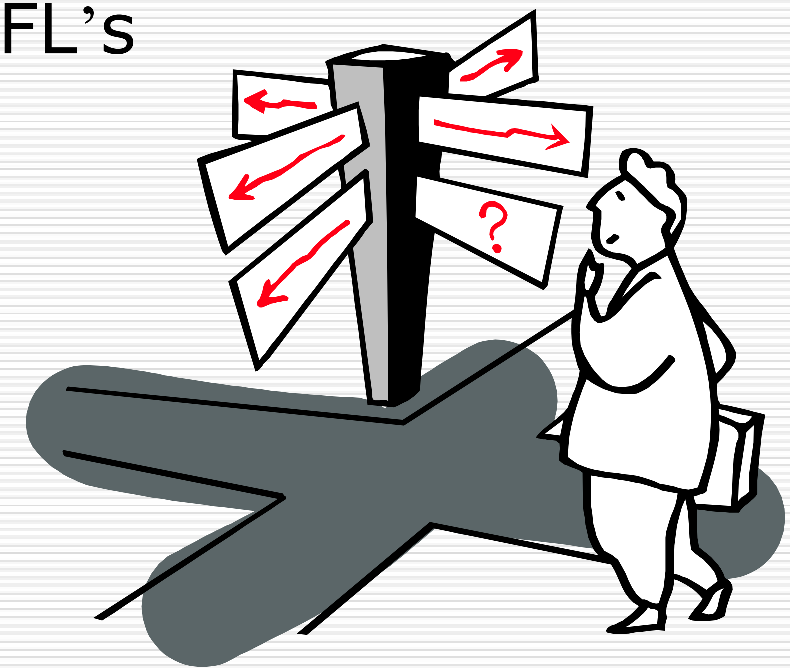
2.3 Properties of Context-free Languages

