# Properties of Context Free Languages

Dr. Mousumi Dutt

# CFL Closure Properties

# Closure properties of CFL

- **Closure properties** consider operations on CFL that are guaranteed to produce a CFL
- The CFL’s are closed under *substitution*, *union*, *concatenation*, *closure (star)*, *reversal*, *homomorphism* and *inverse homomorphism*.
- CFL’s are not closed under *intersection* (but the intersection of a CFL and a regular language is always a CFL), *complementation*, and *set-difference*.

# Closure Property Results

- CFLs are closed under:
  - Union
  - Concatenation
  - Kleene closure operator
  - Substitution
  - Homomorphism, inverse homomorphism
  - reversal
- CFLs are *not* closed under:
  - Intersection
  - Difference
  - Complementation

Note: Reg languages are closed under these operators

# Strategy for Closure Property Proofs

- First prove “closure under **substitution**”
- Using the above result, prove other closure properties
- CFLs are closed under:
    - Union
    - Concatenation
    - Kleene closure operator
    - Substitution
    - Homomorphism, inverse homomorphism
    - Reversal

Prove this first

**Note:** s(L) can use a different alphabet

# The ***Substitution*** operation

For each $a \in \Sigma$, then let $s(a)$ be a language

If $w=a_1a_2\ldots a_n \in L$, then:

- $s(w) = \{ x_1x_2 \ldots \} \in s(L)$, s.t., $x_i \in s(a_i)$

**Example:**

- Let $\Sigma=\{0,1\}$
- Let: $s(0) = \{a^nb^n \mid n \geq 1\}$, $s(1) = \{aa,bb\}$
- If $w=01$, $s(w)=s(0).s(1)$
  - E.g., $s(w)$ contains $a^1 b^1 aa$, $a^1 b^1 bb$, $a^2 b^2 aa$, $a^2 b^2 bb$, … and so on.

# CFLs are closed under Substitution

IF L is a CFL and a substitititution defined on L, s(L), is s.t., s(a) is a CFL for every symbol a, THEN:

- s(L) is also a CFL

**What is s(L)?**

$$\underline{L} \qquad\qquad \underline{s(L)}$$

$$\left\{ \begin{array}{c} w_1 \\ w_2 \\ w_3 \\ w_4 \\ \ldots \end{array} \right\} \xrightarrow{s(L)} \left\{ \begin{array}{c} s(w_1) \\ s(w_2) \\ s(w_3) \\ s(w_4) \\ \ldots \end{array} \right\}$$

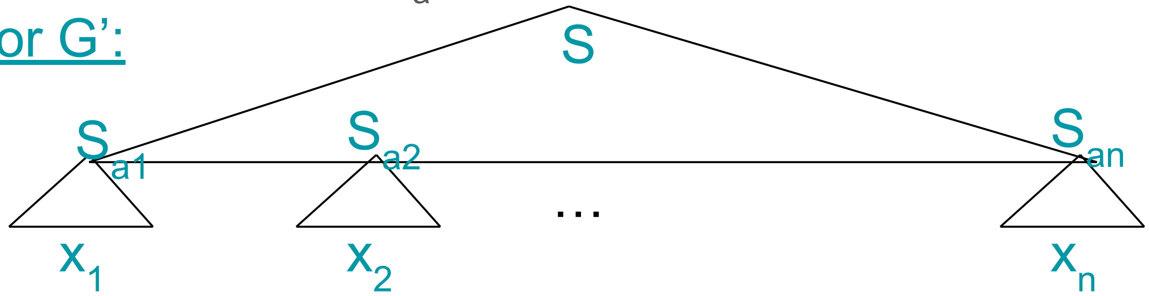
<u>Note:</u> each s(w) is itself a set of strings

# CFLs are closed under *Substitution*

- G=(V,T,P,S) : CFG for L
- Because every s(a) is a CFL, there is a CFG for each s(a)
  - Let $G_a = (V_a,T_a,P_a,S_a)$
- Construct G'=(V',T',P',S) for s(L)

- P' consists of:
  - The productions of P, but with every occurrence of terminal "a" in their bodies replaced by $S_a$.
  - All productions in any $P_a$, for any $a \in \Sigma$

Parse tree for G':



# Substitution

- Each symbol in the strings of one language is replaced by an entire CFL language
- Useful in proving some other closure properties of CFL
- Example: $S(0) = \{a^n b^n \mid n \geq 1\}$, $S(1) = \{aa, bb\}$ is a substitution on alphabet $\Sigma = \{0, 1\}$.

# Substitution

- **Theorem:** If a substitution s assigns a CFL to every symbol in the alphabet of a CFL L, then s(L) is a CFL.
- **Proof**
  - Let $G = (V, \sum, P, S)$ be grammar for L
  - Let $G_a = (V_a, T_a, P_a, S_a)$ be the grammar for each $a \in \sum$ with $V \cap V_a = \varphi$
  - $G' = (V', T', P', S)$ for s(L) where
    - $V' = V \cup V_a$
    - $T'$ = union of $T_a$ for all $a \in \sum$
    - $P'$ consists of
      - All productions in any $P_a$ for $a \in \sum$
      - In productions of P, each terminal a is replaced by $S_a$
- A detailed proof that this construction works is in the reader.
    - Intuition: this replacement allows anystring in La to take the place of any occurrence of a in any string of L.

