Intro to Theory of Computation

CS
464

LECTURE 8

Last time:

- Context-free grammars (CFG)
- Equivalence of CFGs and PDAs

Today:

- Converting a PDA to a CFG
- Pumping Lemma for CFLs

Sofya Raskhodnikova

A language is generated by a CFG

$\Rightarrow$

It is recognized by a PDA

Converting a PDA to a CFG

Given PDA P = (Q, Σ, Γ, δ, q, F)

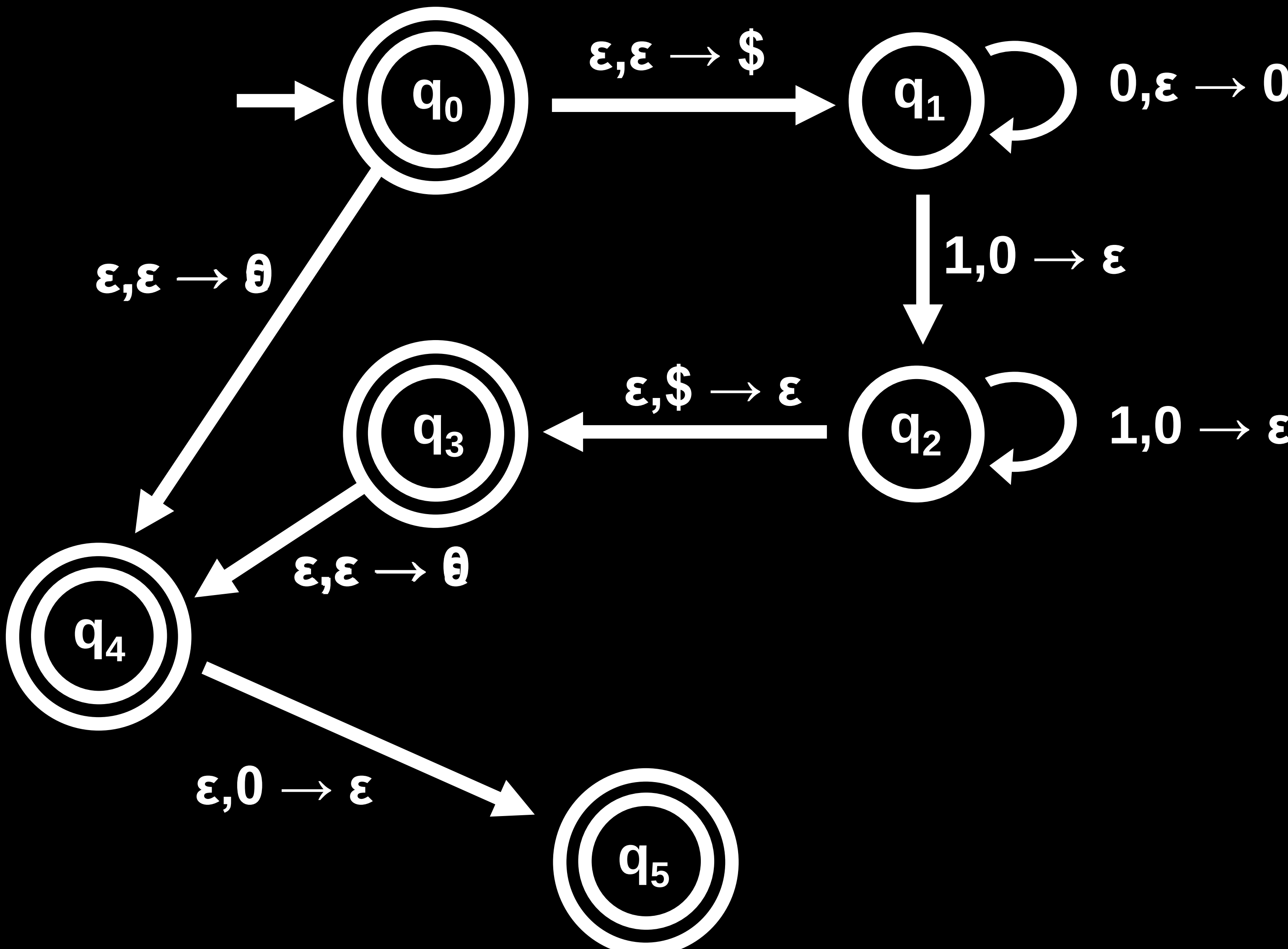
Construct a CFG G = (V, Σ, R, S) with L(G)=L(P)