- Pumping Lemma for CFL's
- Closure Properties
- Decision Properties

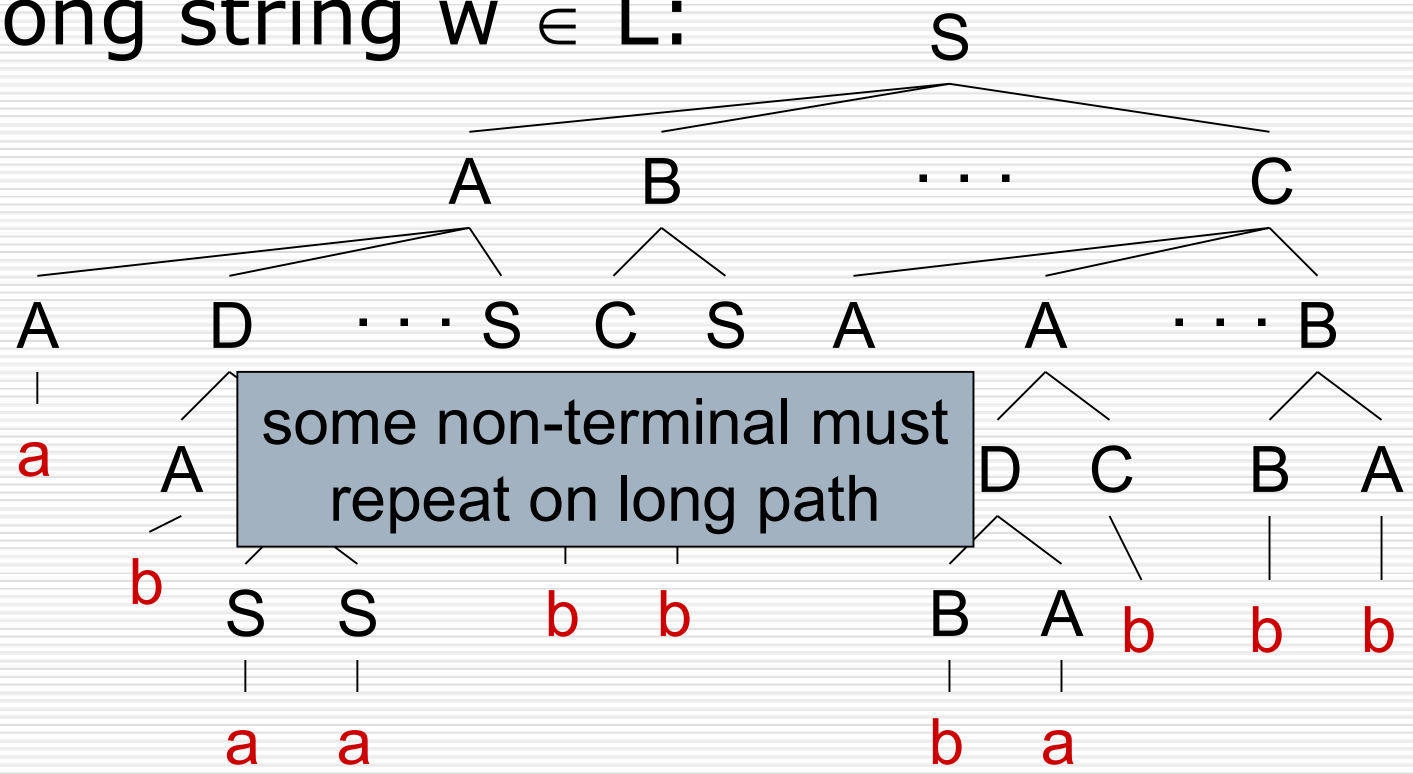


Pumping Lemma for CFL's

- Similar to regular-language PL, but you have to pump two strings in the middle of the string, in tandem (i.e., the same number of copies of each). Formally:
  - $\forall$ CFL L
  - $\exists$ integer p
  - $\forall$ w in L, with $|w| \geq p$
  - $\exists$ uvxyz = w such that $|vxy| \leq p$ and $|vy| > 0$
  - $\forall i \geq 0$, uv^ixy^iz is in L.

