Lecture 8

Last time:
• Context-free grammars (CFG)
• Equivalence of CFGs and PDAs

Today:
• Converting a PDA to a CFG
• Pumping Lemma for CFLs
A language is generated by a CFG

⇒

It is recognized by a PDA
Converting a PDA to a CFG

Given PDA $P = (Q, \Sigma, \Gamma, \delta, q, F)$
Construct a CFG $G = (V, \Sigma, R, S)$ with $L(G) = L(P)$

First, simplify $P$ so that:

1. It has a single accept state, $q_{\text{accept}}$

2. It empties the stack before accepting

3. Each transition does exactly one of:
   • pushes a symbol;
   • pops a symbol.
SIMPLIFY

$q_0$ → $q_1$ (ε,ε → $\$\$)$
$q_0$ → $q_2$ (0,ε → 0)
$q_0$ → $q_3$ (ε,ε → θ)
$q_1$ → $q_2$ (1,0 → ε)
$q_2$ → $q_3$ (1,0 → ε)
$q_3$ → $q_4$ (ε,ε → θ)
$q_3$ → $q_5$ (ε,0 → ε)
$q_4$ → $q_5$ (ε,ε → θ)
$q_5$ → $q_1$ (ε,ε → θ)
For each pair of states $p$ and $q$ in $P$, add a variable $A_{pq}$ to CFG that generates all strings that can take $P$ from $p$ to $q$ without changing the stack:

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

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

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

2/4/2016

Sofya Raskhodnikova; based on slides by Nick Hopper
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
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 $x$
$V = \{ A_{pq} \mid p,q \in Q \}$

$S = A_{q_0q_{\text{accept}}}$

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

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

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 $A_{pp} \rightarrow \epsilon$
What strings does $A_{q_0 q_1}$ generate?  none
What strings does $A_{q_1 q_2}$ generate?  $\{0^n1^n \mid n > 0\}$
What strings does $A_{q_1 q_3}$ generate?  none
A language is generated by a CFG

\[ \iff \]

It is recognized by a PDA
1. Ambiguous CFGs.
2. Chomsky normal form for CFGs
3. (Skipping Chapter 2.4 in Sipser).
Context-free or not?

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

**YES** \( L_2 = \{ xy \mid x, y \in \{0,1\}^*, |x|=|y| \text{ and } x \neq y \} \)
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. $uv^i xy^i z \in L$ for all $i \geq 0$
Examples

there exist $uvxyz = w$, where:

Example: $L = \{w \in \{0,1\}^* \mid w = w^R\}$.

1. $w = 0$; $u,v,x,y,z$ = ?
2. $|vxy| \leq P$

Example: $L = \{w \in \{a,b\}^* \mid \#a > \#b \text{ in } w\}$.

3. $uvxyz \in L$ for all $i \geq 0$

1. $w = a$; $u,v,x,y,z = ?$
2. $w = aab$; $u,v,x,y,z = ?$
I-clicker problem (frequency: AC)

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

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.
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 $|V|+2$. 
The longest path in T must have $\geq |V|+1$ variables

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

1. $|vy| > 0$
2. $|vxy| \leq P$
Negating the pumping lemma

If \( L \) is not context-free,

then there exists a \( w \in L \) with \( |w| \geq P \) for every \( P \),

there is an \( \nu \), \( \nu x y z = w \), where:

1. \( |vy| > 0 \)
2. \( |vxy| \leq P \)

there is an \( i \geq 0 \), \( \nu v^i x y^i z \notin L \)
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

\[ |s| \geq P \text{ and } s \in L. \]

So there should be \( uvxyz = s \) with:

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

\[
s = 00...0011...1100...0011...11
\]
No matter what P is, the string s = 0P1P0P1P has length P, so there should be uvxyz = s with:

1) |vy| > 0
2) |vxy| ≤ P
3) ∀i, uvixyiz ∈ L.

Then pumping down would move a P 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. P

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

s = 00...0011...1100...0011...11

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.

P

P

P

P

P

s = 00...0011...1100...0011...11

vxy

{ww | w ∈ {0,1}^*} is not a CFL: proof