First, simplify P so that:

1. **It has a single accept state, q_{accept}**
2. **It empties the stack before accepting**
3. **Each transition does exactly one of:**

- **pushes a symbol;**
- **pops a symbol.**

SIMPLIFY



From PDA to CFG: main idea

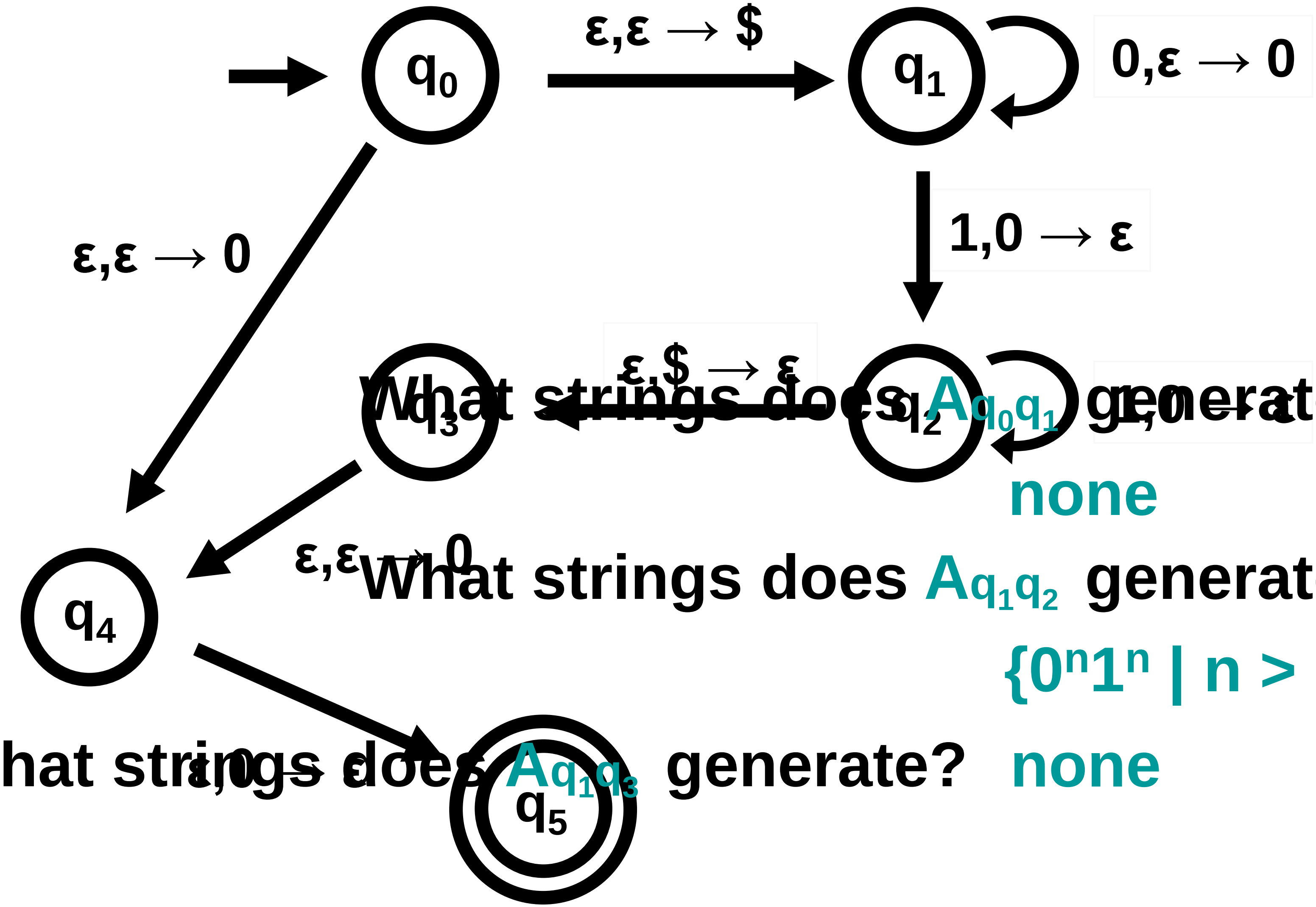
For each pair of states p and q in P, add a variable A_{pq} to CFG that generates all strings that that can take P from p to q without changing the stack*