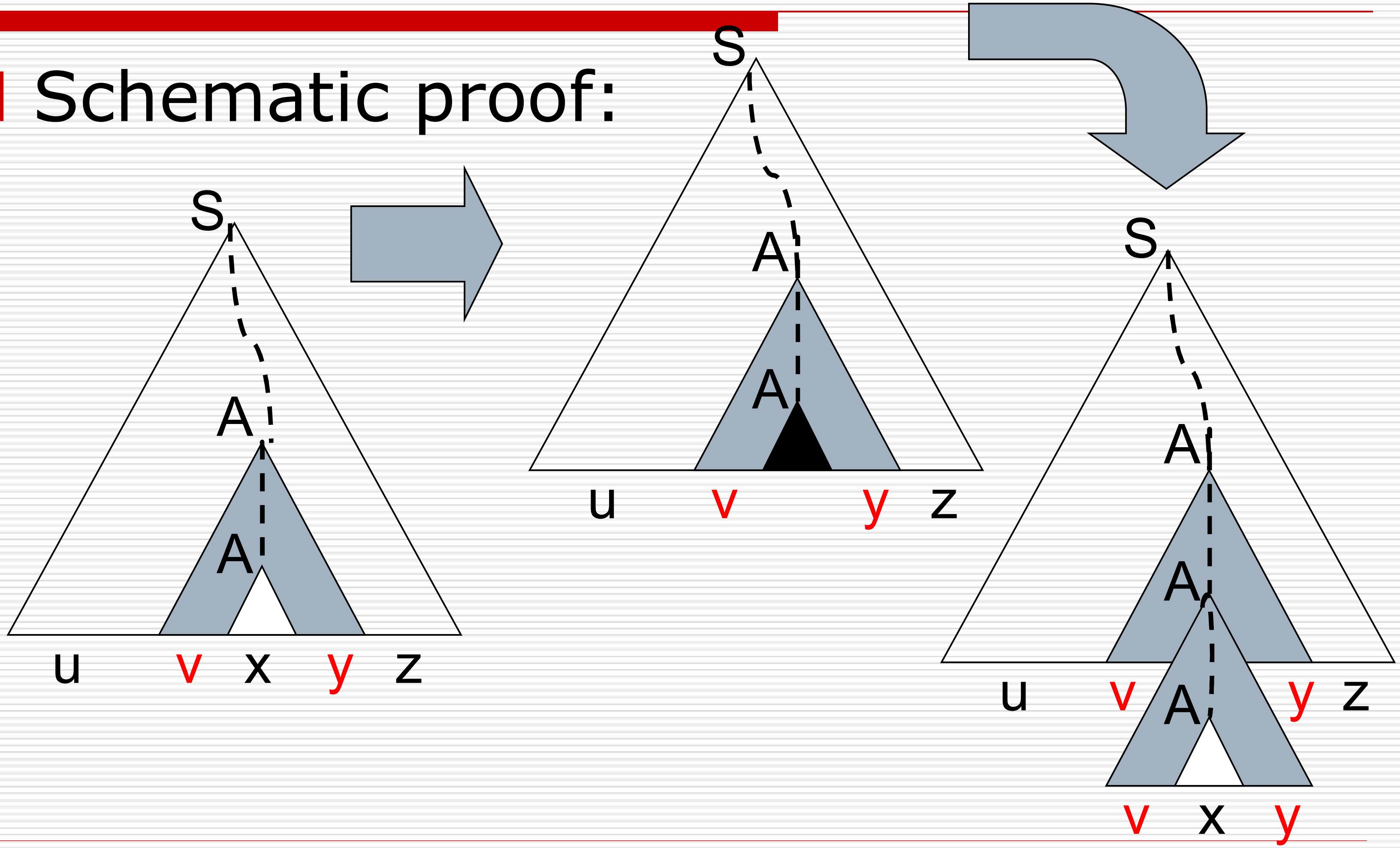
CFL Pumping Lemma

Proof: consider a parse tree for a very long string w ∈ L:



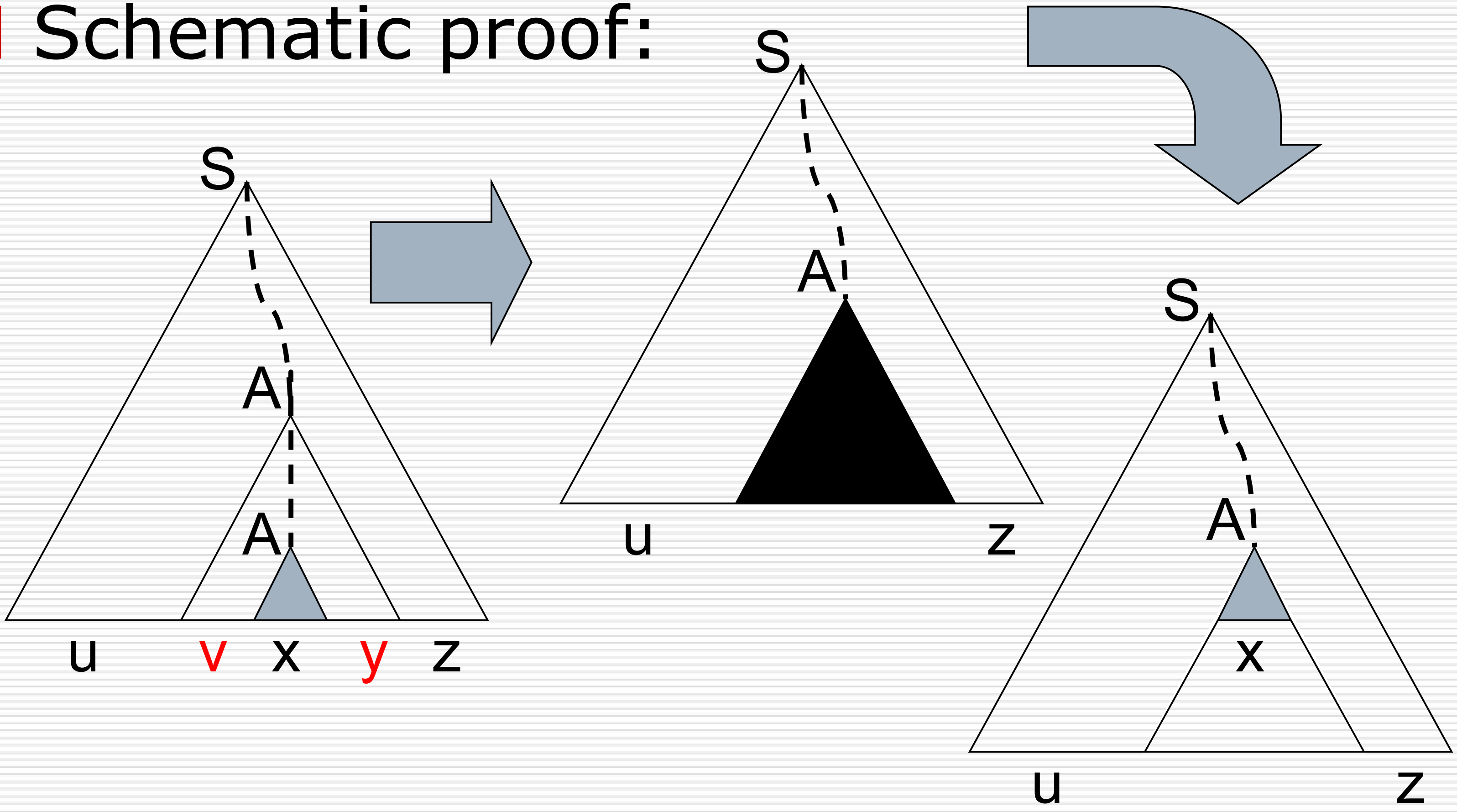
CFL Pumping Lemma

- Schematic proof:



CFL Pumping Lemma

- Schematic proof:



CFL Pumping Lemma

- how large should pumping length p be?
- need to ensure other conditions:

  $|vy| > 0$ $|vxy| \leq p$

- b = max # symbols on rhs of any production (assume $b \geq 2$)
- if parse tree has height $< h$, then string generated has length $< b^h$ (so length $\geq b^h$ implies height $\geq h$)

CFL Pumping Lemma

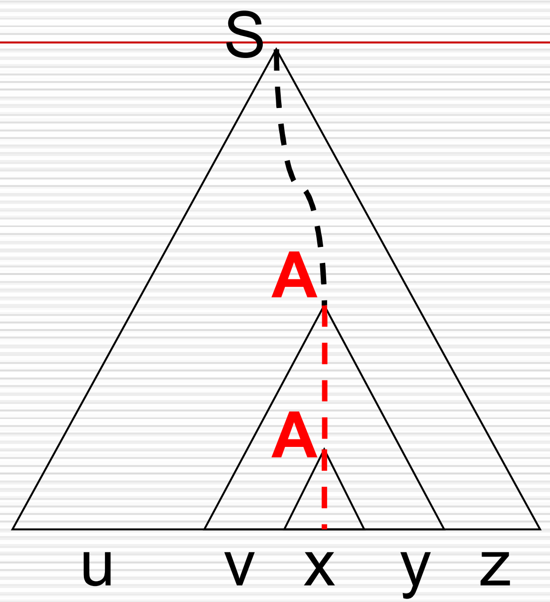
- let m be the # of nonterminals in the grammar
- to ensure path of length at least m+1, require

$$|w| \geq \mathbf{p} = b^{m+1}$$

- since $|w| \geq b^{m+1}$, any parse tree for w has height $\geq m+1$
- let T be the smallest parse tree for w
- longest root-leaf path must consist of at least m+1 non-terminals and 1 terminal.

CFL Pumping Lemma

- there must be a repeated non-terminal **A** on long path
- select a repetition among the lowest m+1 non-terminals on path.
- pictures show that for every i ≥ 0, $uv^ixy^iz \in L$



is $|vy| > 0$?

smallest parse tree T ensures

is $|vxy| \leq p$?

red path has length $\leq m+1$, so $\leq b^{m+1} = p$ leaves

Example 2.20

- $B = \{a^n b^n c^n \mid n \geq 0\}$
- Assume B is a CFL, let p be the pumping length
- Choose $w = a^p b^p c^p$ in B ($|w| > p$)
- Applying PL, $w = uvxyz$, where $|vy|>0$ and $|vxy| \leq p$, such that uv^ixy^iz in B for all $i \geq 0$
- Two possible cases:
  - $vxy = a^*b^*$, uv^2xy^2z will result in more a's and/or more b's than c's, not in B
  - $vxy = b^*c^*$, uv^2xy^2z will result in more b's and/or more c's than a's, not in B
- Contradiction, B is not a CFL

Example 2.21

- $C = \{a^i b^j c^k \mid k \geq j \geq i \geq 0\}$
- Assume C is a CFL, let p be the pumping length
- Choose $w = a^p b^p c^p$ in C ($|w| > p$)
- Applying PL, $w = uvxyz$, where $|vy|>0$ and $|vxy| \leq p$, such that uv^ixy^iz in C for all $i \geq 0$
- Two possible cases:
  - $vxy = a^*b^*$, uv^2xy^2z will result in more a's and/or more b's than c's, not in C
  - $vxy = b^*c^*$, $uv^0xy^0z = uxz$ will result in fewer b's and/or fewer c's than a's, not in C
- Contradiction, C is not a CFL