# Example (1)

- $L = \{0^n1^n \mid n \geq 1\}$, generated by the grammar $S \rightarrow 0S1 \mid 01$,
- $s(0) = \{a^nb^m \mid m \leq n\}$, generated by the grammar $S \rightarrow aSb \mid A$; $A \rightarrow aA \mid ab$,
- $s(1) = \{ab, abc\}$, generated by the grammar $S \rightarrow abA$, $A \rightarrow c \mid \varepsilon$
- Rename second and third S's to $S_0$ and $S_1$ respectively. Rename second A to B. Resulting grammars are:

  $S \rightarrow 0S1 \mid 01$

  $S_0 \rightarrow aS_0b \mid A$; $A \rightarrow aA \mid ab$

  $S_1 \rightarrow abB$; $B \rightarrow c \mid \varepsilon$

# Example(1) Contd...

- In the first grammar replace 0 by $S_0$ and 1 by $S_1$. The combined grammar:

  $G' = (\{S, S_0, S_1, A, B\}, \{a, b\}, P', S)$,

  where $P' = \{S \to S_0SS_1 \mid S_0S_1,\ S_0 \to aS_0b \mid A,\ A \to aA \mid ab,\ S_1 \to abB,\ B \to c \mid \varepsilon\}$

# Application of Substitution

- Closure under union of CFL's $L_1$ and $L_2$
- Closure under concatenation of CFL's $L_1$ and $L_2$
- Closure under Kleene's star (closure * and positive closure $^+$) of CFL's $L_1$
- Closure under homomorphism of CFL $L_i$ for every $a_i \in \Sigma$

# Substitution of a CFL: example

- Let L = language of binary palindromes s.t., substitutions for 0 and 1 are defined as follows:
  - $s(0) = \{a^n b^n \mid n \geq 1\}$, $s(1) = \{xx, yy\}$
- Prove that s(L) is also a CFL.

---

**CFG for L:**

$S \Rightarrow 0S0 \mid 1S1 \mid \varepsilon$

**CFG for s(0):**

$S_0 \Rightarrow aS_0b \mid ab$

**CFG for s(1):**

$S_1 \Rightarrow xx \mid yy$

***Therefore, CFG for s(L):***

$S \Rightarrow S_0SS_0 \mid S_1 S S_1 \mid \varepsilon$
$S_0 \Rightarrow aS_0b \mid ab$
$S_1 \Rightarrow xx \mid yy$

# CFLs are closed under ***union***

Let $L_1$ and $L_2$ be CFLs

<u>To show:</u> $L_2 \cup L_2$ is also a CFL

<u>Let us show by using the result of *Substitution*</u>

- Make a new language:
  - $L_{new} = \{a,b\}$ s.t., $s(a) = L_1$ and $s(b) = L_2$

  $\implies s(L_{new})$ == same as == $L_1 \cup L_2$

- A more direct, alternative proof
  - Let $S_1$ and $S_2$ be the starting variables of the grammars for $L_1$ and $L_2$
    - Then, $S_{new} \Rightarrow S_1 \mid S_2$

# Union

- Use L= {a, b}, s(a) = $L_1$ and s(b)=$L_2$.s(L)= $L_1 \cup L_2$
- To get grammar for $L_1 \cup L_2$ ?
  - Add new start symbol S and rules $S \rightarrow S_1|S_2$
  - We get grammar G = (V, T, P, S) where

  $V = V_1 \cup V_2 \cup \{ S \}$, where $S \notin V_1 \cup V_2$

  $P = P_1 \cup P_2 \cup \{ S \rightarrow S_1 \mid S_2 \}$
- Example:
  - $L_1 = \{ a^n b^n \mid n \geq 0 \}$ , $L_2 = \{ b^n a^n \mid n \geq 0 \}$
  - $G_1 : S_1 \rightarrow aS_1b \mid \varepsilon$, $G_2 : S_2 \rightarrow bS_2a \mid \varepsilon$
  - L1 $\cup$ L2 is G = ({$S_1$, $S_2$ , S}, {a, b}, P, S) where P = {P1 $\cup$ P2 $\cup$ $\{S \rightarrow S_1 \mid S_2 \}$}

# CFLs are closed under ***concatenation***

- Let $L_1$ and $L_2$ be CFLs

Let us show by using the result of *Substitution*

- Make $L_{new}$ = {ab} s.t.,
  $s(a) = L_1$ and $s(b)= L_2$

  $\implies L_1 L_2 = s(L_{new})$

- A proof without using substitution?

# Concatenation

- Let $L=\{ab\}$, $s(a)=L_1$ and $s(b)=L_2$. Then $s(L)=L_1L_2$
- To get grammar for $L_1L_2$ ?
  - Add new start symbol and rule $S \rightarrow S_1S_2$
  - We get $G = (V, T, P, S)$ where

    $V = V_1 \cup V_2 \cup \{ S \}$, where $S \notin V_1 \cup V_2$

    $P = P_1 \cup P_2 \cup \{ S \rightarrow S_1S_2 \}$
- Example:
  - $L_1 = \{ a^nb^n \mid n \geq 0 \}$ with $G_1$: $S_1 \rightarrow aS_1b \mid \varepsilon$
  - $L_2 = \{ b^na^n \mid n \geq 0 \}$ with $G_2$ : $S_2 \rightarrow bS_2a \mid \varepsilon$
  - $L_1L_2 = \{ a^nb^{\{n+m\}}a^m \mid n, m \geq 0 \}$ with $G = (\{S, S_1, S_2\}, \{a, b\}, \{S \rightarrow S_1S_2, S_1 \rightarrow aS_1b \mid \varepsilon, S_2 \rightarrow bS_2a\}, S)$

# CFLs are closed under ***Kleene Closure***

- Let L be a CFL
  - Let $L_{new}$ = {a}* and s(a) = $L_1$
    - Then, L* = s($L_{new}$)