$$V = \{A_{pq} \mid p,q \in Q\}$$

$$S = A_{q_0 q_{accept}}$$

*Starting from any stack S in p, including empty stack, P has stack S at q.

Example



What strings does $A_{q_0q_1}$ generate?

none

What strings does $A_{q_1q_2}$ generate?

$\{0^n1^n \mid n > 0\}$

What strings does $A_{q_1q_3}$ generate? none

From PDA to CFG: main idea

A_{pq} generates all strings that take P from p to q without changing the stack

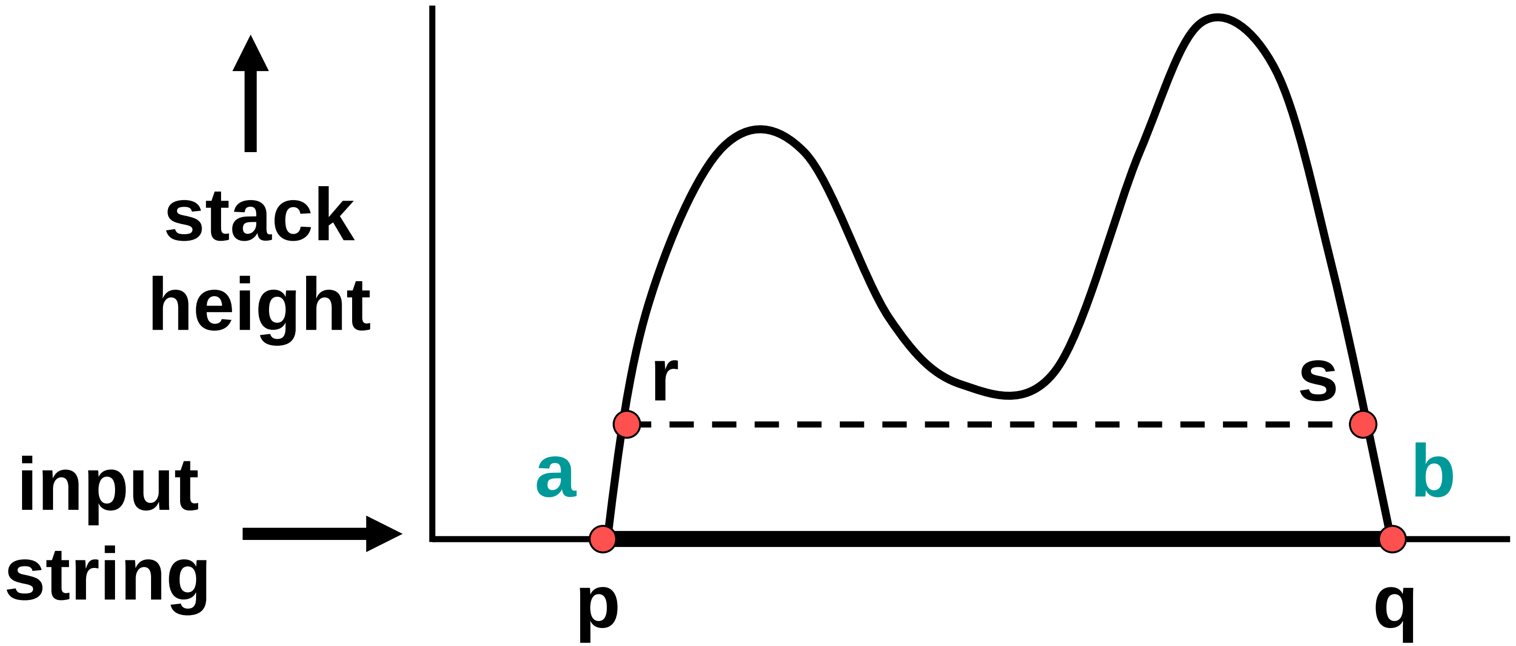
Let x be such a string

- P's **first** move on x must be a **push**
- P's **last** move on x must be a **pop**

Consider the stack while reading x. Either:

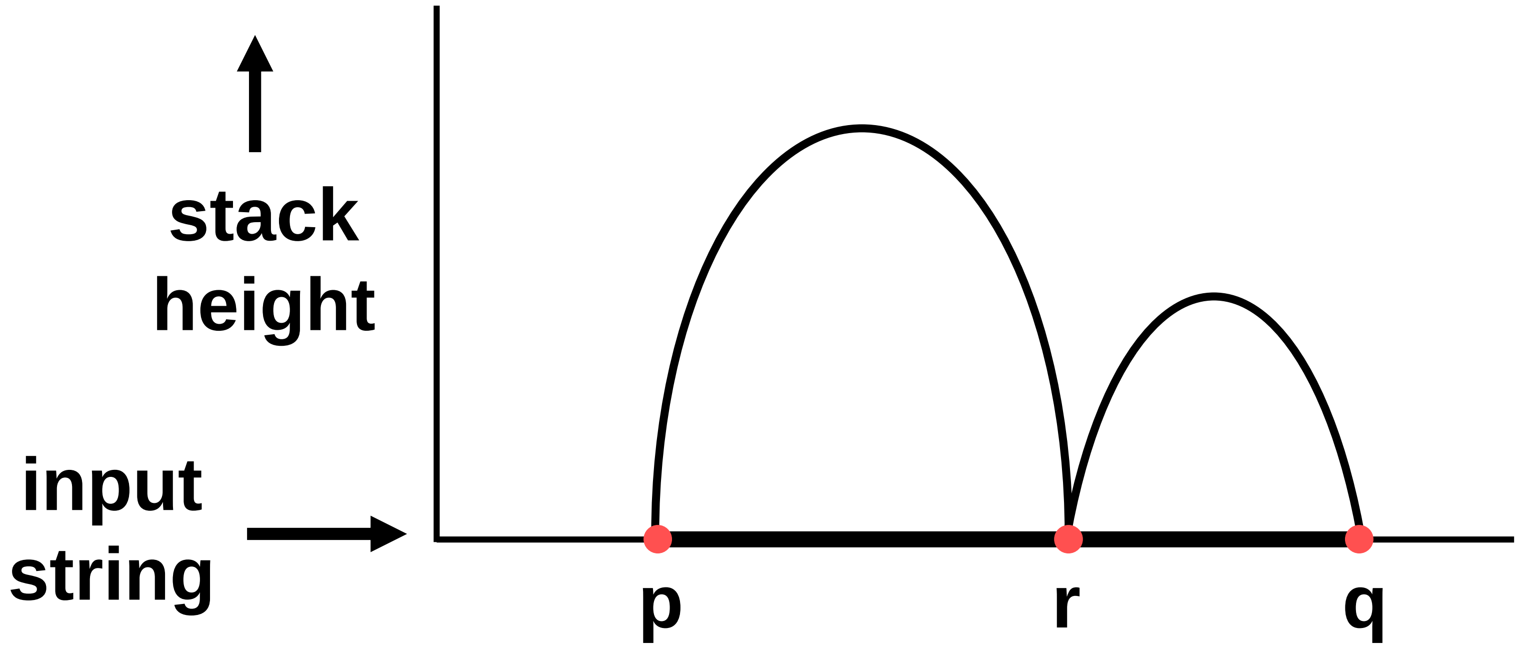
1. New portion of the stack first empties **only at the end of x**
2. New portion empties **before the end of x**