Example 2.22

- $D = \{ ww \mid w \in \{0, 1\}^* \}$
- Assume D is a CFL, let p be the pumping length
- Choose $w = 0^p1^p0^p1^p$ in D ($|w| > p$)
  - Why not choose $w = 0^p10^p1$?
- Applying PL, $w = uvxyz$, where $|vy|>0$ and $|vxy|\leq p$, such that uv^ixy^iz in D for all $i\geq 0$

Example 2.22 (Cont'd)

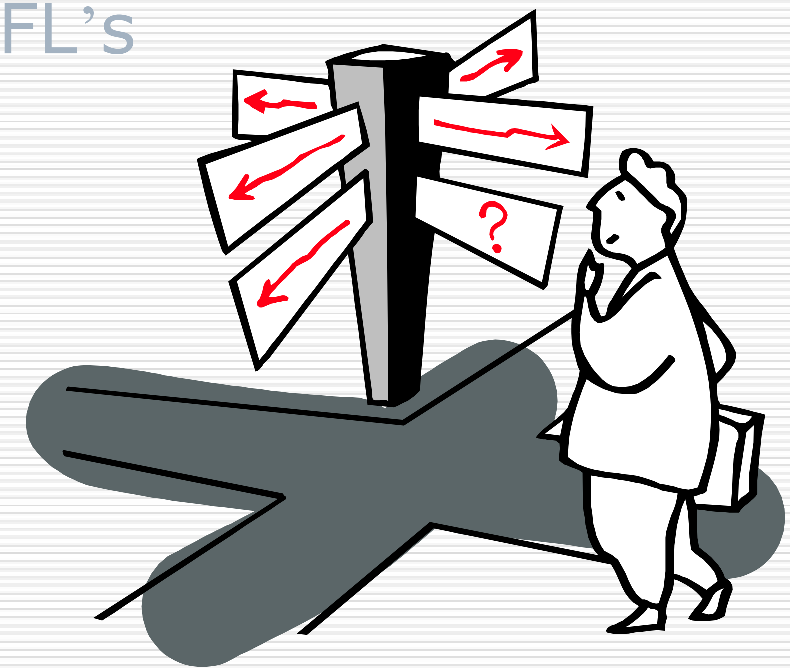
- Three possible cases:
  - vxy in first half
    - then uv^2xy^2z = 0??...?1??...?, not in D
  - vxy in second half
    - then uv^2xy^2z = ??...?0??...?1, not in D
  - vxy straddles midpoint
    - then $uv^0xy^0z = uxz = 0^p1^i0^j1^p$ with $i \neq p$ or $j \neq p$, not in D
- Contradiction, D is not a CFL

Exercise

- Prove that L = $\{0^{k^2} \mid k \text{ is any integer}\}$ is not a CFL

2.3 Properties of Context-free Languages

- Pumping Lemma for CFL's
- Closure Properties
- Decision Properties



Closure Under Substitution

- If a substitution s assigns a CFL to every symbol in the alphabet of a CFL L, then s(L) is a CFL.
  - Take a grammar for L and a grammar for each language $L_a = s(a)$.
  - Make sure all the variables of all these grammars are different (rename variables whenever necessary).
  - Replace each terminal a in the productions for L by S_a, the start symbol of the grammar for L_a.
  - Intuition: this replacement allows any string in L_a to take the place of any occurrence of a in any string of L.

Example

- $L = \{0^n 1^n \mid n \geq 1\}$, generated by CFG $S \to 0S1 \mid 01$.
- $s(0) = \{a^n b^m \mid m \leq n\}$, generated by CFG $S \to aSb \mid A$; $A \to aA \mid ab$.
- $s(1) = \{ab, abc\}$, generated by CFG $S \to abA$; $A \to c \mid \varepsilon$.