# Kleene's star

- Use $L=\{a\}^*$ or $L=\{a\}^+$, $s(a)=L_1$. Then $s(L)=L_1^*$ (or $s(L)=L_1^+$).
- Example:
  - $L_1 = \{a^n b^n \mid n \geq 0\}$ $(L_1)^* = \{ a^{\{n1\}} b^{\{n1\}} \dots a^{\{nk\}} b^{\{nk\}} \mid k \geq 0 \text{ and } n_i \geq 0 \text{ for all } i \}$
  - $L_2 = \{ a^{\{n2\}} \mid n \geq 1 \}$, $(L_2)^* = a^*$
- To get grammar for $(L_1)^*$
  - Add new start symbol S and rules $S \rightarrow SS_1 \mid \varepsilon$.
  - We get $G = (V, T, P, S)$ where

    $V = V_1 \cup \{ S \}$, where $S \notin V_1$

    $P = P_1 \cup \{ S \rightarrow SS_1 \mid \varepsilon \}$

We won’t use substitution to prove this result

# CFLs are closed under ***Reversal***

- Let L be a CFL, with grammar G=(V,T,P,S)
- For $L^R$, construct $G^R$=(V,T,$P^R$,S) s.t.,
  - If A==> $\alpha$ is in P, then:
    - A==> $\alpha^R$ is in $P^R$
    - (that is, reverse every production)

# Reversal

- The CFL's are closed under reversal
- This means then if L is a CFL, so $L^R$ is a CFL
- It is enough to reverse each production of a CFL for L, i.e., substitute $A \rightarrow \alpha$ by $A \rightarrow \alpha^R$
- Example:
  - $L = \{ a^n b^n \mid n \geq 0 \}$ with $P : S \rightarrow aSb \mid \varepsilon$
  - $L^R = \{ b^n a^n \mid n \geq 0 \}$ with $P^R : S \rightarrow bSa \mid \varepsilon$

# CFLs are *not* closed under Intersection

- <u>Existential proof:</u>
  - $L_1 = \{0^n 1^n 2^i \mid n \geq 1, i \geq 1\}$
  - $L_2 = \{0^i 1^n 2^n \mid n \geq 1, i \geq 1\}$
- Both $L_1$ and $L_2$ are CFLs
  - Grammars?
- But $L_1 \cap L_2$ *cannot* be a CFL
  - Why?
- We have an example, where intersection is not closed.
- Therefore, CFLs are not closed under intersection

# Intersection

- The CFL's are not closed under intersection
- Example:
  - L = $\{0^n 1^n 2^n | n \geq 1\}$ is not context-free.
  - L1 = $\{0^n 1^n 2^i | n \geq 1, i \geq 1\}$, L2 = $\{0^i 1^n 2^n | n \geq 1, i \geq 1\}$ are CFL's with corresponding grammars for L1: S->AB; A->0A1 | 01; B->2B | 2 , and for L2: S ->AB; A->0A | 0; B->1B2 | 12.
  - However, $L = L_1 \cap L_2$
  - Thus intersection of CFL's is not CFL

# Intersection

- Theorem: If L is CFL and R is a regular language, then L ∩ R is a CFL.

# Intersection with RL Proof

- $P=(Q_P, \sum, \Gamma, \delta_P, q_P, Z_0, F_P)$ be PDA to accept CFL by final state
- $A=(Q_A, \sum, \delta_A, q_A, F_A)$ be a DFA for RL
- Construct PDA $P' = (Q, \sum, \Gamma, \delta, q_o, Z_0, F)$ where
  - $Q = Q_p \times Q_A$
  - $q_o = (q_p, q_A)$
  - $F = (F_P \times F_A)$
  - $\delta$ is in the form $\delta((q, p), a, X) = ((r, s), \gamma)$ such that
    1. $s = \delta_A(p, a)$
    2. $(r, \gamma)$ is in $\delta_P(q, a, X)$

# Proof Contd…

- For each move of PDA P, we make the same move in PDA P′ and also we carry along the state of DFA A in a second component of P′.
- P′ accepts a string w iff both P and A accept w.
- w is in L ∩ R.
- The moves $((q_p, q_A), w, Z) \vdash^*_{P'} ((q, p), \varepsilon, \gamma)$ are possible iff $(q_p, w, Z) \vdash^*_{P} (q, \varepsilon, \gamma)$ moves and $p = \delta^*(q_A, w)$ transitions are possible.

# Set Difference with RL

- For a CFL's L, and a regular language R.

  L - R is a CFL.

  Proof:

  - R is regular and $R^C$ is also regular
  - $L - R = L \cap R^C$
  - Complement of of Regular Language is regular
  - Intersection of a CFL and a regular language is CFL

# Set Difference

- $L_1$ and $L_2$ are CFLs. $L_1$ - $L_2$ is not necessarily a CFL

  Proof:

  - L1 = $\sum^*$ - L
  - $\sum^*$ is regular and is also CFL
  - But $\sum^*$ - L = $L^C$
  - If CFLs were closed under set difference, then $\sum^*$ - L = $L^C$ would always be a CFL.
  - But CFL's are not closed under complementation

# CFLs are not closed under difference

- Follows from the fact that CFLs are not closed under complementation

- Because, if CFLs are closed under difference, then:
  - $\overline{L} = \Sigma^* - L$
  - So $\overline{L}$ has to be a CFL too
  - Contradiction

# Complementation

- $L^C$ is not necessarily a CFL
- Proof:
  - Assume that CFLs were closed under complement.
  - If L is a CFL then $L^C$ is a CFL
  - Since CFLs are closed under union, $\mathbf{L_1^C \cup L_2^C}$ is a CFL
  - And by our assumption $\mathbf{(L_1^C \cup L_2^C)^C}$ is a CFL
  - But $\mathbf{(L_1^C \cup L_2^C)^C = L_1 \cap L_2}$ which we just showed isn't necessarily a CFL.
  - Contradiction!