1. New portion of the stack first empties only at the end of x



$$A_{pq} \rightarrow aA_{rs}b$$

2. New portion empties before the end of x



$$A_{pq} \rightarrow A_{pr}A_{rq}$$

CFG construction

$V = \{A_{pq} \mid p,q \in Q\}$

$S = A_{q_0 q_{accept}}$

For each $p,q,r,s \in Q$, $t \in \Gamma$ and $a,b \in \Sigma_\varepsilon$

If $(r,t) \in \delta(p,a,\varepsilon)$ and $(q,\varepsilon) \in \delta(s,b,t)$

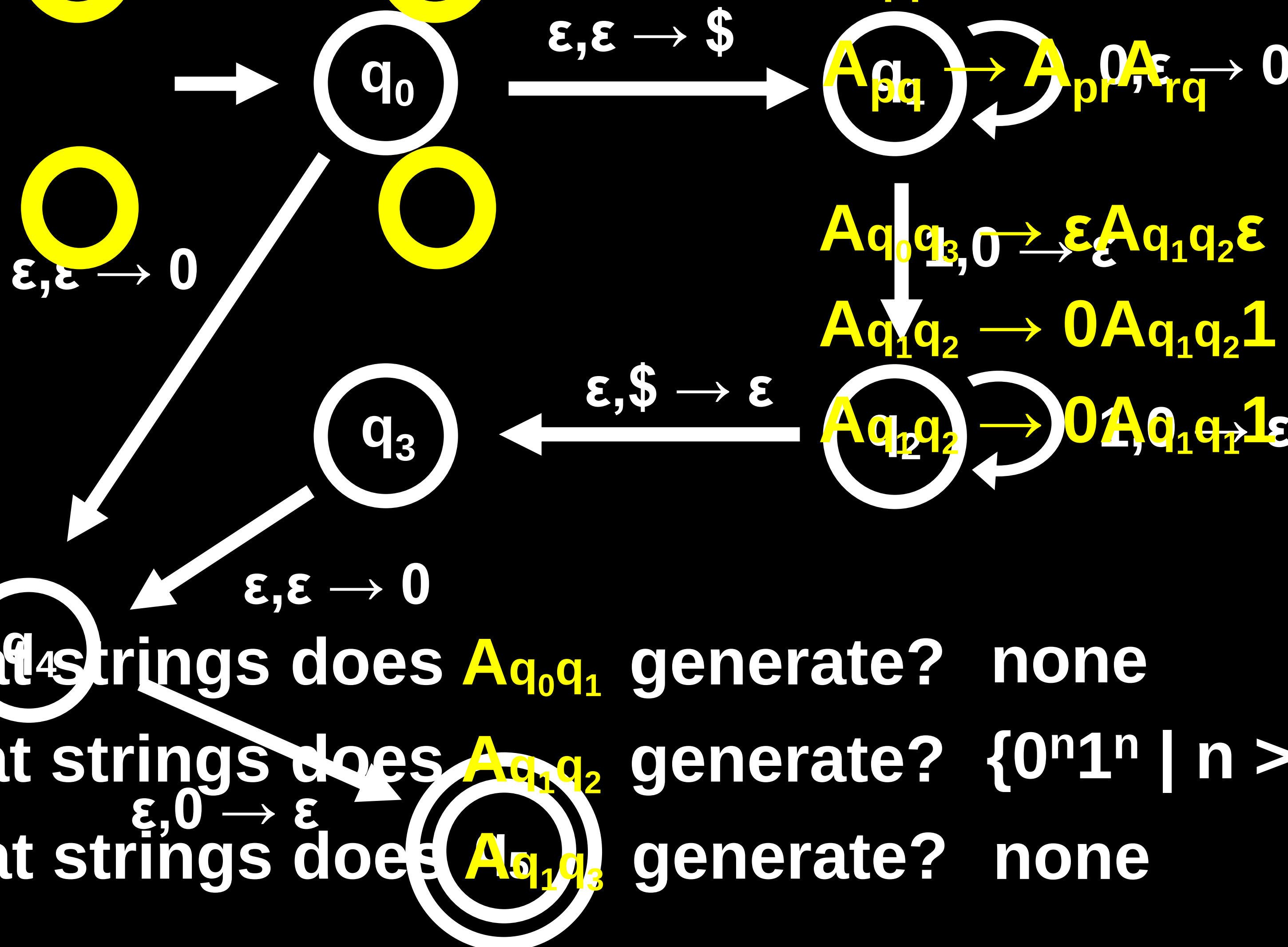
Then add the rule $A_{pq} \rightarrow aA_{rs}b$