1. Rename second and third S's to S_0 and S_1, respectively. Rename second A to B. Resulting grammars are:
   - $S_0 \to aS_0b \mid A$; $A \to aA \mid ab$
   - $S_1 \to abB$; $B \to c \mid \varepsilon$
2. In the first grammar, replace 0 by S_0 and 1 by S_1. The combined grammar:
   - $S \to S_0SS_1 \mid S_0S_1$
   - $S_0 \to aS_0b \mid A$; $A \to aA \mid ab$
   - $S_1 \to abB$; $B \to c \mid \varepsilon$

Consequences of Closure Under Substitution

- Closure of CFL's under union, concatenation, star, homomorphism.
  - union: let L_1 and L_2 be CFL's, L = {1, 2}, $s(1)=L_1$, $s(2)=L_2$. Then $L_1 \cup L_2 = s(L)$.
  - concatenation: let L = {12}. Then $L_1L_2 = s(L)$.
  - star: let L = {1}*. Then $L_1^* = s(L)$.
  - homomorphism: let L be a CFL over Σ, h be a homomorphism over Σ. Define s by s(a) = {h(a)}. Then h(L) = s(L).

Closure Under Reversal

- If $L \in CFL$, then $L^R \in CFL$.

- PROOF: Suppose L is generated by CFG $G = (V, T, P, S)$, construct $G^R = (V, T, P^R, S)$, where

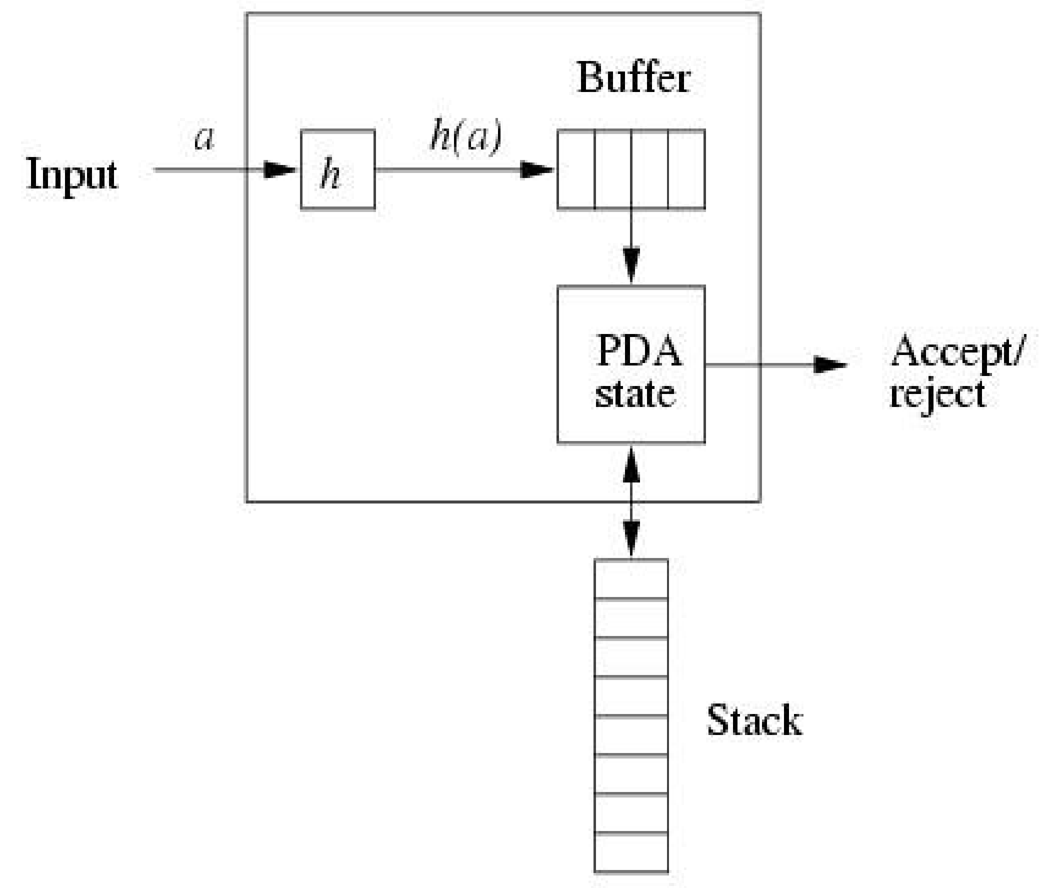
$$P^R = \{A \rightarrow \alpha^R \mid A \rightarrow \alpha \text{ in } P\}$$

  Prove that $L^R = L(G^R)$ by induction on the lengths of derivations

Closure Under Inverse Homom.

- PDA-based construction.
  - Keep a “buffer” in which we place h(a) for some input symbol a.
  - Read inputs from the front of the buffer (ε OK).
  - When the buffer is empty, it may be reloaded with h(b) for the next input symbol b, or we may continue making ε-moves.