# CFLs are not closed under complementation

- Follows from the fact that CFLs are not closed under intersection

- $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$

<u>Logic:</u> if CFLs were to be closed under complementation

□ the whole right hand side becomes a CFL (because CFL is closed for union)

□ the left hand side (intersection) is also a CFL

□ but we just showed CFLs are NOT closed under intersection!

□ CFLs *<u>cannot</u>* be closed under complementation.

# Homomorphism

- Closure under homomorphism of CFL L for every $a \in \Sigma$
- Suppose L is a CFL over alphabet $\Sigma$ and h is a homomorphism on $\Sigma$.
- Let s be a substitution that replaces every $a \in \Sigma$, by h(a). ie $s(a) = \{h(a)\}$.
- Then $h(L) = s(L)$.
- $h(L) = \{h(a_1)\ldots h(a_k) \mid k \geq 0\}$ where $h(a_k)$ is a homomorphism for every $a_k \in \Sigma$.

# Inverse homomorphism

- To recall: If h is a homomorphism, and L is any language, then $h^{-1}(L)$, called an *inverse homomorphism,* is the set of all strings w such that $h(w) \in L$
- The CFL's are closed under inverse homomorphism.
- Theorem: If L is a CFL and h is a homomorphism, then $h^{-1}(L)$ is a CFL

# Proof Contd…

- After input a is read, h(a) is placed in a buffer.
- Symbols of h(a) are used one at a time and fed to PDA being simulated.
- Only when the buffer is empty does the PDA read another of its input symbol and apply homomorphism to it.

# Proof Contd…

- Suppose h applies to symbols of alphabet Σ and produces strings in T*.
- Let PDA $P = (Q, T, \Gamma, \delta, q_0, Z_0, F)$ that accept CFL L by final state.
- Construct a new PDA $P' = (Q', \Sigma, \Gamma, \delta', (q_0, \varepsilon), Z_0, F \text{ X } \{\varepsilon\})$ to simulate language of $h^{-1}(L)$, where
  - Q′ is the set of pairs (q, x) such that
    - q is a state in Q
    - x is a suffix of some string h(a) for some input string a in Σ

# Proof Contd…

- $\delta'$ is defined by
  - $\delta'((q, \varepsilon), a, X) = \{((q, h(a)), a, X)\}$
  - If $\delta(q, b, X) = \{(p, \gamma)\}$ where $b \in T$ or $b = \varepsilon$ then $\delta'((q, bx), \varepsilon, X) = \{((p, x), \gamma)\}$
- The start state of P' is $(q0, \varepsilon)$
- The accepting state of $P'$ is $(q, \varepsilon)$, where q is an accepting state of P.
- $(q_0,h(w),Z_0) \mid\!-^{*}P\ (p,\varepsilon,\gamma)$ iff $((q_0,\varepsilon),w,Z_0) \mid\!-^{*}P'\ ((p, \varepsilon), \varepsilon, \gamma)$
- P accepts h(w) if and only if $P'$ accepts w, because of the way the accepting states of $P'$ are defined.
- Thus $L(P')=h^{-1}(L(P))$

# Decision Properties

- Emptiness test
  - Generating test
  - Reachability test
- Membership test
  - PDA acceptance

# Pumping Lemma for CFL

$$L_3 = \{\mathsf{a}^n\mathsf{b}^n\mathsf{c}^n \mid n \geqslant 0\}$$

Let's try to design a CFG or PDA

$$S \to \mathsf{a}B\mathsf{c} \mid \varepsilon$$

$$B \to ???$$

read a / push x

read b / pop x

???

# Pumping Lemma for CFL

Suppose we could construct some CFG $G$ for $L_3$

e.g.

$$S \rightarrow CC \mid BC \mid \text{a}$$
$$B \rightarrow CS \mid \text{b}$$
$$C \rightarrow SB \mid \text{c}$$

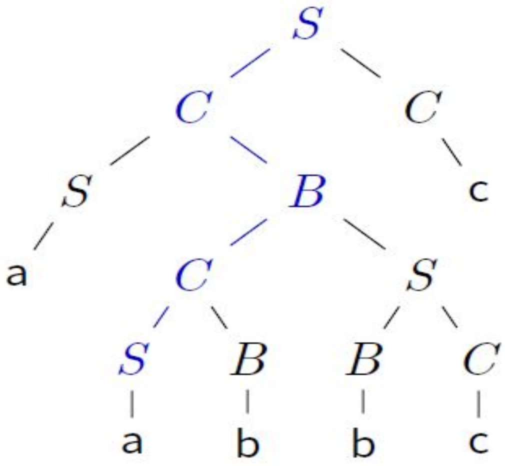
How does a long derivation look like?

$$\begin{aligned}
S &\Rightarrow CC \\
&\Rightarrow SBC \\
&\Rightarrow SCSC \\
&\Rightarrow SSBSC \\
&\Rightarrow SSBBCC \\
&\Rightarrow \text{a}SBBCC \\
&\Rightarrow \text{aa}BBCC \\
&\Rightarrow \text{aab}BCC \\
&\Rightarrow \text{aabb}CC \\
&\Rightarrow \text{aabbc}C \\
&\Rightarrow \text{aabbcc}
\end{aligned}$$

