree of $w$ must repeat a variable.

derivation tree for w →

R = repeated variable




$S \overset{*}{\Rightarrow} uRz$

$R \overset{*}{\Rightarrow} vRy$

$R \overset{*}{\Rightarrow} x$

Derivation of w:

$S \overset{*}{\Rightarrow} uRz \overset{*}{\Rightarrow} uvRyz$
$\overset{*}{\Rightarrow} uvxyz$

PL: $\exists p \geq 0\ \forall w \in \Sigma^*$, $w \in L$, if $|w| \geq p$ then can write $w = uvxyz$, $|vxy| \leq p$, $|vy| \geq 1$ such that $\forall i \geq 0$ $w^i = uv^ixy^iz \in L$.

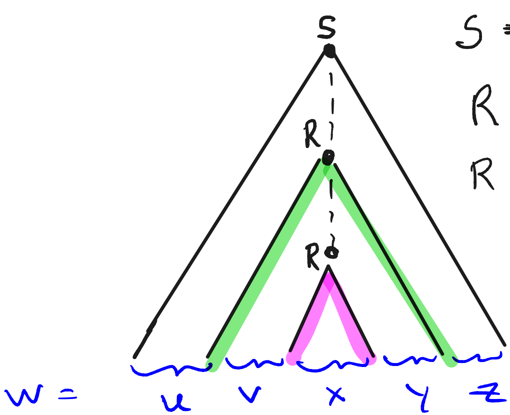
## Proof of PL

Let $L$ be a CFL and let $G$ be a CFL generating $L$, where #variables in $G$ is $k$ and $b$ = max length of strings in rules.

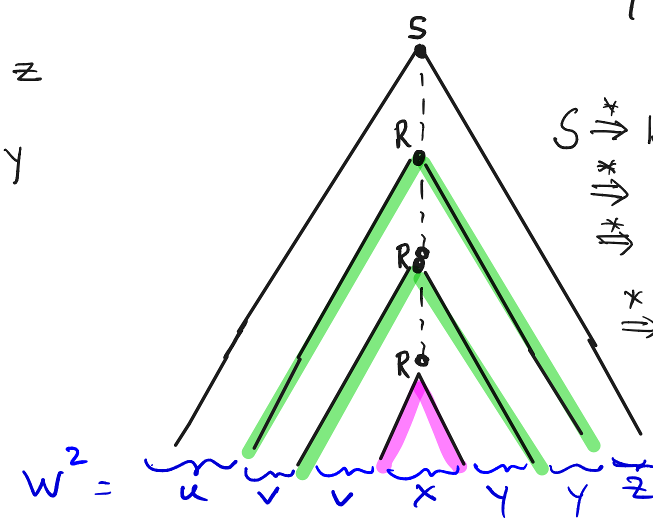
Let $p = b^{k+2}$, and let $w \in L$, $|w| \geq p$.

By claim, the derivation tree of $w$ must repeat a variable, say $R$.

Then $w^2 = uv^2xy^2z$ also has a derivation tree. (and similarly for any $w^i = uv^ixy^iz$)




$$S \overset{*}{\Rightarrow} uRz$$
$$R \overset{*}{\Rightarrow} vRy$$
$$R \overset{*}{\Rightarrow} x$$




$$\begin{aligned} S &\overset{*}{\Rightarrow} uRz \\ &\overset{*}{\Rightarrow} uvRyz \\ &\overset{*}{\Rightarrow} uvvRyyz \\ &\overset{*}{\Rightarrow} \underbrace{uvvxyyz}_{w^2} \end{aligned}$$

**Example:** Let $\mathcal{G} = (V, \Sigma, R, S)$. $V = \{S, A, B, C\}$ $\Sigma = \{0, 1\}$

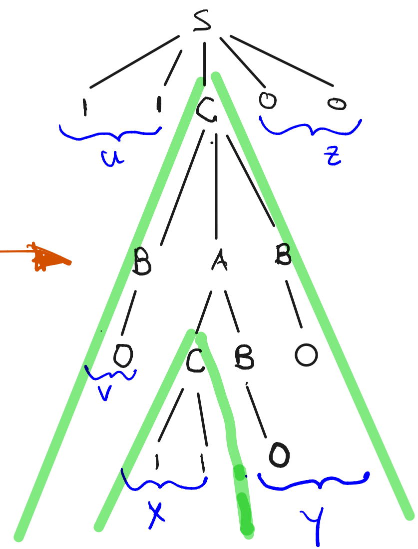
$S \to 11C00$

$C \to BAB \mid 11$

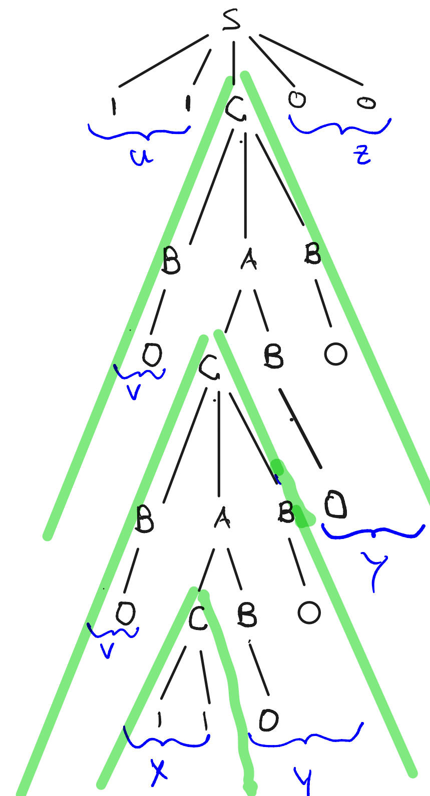
$B \to 0$

$A \to CB$

$w = \underbrace{11}_{u}\,\underbrace{0}_{v}\,\underbrace{11}_{x}\,\underbrace{00}_{y}\,\underbrace{00}_{z}$