Closure Under Inverse Homom.



Formal Construction of PDA for $h^{-1}(L)$

Let $L = L(P)$ for some PDA P, construct some PDA P' to accept $h^{-1}(L)$

- States are pairs $[q, w]$, where:
  - q is a state of P
  - w is a suffix of $h(a)$ for some symbol a
    - Thus, only a finite number of possible values for w
- Stack symbols of P' are those of P
- Start state of P' is $[q_0, \varepsilon]$

Formal Construction of PDA for $h^{-1}(L)$

- Final states of P′ are the states [q, ε] such that q is a final state of P
- δ′([q, ε], a, X) = {([q, h(a)], X)} for any input symbol a of P′ and any stack symbol X.
- δ′([q, bw], ε, X) contains ([p,w], α) if δ(q, b, X) contains (p, α), where b is either an input symbol of P or ε.
  - Simulate P from the buffer

Proving $L(P') = h^{-1}(L(P))$

- Key argument:
  - P' makes the transition $([q_0, \varepsilon], w, Z_0) \vdash^* ([p, \varepsilon], \varepsilon, \gamma)$ if and only if P makes the transition $(q_0, h(w), Z_0) \vdash^* (p, \varepsilon, \gamma)$
- Proof in both directions is an induction on the number of moves made.

Non-closure Under Intersection

- A counter example: $L=\{0^n1^n2^n \mid n\geq 1\}$ is not a CFL. But
  - $L_1=\{0^n1^n2^i \mid n\geq 1, i\geq 1\}$ is a CFL, can be generated by grammar:

    $S\rightarrow AB$

    $A\rightarrow 0A1 \mid 01$

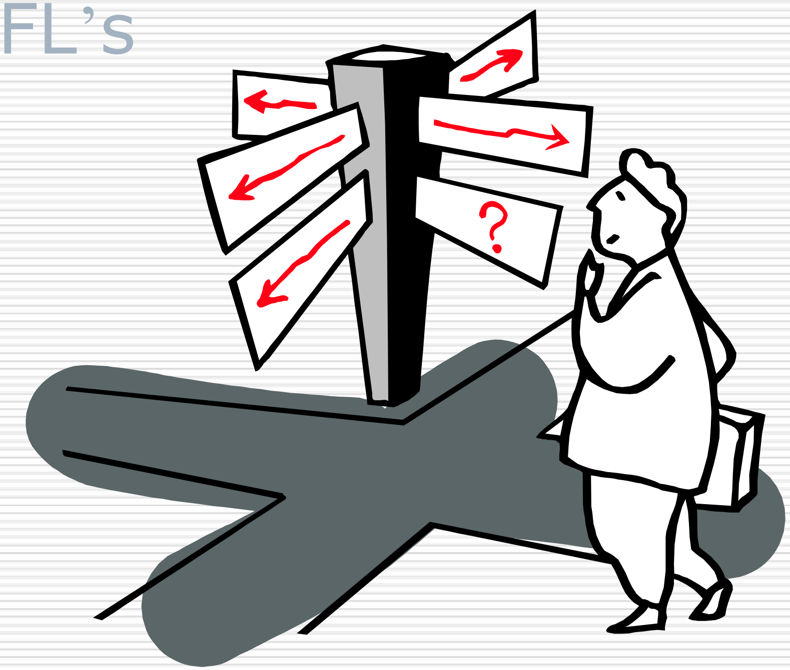
    $B\rightarrow 2B \mid 2$
  - Likewise $L_2=\{0^i1^n2^n \mid n\geq 1, i\geq 1\}$ is a CFL.
  - $L = L_1\cap L_2$

Non-closure Under Complement

- Since CFL's are closed under union, if they were also closed under complement, they would be closed under intersection by DeMorgan's law.

2.3 Properties of Context-free Languages

- Pumping Lemma for CFL's
- Closure Properties
- Decision Properties



Decision Properties of CFL's

- Testing emptiness of a CFL
- Testing finiteness of a CFL
- Testing membership of a string in a CFL
- Preview of undecidable CFL problems

Testing Emptiness

- Use a CFG representation of the given CFL, check if the start symbol is useless.