# Repetition in long derivations

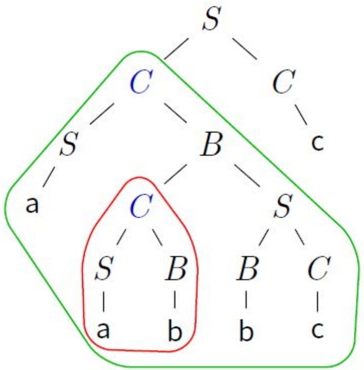
If a derivation is long enough, some variable must appear twice on the same root-to-leave path in a parse tree

$$
\begin{aligned}
S &\Rightarrow CC \\
&\Rightarrow SBC \\
&\Rightarrow SCSC \\
&\Rightarrow SSBSC \\
&\Rightarrow SSBBCC \\
&\Rightarrow \mathsf{a}SBBCC \\
&\Rightarrow \mathsf{aa}BBCC \\
&\Rightarrow \mathsf{aab}BCC \\
&\Rightarrow \mathsf{aabb}CC \\
&\Rightarrow \mathsf{aabbc}C \\
&\Rightarrow \mathsf{aabbcc}
\end{aligned}
$$

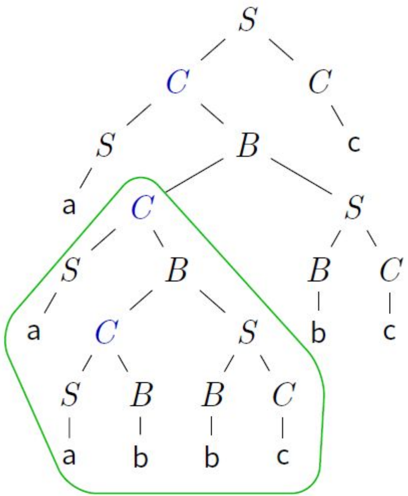


# Pumping Lemma for CFL

Then we can "cut and paste" part of parse tree




aabbcc



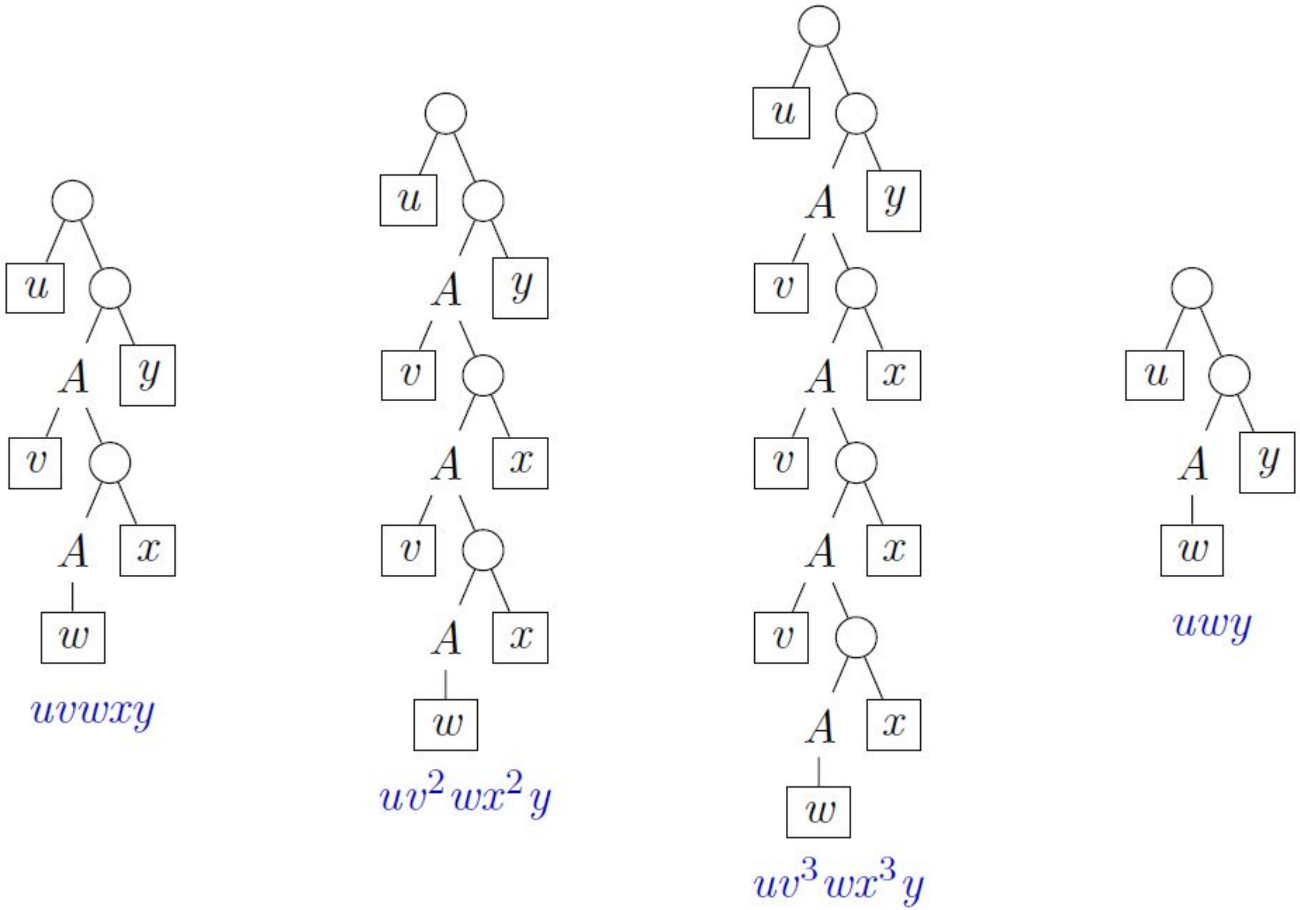

aaabbcbcc

We can repeat this many times

$$\text{aabbcc} \Rightarrow \text{aaabbcbcc} \Rightarrow \text{aaaabbcbcbcc} \Rightarrow \ldots$$
$$\Rightarrow \text{a}(\text{a})^i\text{b}(\text{bc})^i\text{c}$$