For each $p,q,r \in Q$,

add the rule $A_{pq} \rightarrow A_{pr}A_{rq}$

For each $p \in Q$,

add the rule $A_{pp} \rightarrow \varepsilon$



$A_{qq} \to \varepsilon$

$A_{pq} \to A_{pr}A_{rq}$

$A_{q_0q_3} \to \varepsilon A_{q_1q_2}\varepsilon$

$A_{q_1q_2} \to 0A_{q_1q_2}1$

$A_{q_1q_2} \to 0A_{q_1q_1}1$

What strings does $A_{q_0q_1}$ generate? none

What strings does $A_{q_1q_2}$ generate? $\{0^n1^n \mid n > 0\}$

What strings does $A_{q_1q_3}$ generate? none

CS 464

Equivalence of CFGs & PDAs

A language is generated by a CFG

$\Leftrightarrow$

It is recognized by a PDA

Read on your own

1. Ambiguous CFGs.
2. Chomsky normal form for CFGs
3. (Skipping Chapter 2.4 in Sipser).

CS 464

Context-free or not?

NOT $L_1 = \{ xy \mid x,y \in \{0,1\}^* \text{ and } x=y\}$

YES $L_2 = \{xy \mid x,y \in \{0,1\}^*, |x|=|y| \text{ and } x \neq y\}$

Pumping lemma for CFLs

Let L be a context-free language

Then there exists P such that
for every $w \in L$ with $|w| \geq P$

there exist $uvxyz=w$, where:

1. $|vy| > 0$
2. $|vxy| \leq P$
3. $uv^ixy^iz \in L$ for all $i \geq 0$

Examples

there exist uvxyz=w, where:

1. $|vy| \geq 1$
2. $|vxy| \leq P$
3. $uv^ixy^iz \in L$ for all $i \geq 0$

Example: $L = \{ w \in \{0,1\}^* \mid w = w^R \}$.
w = 0; u,v,x,y,z = ?
w = 010; u,v,x,y,z=?

Example: $L = \{ w \in \{a,b\}^* \mid \#a > \#b \text{ in } w \}$.
w = a; u,v,x,y,z = ?
w = aab; u,v,x,y,z=?