Testing Finiteness

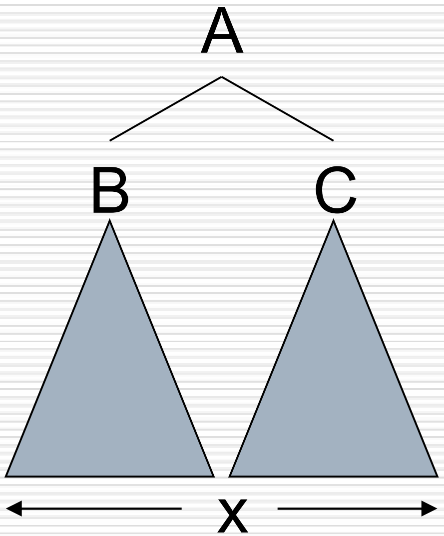
- Let L be a CFL. Then there is some pumping-lemma constant **n** for L.
- Test all strings of length between **n** and **2n-1** for membership (as in next section).
- If there is any such string, it can be pumped, and the language is infinite.
- If there is no such string, then n-1 is an upper limit on the length of strings, so the language is finite.
  - Trick: If there were a string z = uvxyz of length 2n or longer, you can find a shorter string uxz in L, but it's at most n shorter. Thus, if there are any strings of length 2n or more, you can repeatedly cut out vy to get, eventually, a string whose length is in the range n to 2n-1.

Testing Membership

- Can simulate NPDA, but this takes exponential time in the worst case.
- There is an $O(n^3)$ algorithm (n = length of w) that uses a “dynamic programming” technique.
  - Called Cocke-Younger-Kasami (CYK) algorithm.

Testing Membership

- Convert CFG into Chomsky Normal form.
- Parse tree for string x generated by nonterminal A:



If $A \Rightarrow^k x$ $(k > 1)$ then there must be a way to split x:

$$x = yz$$

- $A \rightarrow BC$ is a production and
- $B \Rightarrow^i y$ and $C \Rightarrow^j z$ for $i, j < k$

CYK Algorithm

- Input is $w = a_1a_2\ldots a_n$
- We construct a triangular table, where $X_{ij} = \{A \mid A \Rightarrow^* a_ia_{i+1}\ldots a_j\}$
- See if $S \in X_{1n}$, i.e. $S \Rightarrow^* w$

X_{15}				
X_{14}	X_{25}			
X_{13}	X_{24}	X_{35}		
X_{12}	X_{23}	X_{34}	X_{45}	
X_{11}	X_{22}	X_{33}	X_{44}	X_{55}
a_1	a_2	a_3	a_4	a_5

CYK Algorithm

- Fill the table row-by-row, upwards.
- Compute X_{ij} in the induction
  - Basis: $X_{ii} = \{A \mid A \rightarrow a_i \text{ is in } G\}$
  - Induction: $A \in X_{ij}$ if there exists $A \rightarrow BC$ in G, and $B \in X_{ik}$, $C \in X_{(k+1)j}$ for some k ($i<=k<j$).
- For $w = a_1a_2 \ldots a_n$, there are $O(n^2)$ entries X_{ij} to compute. For each X_{ij} we need to compare at most n pairs $(X_{ik}, X_{(k+1)j})$. Total work is $O(n^3)$.

Example

$S \rightarrow AS \mid SB \mid AB$

$A \rightarrow a$

$B \rightarrow b$

and the input string is *aabb*.

Preview of Undecidable CFL Problems

- Is a given CFG ambiguous?
- Is a given CFL inherently ambiguous?
- Is the intersection of two CFL's empty?
- Are two CFL's the same?
- Is a given CFL universal (equal to Σ^*)?

Summary of Chap. 2

- Context-Free Grammars (CFG's) describe Context-Free Languages (CFL's)
  - formal definition
    - terminals, non-terminals
    - productions
  - derivations
  - parse trees, yields
  - ambiguity
  - Chomsky Normal Form (CNF)
- Pushdown Automata (PDA)

Summary of Chap. 2

- NPDA's and CFG's are equivalent
- Deterministic PDA's are weaker
- CFL Pumping Lemma is used to show certain languages are not CFL's
- Some of the closure properties of regular languages carry over to CFL's. But CFL's are not closed under intersection nor complement.
- We can test for emptiness, finiteness and membership. But for instance, equivalence of CFL's is undecidable.

So far…

- We proved (using constructions of FA's and PDA's and the two pumping lemmas):