Every sufficiently large derivation will have a middle part that can be repeated indefinitely

# Pumping Lemma for CFL



# Pumping Lemma for CFL

$$L_3 = \{\mathsf{a}^n\mathsf{b}^n\mathsf{c}^n \mid n \geqslant 0\}$$

If $L_3$ has a context-free grammar $G$, then for any sufficiently long $s \in L(G)$

$s$ can be split into $s = uvwxy$ such that $L(G)$ also contains $uv^2wx^2y$, $uv^3wx^3y, \ldots$

What happens if $s = \mathsf{a}^m\mathsf{b}^m\mathsf{c}^m$

No matter how it is split, $uv^2wx^2y \notin L_3$

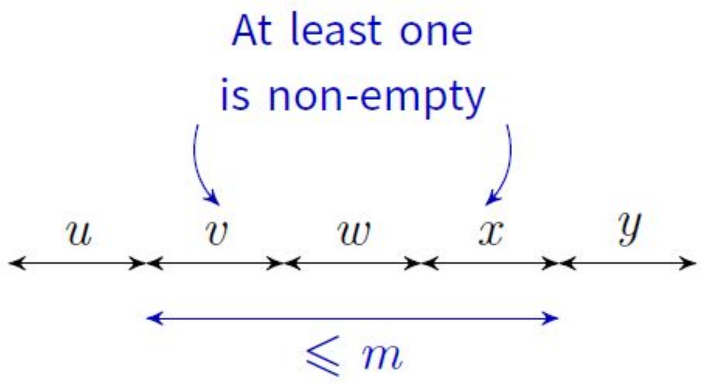
# Pumping Lemma for CFL

For every context-free language $L$

There exists a number $m$ such that for every long string $s$ in $L$ ($|s| \geqslant m$),

we can write $s = uvwxy$ where

1. $|vwx| \leqslant m$
2. $|vx| \geqslant 1$
3. For every $i \geqslant 0$, the string $uv^iwx^iy$ is in $L$



# Pumping Lemma for CFL

To prove $L$ is not context-free, it is enough to show that

For every $m$ there is a long string $s \in L, |s| \geqslant m$, such that for every way of writing $s = uvwxy$ where

1. $|vwx| \leqslant m$
2. $|vx| \geqslant 1$

there is $i \geqslant 0$ such that $uv^i wx^i y$ is not in $L$

# Pumping Lemma for CFL

$$L_3 = \{\mathsf{a}^n\mathsf{b}^n\mathsf{c}^n \mid n \geqslant 0\}$$

1. for every $m$
2. there is $s = \mathsf{a}^m\mathsf{b}^m\mathsf{c}^m$ (at least $m$ symbols)
3. no matter how the pumping lemma splits $s$ into $uvwxy$ ($|vwx| \leqslant m, |vx| \geqslant 1$)
4. $uv^2wx^2y \notin L_3$ (but why?)

# Pumping Lemma for CFL

Case 1: $v$ or $x$ contains two kinds of symbols

$$\text{aa}\underbrace{\text{aabb}}_{v}\text{bbcccc}$$

Then $uv^2wx^2y \notin L_3$ because the pattern is wrong

Case 2: $v$ and $x$ both contain one kind of symbol

$$\text{a}\underbrace{\text{aaa}}_{v}\text{b}\underbrace{\text{bb}}_{x}\text{bbcccc}$$

Then $uv^2wx^2y$ does not have the same number of a's, b's and c's

Conclusion: $uv^2wx^2y \notin L_3$

# Pumping Lemma for CFL

Which is context-free?

$L_1 = \{\mathsf{a}^n\mathsf{b}^n \mid n \geqslant 0\}$ ✓

$L_2 = \{z \mid z \text{ has the same number of a's and b's}\}$ ✓

$L_3 = \{\mathsf{a}^n\mathsf{b}^n\mathsf{c}^n \mid n \geqslant 0\}$ ✗

$L_4 = \{zz^R \mid z \in \{\mathsf{a},\mathsf{b}\}^*\}$ ✓

$L_5 = \{zz \mid z \in \{\mathsf{a},\mathsf{b}\}^*\}$

# Pumping Lemma for CFL

$$L_5 = \{zz \mid z \in \{\mathrm{a}, \mathrm{b}\}^*\}$$

1. for every $m$
2. there is $s = \mathrm{a}^m\mathrm{b}\mathrm{a}^m\mathrm{b}$ (at least $m$ symbols)
3. no matter how the pumping lemma splits $s$ into $uvwxy$ $(|vwx| \leqslant m, |vx| \geqslant 1)$
4. Is $uv^2wx^2y \notin L_5$?

$$\mathrm{aaa}\underbrace{\mathrm{a}}_{v}\mathrm{aba}\underbrace{\mathrm{a}}_{x}\mathrm{aaab}$$

# Pumping Lemma for CFL

$$L_5 = \{zz \mid z \in \{\mathsf{a}, \mathsf{b}\}^*\}$$

1. for every $m$
2. there is $s = \mathsf{a}^m\mathsf{b}^m\mathsf{a}^m\mathsf{b}^m$ (at least $m$ symbols)
3. no matter how the pumping lemma splits $s$ into $uvwxy$ ($|vwx| \leqslant m, |vx| \geqslant 1$)
4. Is $uv^iwx^iy \notin L_5$ for some $i$?

