10-4

Chomsky Normal Form:

Every rule in R is either:

$A \to BC$ $\quad A \in V, B \in V-S, C \in V-S$

or $A \to a$ $\quad a \in \Sigma$

or $S \to \varepsilon$

Algorithm:
1. Add a new start state
2. Collapse $A \to \varepsilon$ rules up
3. Remove unit rules $(A \to B)$
4. Add intermediate variables

Example: $S \to 0S1 \mid \varepsilon$

(1) $S_{new} \to S$
$S \to 0S1 \mid \varepsilon$

(2) $S_{new} \to S \mid \varepsilon$
$S \to 0S1 \mid 01$

(3) $S_{new} \to 0S1 \mid 01 \mid \varepsilon$
$S \to 0S1 \mid 01$

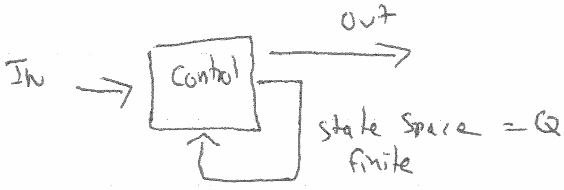
(4) $S_{new} \to AC \mid AB \mid \varepsilon$
$S \to AC \mid AB$
$A \to 0$ $\quad C \to SB$
$B \to 1$

$S \to \varepsilon$
$S \to a$
$S \to AB$
$AB \to aB$
$AB \to CDB$

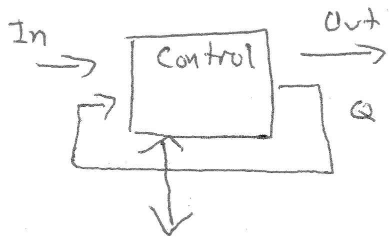
If input string has n-chars
the derivation is ..
tree is at most n+1 high
when $|s| = 2^n+1$

11-1) L-system

PDA - push-down automata is the "machine" for CFL



DFA

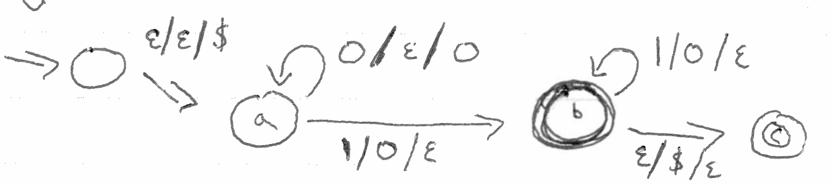


PDA actions: ignore stack
push onto stack
pop stack
replace the top of stack

Turn access 3 back into
access 1 back?

$Q_0 \rightarrow (Q_0, S_0) \Rightarrow (Q_0, S_0, S_1)$

$Q_0, S_2 \leftarrow (Q_0, S_0), S_2 \leftarrow (Q_0, S_0, S_1), S_2$



$0^n 1^n$

a/b/c

in — top of stack — new top of stack

a/ε/ε — ignore
a/ε/c — push
a/b/ε — pop
a/b/c — replace

stack [q] input

) $[a]0011 ⇒ $0[a]011 ⇒ $00[a]11
⇒ $0[b]1 ⇒ $[b] ⇒ [c] ✓

$S \rightarrow 0S1 \mid \varepsilon$

) ε[a]00111 ⇒* $[b]1 ⇒ X

) ε[a]00011 ⇒* 00[b]1 ⇒ $0[b] X

1-2/

A PDA $p = \langle Q, \Sigma, \Gamma, \delta, q_0, F \rangle$

Q = a finite set (states)

Σ = an alphabet (input)

Γ = an alphabet (stack language)

$q_0 \in Q$ (start state)

$F \subseteq Q$ (accept states)

DFA: $\delta : Q \times \Sigma \Rightarrow Q$

NFA: $\delta : Q \times \Sigma_\varepsilon \Rightarrow P(Q)$

transition w/o reading — non-det

PDA: $\delta : Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \rightarrow P(Q \times \Gamma_\varepsilon)$

transition w/o read — ignore stack — read stack — non-det — write stack — doesn't write

$L(p) = \{ w \mid q_0 \overset{w}{\underset{\varepsilon}{\Rightarrow}}{}^* q_f \text{ where } q_f \in F \}$

q_i reaches q_j on input w with stack g

$$q_i \overset{w}{\underset{g}{\Rightarrow}}{}^* q_j$$

$$q_i \overset{\varepsilon}{\underset{g}{\Rightarrow}}{}^* q_i \qquad \frac{\delta(q_i, a, b) \ni (q_j, c) \quad q_j \overset{w}{\underset{cg}{\Rightarrow}}{}^* q_k}{q_i \overset{aw}{\underset{bg}{\Rightarrow}}{}^* q_k} \quad a \in \Sigma_\varepsilon \quad b, c \in \Gamma_\varepsilon$$

