02/09/05 — *Lecture 8*

CFG ⇒ PDA

a
A
S
b
a

Stores: aASba

- Top of the stack points to the "beginning" of the intermediate string.
- Any time a terminal is on the top, we will read it in <u>and</u> read in the corresponding input character

- Popping a variable ⇒ replace with rule $A \to aSb$

pop A push aSb

$\{\delta(q, a, s) \longrightarrow (r, \underbrace{U}_{\text{is a string}} = \underbrace{u_1 u_2 \dots u_\ell}_{\text{characters}})$

$$\begin{cases} \delta(q, a, s) \to (q_1, u_\ell) \\ \delta(q_1, \varepsilon, \varepsilon) \to (q_2, u_{\ell-1}) \\ \vdots \\ \delta(q_{\ell-1}, \varepsilon, \varepsilon) \to (r, u_1) \end{cases}$$

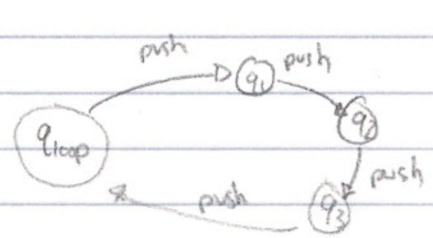
push more than one character, need more transition rules.

STATES = $q_{start}, q_{loop}, q_{accept} \cup \underbrace{E}_{\text{added states}}$

$ - start token

read pop

$\delta(q_{start}, \varepsilon, \varepsilon) = (q_{loop}, S\$)$
$\delta(q_{loop}, \varepsilon, A) = \{(q_{loop}, w) \mid A \to w \text{ is a rule}\}$
$\delta(q_{loop}, a, a) = \{(q_{loop}, \varepsilon)\}$
$\delta(q_{loop}, \varepsilon, \$) = \{(q_{accept}, \varepsilon)\}$



Ex: <u>Grammar</u>

$S \to ASA \mid aBb$

$A \to B \mid S$

$B \to b \mid \varepsilon$

<u>Derivation</u>: $\overset{1}{S} \to \overset{2}{ASA} \to \overset{3}{BSA} \to \overset{4}{b}\,\underset{5}{SA} \to b\underset{6}{a}\overset{7}{B}bA \to ba\underset{8}{b}\overset{9}{A} \to bab\overset{10}{B} \to bab$ (11)

<u>Stack</u>: input: bab

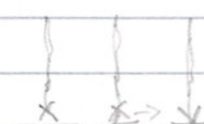
					a					
	A	B	b		B	B				
	S	S	S	S	b	b	b			
S	A	A	A	A	A	A	A	A	B	
$	$	$	$	$	$	$	$	$	$	$
1	2	3	4	5	6	7	8	9	10	11

PDA ⇒ CFG

each pair of states p + q, creates a variable A_{pq} (Will generate all strings that can take us from state p on an empty stack to state q on empty stack)

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

→ What pushed stays to end
-OR- → doesn't stay



$A_{pq} \to a A_{rs} b$

? read a from state p taking you to state r, read b from state s taking you to q

Exam Review (by book section)

OPEN notes: 2 pages front + back

1.1 DFA

Formal def: Q = states

Σ = input alpha

$\delta: Q \times \Sigma \to Q$

$q_0 \in Q$ (start state)

$F \subseteq Q$

1.2 NFA

everything same except transition function δ

* $\delta: Q \times \Sigma_\varepsilon \to P(Q)$

// epsilon transitions
// get into state + get stuck
// accepts if at least one accepts

1.3 Regular Expressions

$\underbrace{a, \varepsilon, \emptyset,}_{\text{atoms}} R_1 \cup R_2, R_1 \circ R_2, R_1^*$

Constructions

NFA $\to$ DFA (n states $\to 2^n$ states)

NFA $\to$ GNFA $\to$ RE

repeatedly ripping out a state

R.E. $\to$ NFA

1.4 Pumping Lemma

for any string A in a reg lang of length p or longer
there exists a way to partition A into xyz such that

1) $xy^iz \in L \quad i=0,1,2\ldots$
2) $|y| > 0$
3) $|xy| \leq p$

① Pick one string of length p or longer
② Consider ALL ways to partition this string within the restrictions
③ Show that for all of these possibilities, there exists an i such that $xy^iz \notin L$

CH 2

Chomsky Normal Form

2.1 CFG: V = variables CFG → CNF
Σ = alphabet (terminals)
R = rules $V \rightarrow$ string of V, Σ
S = start symbol

2.2 PDA: Q = states
Σ = input alpha
Γ = stack alpha
$\delta: Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \rightarrow P(Q \times \Gamma_\varepsilon)$
$q_0 \in Q$
$F \subseteq Q$

2.3 Pumping lemma for CFL

1) $uv^ixy^iz \in L \quad i=0,1,2$
2) $|vy| > 0$
3) $|vxy| \leq p$

2.3 Pumping lema for CFL

1) $uv^ixy^iz \in L$ $\quad i=0,1,2$
2) $|vy| > 0$
3) $|vxy| \leq p$