Recall that $|vwx| \leqslant m$

# Pumping Lemma for CFL

Three cases

Case 1 $\quad$ aaa $\underbrace{\text{aabbb}}_{vwx}$ bbaaaaabbbbb

$vwx$ is in the first half of $\mathrm{a}^m\mathrm{b}^m\mathrm{a}^m\mathrm{b}^m$

Case 2 $\quad$ aaaaabb $\underbrace{\text{bbbaa}}_{vwx}$ aaabbbbb

$vwx$ is in the middle part of $\mathrm{a}^m\mathrm{b}^m\mathrm{a}^m\mathrm{b}^m$

Case 3 $\quad$ aaaaabbbbbaaa $\underbrace{\text{aabbb}}_{vwx}$ bb

$vwx$ is in the second half of $\mathrm{a}^m\mathrm{b}^m\mathrm{a}^m\mathrm{b}^m$

# Pumping Lemma for CFL

Apply pumping lemma with $i = 0$

Case 1 $\quad$ aaa $\underbrace{\text{aabbb}}_{vwx}$ bbaaaaabbbbb

$uwy$ becomes $\mathrm{a}^j\mathrm{b}^k\mathrm{a}^m\mathrm{b}^m$, where $j < m$ or $k < m$

Case 2 $\quad$ aaaaabb $\underbrace{\text{bbbaa}}_{vwx}$ aaabbbbb

$uwy$ becomes $\mathrm{a}^m\mathrm{b}^j\mathrm{a}^k\mathrm{b}^m$, where $j < m$ or $k < m$

Case 3 $\quad$ aaaaabbbbbaaa $\underbrace{\text{aabbb}}_{vwx}$ bb

$uwy$ becomes $\mathrm{a}^m\mathrm{b}^m\mathrm{a}^j\mathrm{b}^k$, where $j < m$ or $k < m$

Not of the form $zz$

This covers all cases, so $L_5$ is not context-free

# The Pumping Lemma for CFLs

Let L be a CFL.

Then there exists a constant N, s.t.,

- if $z \in L$ s.t. $|z| \geq N$, then we can write $z = uvwxy$, such that:
  1. $|vwx| \leq N$
  2. $vx \neq \varepsilon$
  3. For all $k \geq 0$: $uv^k wx^k y \in L$

<u>Note:</u> we are pumping in two places (v & x)

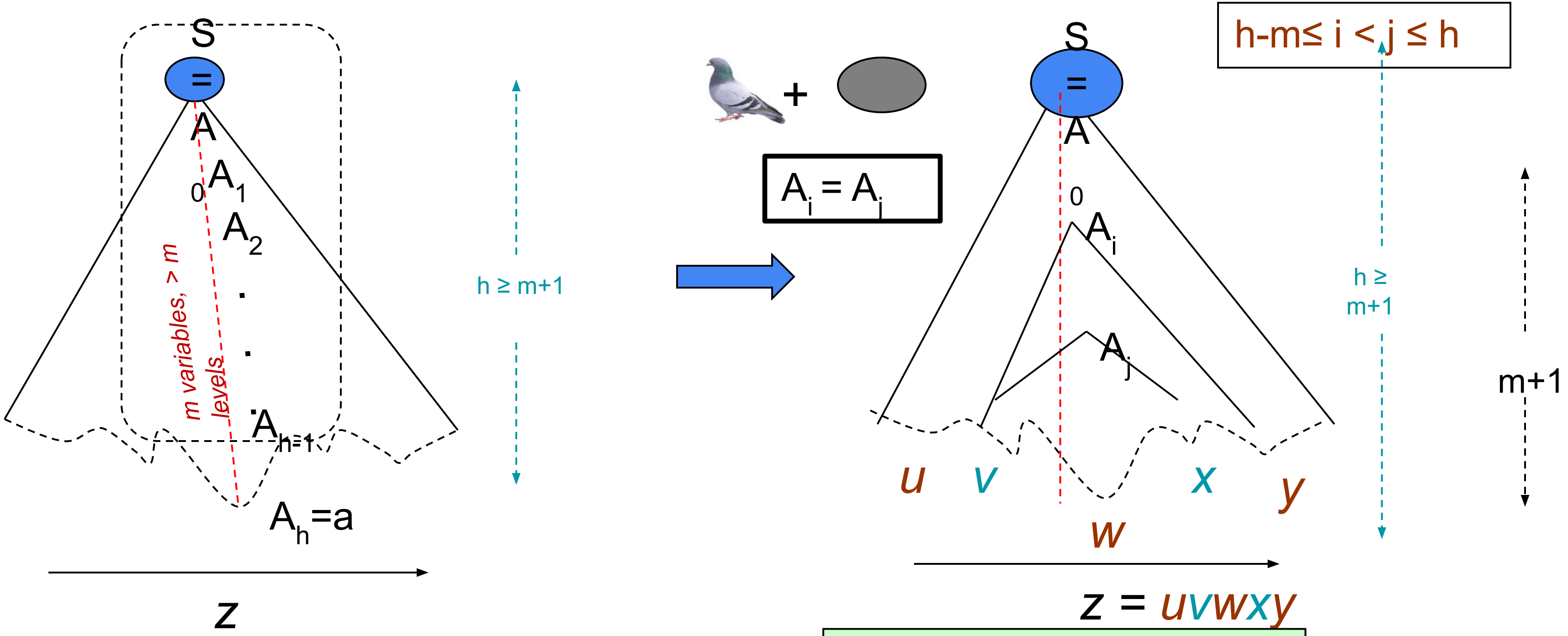
# Proof: Pumping Lemma for CFL

- If L=Φ or contains only ε, then the lemma is trivially satisfied (as it cannot be violated)

- For any other L which is a CFL:
    - Let G be a CNF grammar for L
    - Let m = number of variables in G
    - Choose $N=2^m$.
    - Pick any $z \in L$ s.t. $|z| \geq N$
        - □ the parse tree for z should have a height $\geq m+1$ (by the parse tree theorem)