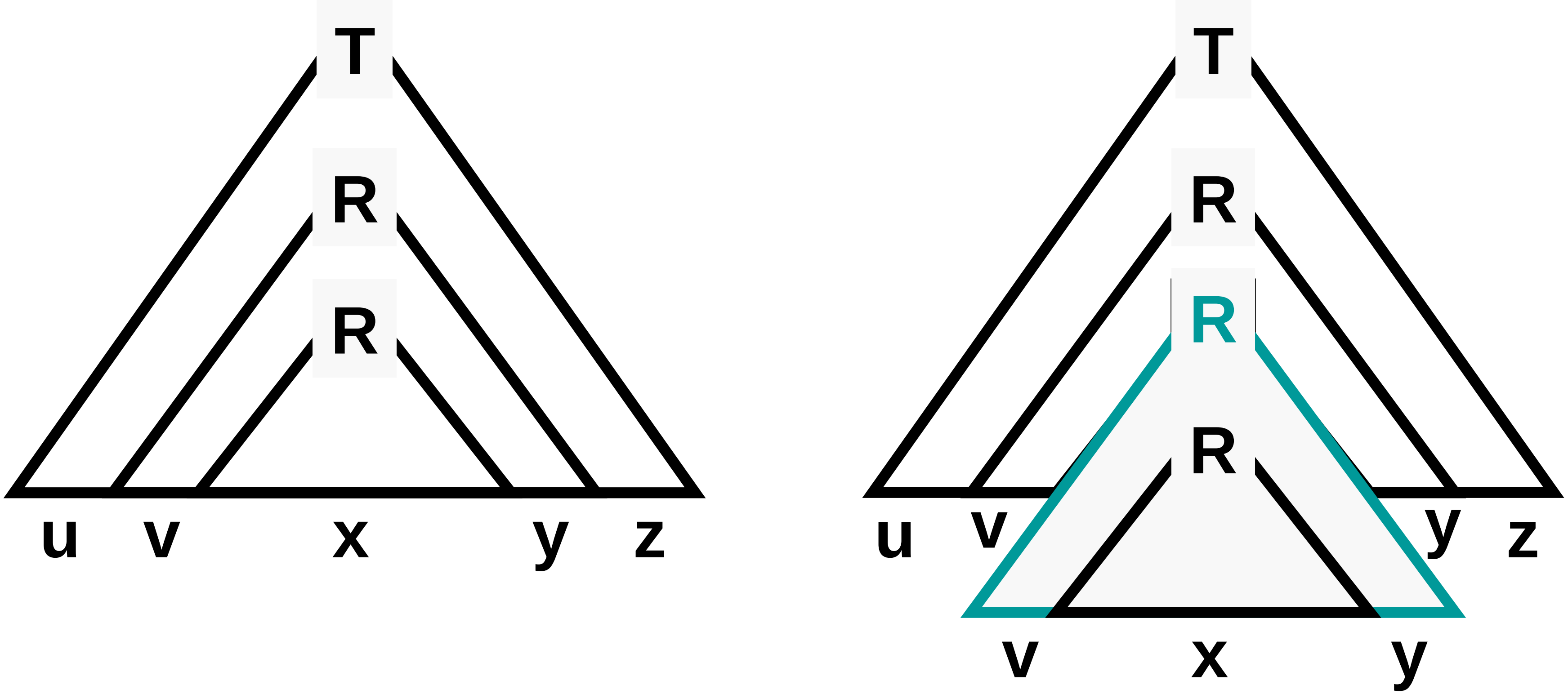
I-clicker problem (frequency: AC)

Example: L = { w $\in$ {a,b}* | #a > #b in w}.
w = aab; u,v,x,y,z=?

A. $v = a, x = a, y = b$
B. $v = aa, x = \varepsilon, y = b$
C. $v = aa, x = \varepsilon, y = \varepsilon$
D. More than one choice above works.
E. None of the choices above work.

Pumping lemma: proof idea

If string w is long enough, then every parse tree for w must have a path that contains a variable more than once.