$w^2 = u v^2 x y^2 z$ also in $L(\mathcal{G})$



**Example 1** We will show $L = \{a^n b^n c^n \mid n \geq 0\}$ is not a CFL.

Let $w = a^{2p} b^{2p} c^{2p}$, where $p$ is from the pumping lemma

then By pumping Lemma $w = uvxyz$, $|vxy| < p$, $|vy| \geq 1$

Case 1: $v$ and $y$ contain only $a$'s. Then $uv^2xy^2z$ will contain too many $a$'s

Case 2: $v$ and $y$ contain only $b$'s or only $c$'s.
Same arg as Case 1

Case 3 $v$ and $y$ together contain both $a$'s and $b$'s.
then $uv^2xy^2z$ will contain too few $c$'s.
Similar if $v$ and $y$ together contain ~~a~~b's + $c$'s

There are no other cases since $|vxy| < p$ so
$vxy$ can't contain all three symbols $a, b, c$

Case 3:

(3a) $vxy$ of form $aa^*bb^*$

examples: $w = \underbrace{aaa}_{u}\,\underbrace{a}_{v}\,\underbrace{bbb}_{x}\,\underbrace{\varepsilon}_{y}\,\underbrace{bcccc}_{z}$

$\underbrace{aaa}_{u}\,\underbrace{ab}_{v}\,\underbrace{b}_{x}\,\underbrace{bb}_{y}\,\underbrace{cccc}_{z}$

(3b) $vxy$ of form $bb^*cc^*$

* Since $|vxy| \le p$, $vxy$ can't contain all 3 symbols $a, b, c$

**Example 2** $L = \{ r\#s \mid r, s \in \{0,1\}^* \text{ and } r \text{ is a substring of } s \}$

Let $p$ be pumping length; Let $w = 0^p 1^p \# 0^p 1^p \in L$.

$$\overbrace{\underbrace{0\,1\,1}_{r}\,1\,1\,0}^{s}$$

then $w = uvxyz$, $|vxy| \le p$, $|vy| \ge 1$

Case 1 $v$ contains $\#$. Then $w^2 = uv^2xy^2z$ has 2 $\#$'s so $\notin L$

Case 2 $y$ " $\#$. same argument as Case 1

Case 3 $u$ contains $\#$. then $w^0 = uv^0xy^0z$ has fewer symbols after the $\#$ than before so $w^0 \notin L$

Case 4 $z$ contains $\#$ then $w^2 = uv^2xy^2z$ has more than $2p$ symbols before "$\#$" and $2p$ after "$\#$" so $\notin L$

Case 5 $x$ contains $\#$

Case (5a): $v \neq \varepsilon$; Then $w = \underbrace{0^p 1^j}_{u} \underbrace{1^{p-j}}_{v} \underbrace{1^j \# 0^k}_{x} \underbrace{0^{p-k} 1^p}_{yz}$

then $w^2 = uv^2xy^2z$ has $1^{st}$ block of 1's of length $> p$ but last block of 1's of length $< p$

Case 3: $w = 0^p 1^p \# 0^p 1^p$

$u$ contains '#' symbol

so $w = \underbrace{0^p 1^p \# 0^i}_{u} \underbrace{0^{p-i} 1^j}_{vxy} \underbrace{1^{p-j}}_{z}$

$w^0$ $0^p 1^p \# \underbrace{\quad\quad}_{\text{length} < 2p}$

Case 4: $z$ contains '#'

$$w = \underbrace{0^{i}}_{u}\,\underbrace{0^{p-i}\,1^{p-j}}_{vxy}\,\underbrace{1^{j}\,\#\,0^{p}\,1^{p}}_{z}$$

or

$$w = \underbrace{0^{j}}_{u}\,\underbrace{0^{p-i-j}}_{vxy}\,\underbrace{0^{i}\,1^{p}\,\#\,0^{p}\,1^{p}}_{z}$$

**Example 2** $L = \{ r\#s \mid r, s \in \{0,1\}^* \text{ and } r \text{ is a substring of } s \}$

Let $p$ be pumping length; Let $w = 0^p 1^p \# 0^p 1^p \in L$.

then $w = uvxyz$, $|vxy| \leq p$, $|vy| \geq 1$

Case 1 $v$ contains $\#$. Then $w^2 = uv^2xy^2z$ has 2 $\#$'s so $\notin L$

Case 2 $y$ " $\#$. Same argument as Case 1

Case 3 $u$ contains $\#$. then $w^0 = uv^0xy^0z$ has fewer symbols after the $\#$ than before so $w^0 \notin L$

Case 4 $z$ contains $\#$ then $w^2 = uv^2xy^2z$ has more than $2p$ symbols before "$\#$" and $2p$ after "$\#$" so $\notin L$

Case 5 $x$ contains $\#$

Case (5a): $v \neq \varepsilon$

Case (5b): $v = \varepsilon$. then $y \neq \varepsilon$. So $w^0 = uv^0xy^0z$ has more symbols to left of $\#$ than to right