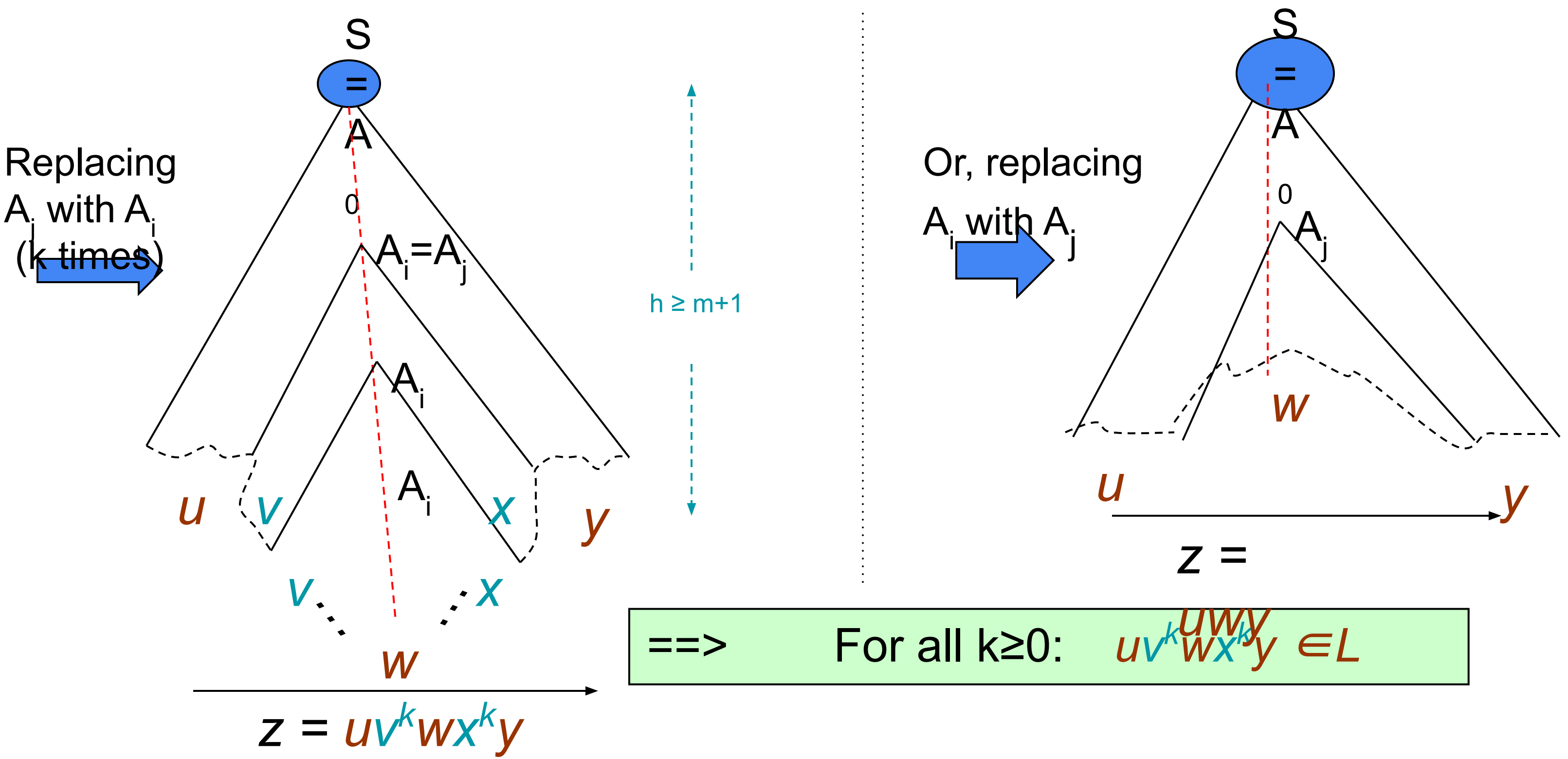
# Parse tree for z

**Meaning:**
Repetition in the last m+1 variables




- Therefore, vx≠ε

# Extending the parse tree…




==> For all k≥0: $uv^kwx^ky \in L$

# Proof contd..

- Also, since $A_i$'s subtree no taller than m+1

  ==> the string generated under $A_i$'s subtree, which is vwx, cannot be longer than $2^m$ (=N)

  But, $2^m$ =N

  ==> $|vwx| \leq N$

  This completes the proof for the pumping lemma.

# Application of Pumping Lemma for CFLs

<u>Example 1:</u> $L = \{a^m b^m c^m \mid m>0\}$

<u>Claim:</u> L is not a CFL

<u>Proof:</u>

- Let N <== P/L constant
- Pick $z = a^N b^N c^N$
- Apply pumping lemma to z and show that there exists at least one other string constructed from z (obtained by pumping up or down) that is $\notin L$

# Proof contd…

- $z = uvwxy$
- As $z = a^N b^N c^N$ *and* $|vwx| \leq N$ *and* $vx \neq \varepsilon$
  - ==> v, x cannot contain all three symbols (a,b,c)
  - ==> we can pump up or pump down to build another string which is $\notin L$

# Example #2 for P/L application

- $L = \{ ww \mid w \text{ is in } \{0,1\}^* \}$
- Show that L is not a CFL
  - Try string $z = 0^N 0^N$
    - what happens?
  - Try string $z = 0^N 1^N 0^N 1^N$
    - what happens?

# Example 3

- L = { $0^{k2}$ | k is any integer)
- Prove L is not a CFL using Pumping Lemma

# Example 4

- $L = \{a^i b^j c^k \mid i<j<k\}$

- Prove that L is not a CFL