Pumping lemma: proof

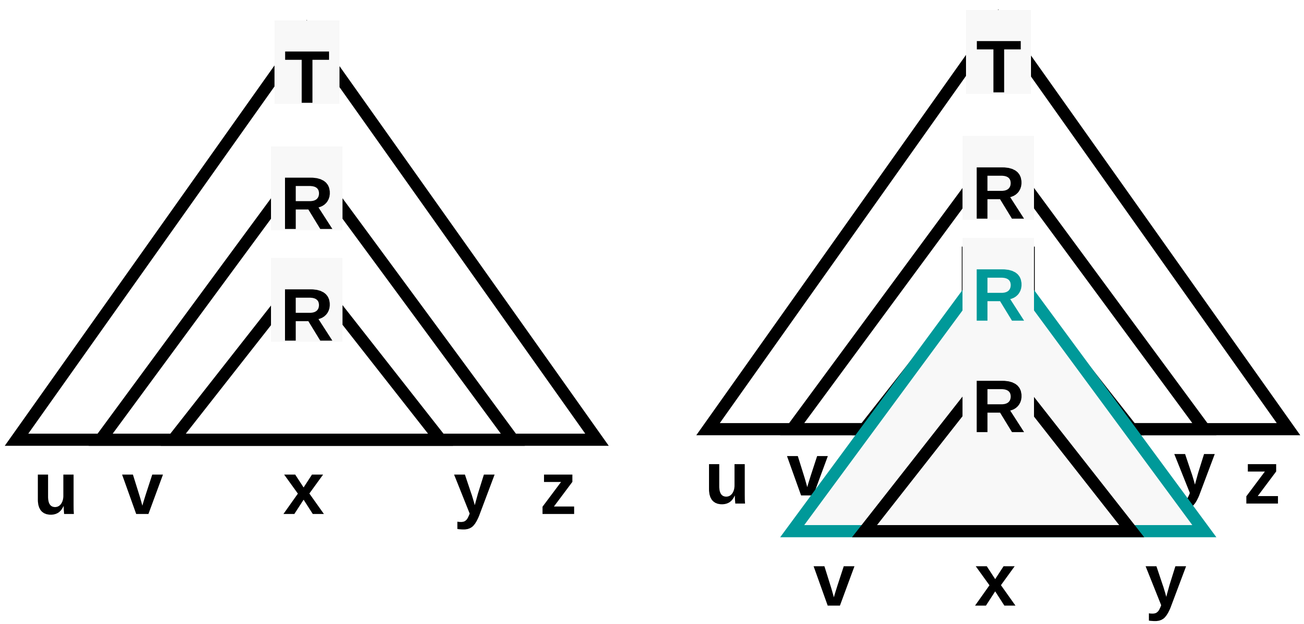
- Let b be the maximum number of symbols on the right-hand side of a rule.
- If the height of a parse tree is h, the length of the string generated is at most: b^h
- Let |V| be the number of variables in G.
- Define p = $b^{|V|+2}$.
- Let w be a string of length at least p.
- Let T be the parse tree for w with the smallest number of nodes.
- T must have height at least |V|+2.

Pumping lemma: proof

The longest path in T must have ≥ |V|+1 variables

Select R to be the variable that repeats among the lowest |V|+1 variables

1. **$|vy| > 0$**
2. **$|vxy| \le P$**



Negating the pumping lemma

L is not context-free

If for every P,
there is a $w \in L$ with $|w| \geq P$

for every $uvxyz=w$, where:

1. $|vy| > 0$
2. $|vxy| \leq P$

there is an $i \geq 0$, $uv^ixy^iz \notin L$

CS 464

Using the pumping lemma

Prove $L = \{ww \mid w \in \{0,1\}^*\}$ is not context-free

Assume L is context-free.

Then there is a pumping length P.

No matter what P is, the string $s = 0^P1^P0^P1^P$ has $|s| \geq P$ and $s \in L$.

So there should be uvxyz=s with:

1) $|vy| > 0$, 2) $|vxy| \leq P$, 3) $\forall\ i,\ uv^ixy^iz \in L$.

$$s = \underbrace{00\ldots00}_{P}\underbrace{11\ldots11}_{P}\underbrace{00\ldots00}_{P}\underbrace{11\ldots11}_{P}$$

{ww | w∈{0,1}*} is not a CFL: proof

No matter what P is, the string s = $0^P1^P0^P1^P$ has $|s| \geq P$ and $s \in L$

Then pumping down must remove at least one 1 or one zero; e.g. uxz = $0^P1^i0^j1^P$ where i ≠ j. Both should be …

vxy cannot be only in the first half since pumping down would move a 0 to the end of the first half.

vxy cannot be only in the last half since pumping up would move a 0 to the end of the first half

So there should be u,v,x,y,z with 1) $|vy| > 0$, 2) $|vxy| \leq P$, 3) $\forall i$, $uv^ixy^iz \in L$ … they are chosen, … cannot be context free!

P P P P

$s = \underbrace{00...00}_{P}\underbrace{11...11}_{P}\underbrace{00...00}_{P}\underbrace{11...11}_{P}$

vxy