Prove that CFG (CFL) = PDA

1. $\forall G \in CFG.\ \exists p \in PDA.\ L(p) = L(g)$ ← compiler
2. $\forall p \in PDA.\ \exists G \in CFG.\ L(G) = L(P)$ ← dissassembler

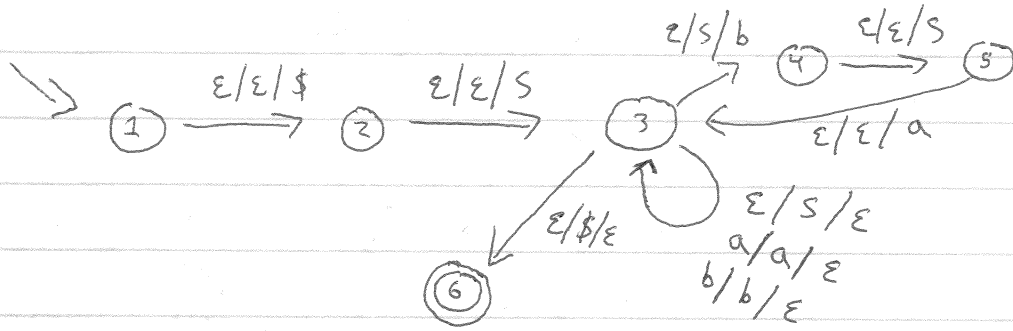
11-3/ Compiler : Idea : Store the right part of the string in the PDA's stack

$S \to aSb \mid \varepsilon$ aabb

$\varepsilon\ [1]\ aabb \to \$\ [2]\ aabb \to \$\ S\ [3]\ '' \xrightarrow{4,5} \$\ b\ S\ a\ [3]\ aabb$

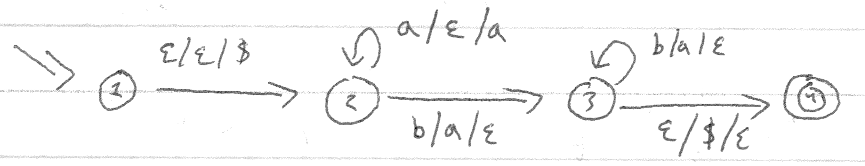
$\to \$bS\ [3]\ abb \xrightarrow{4,5} \$bbSa\ [3]\ abb \to \$bbS\ [3]\ bb \to \$bb\ [3]\ bb$

$\to \$b\ [3]\ b \to \$\ [3] \to [6]$ ✓



compiler *

$\Gamma = \Sigma \cup \{\$, S\}$



handmade

$\Gamma = \Sigma \cup \{\$\}$

$\varepsilon\ [1]\ aabb \Rightarrow \$\ [2]\ aabb \to \$a\ [2]\ abb \to \$aa\ [2]\ bb$

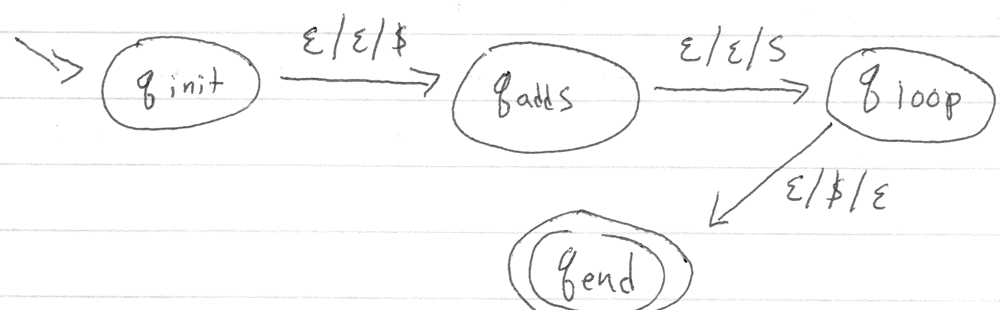
$\to \$a\ [3]\ b \to \$\ [3] \to [4]$ ✓

Assume G is CNF

in : $G = (\Sigma, V, R, S)$

out : $P = (Q, \Sigma, \Gamma, \delta, q_0, F)$

$\Gamma = V \cup \Sigma \cup \{\$\}$ $\quad q_0 = q_{init}$ $\quad F = \{q_{end}\}$



$Q = q_{init}$
q_{loop}
q_{end}
q_{ADDx} for $x \in V \cup \Sigma$

$\forall a \in \Sigma.\ \delta(q_{loop}, a, a) = \{(q_{loop}, \varepsilon)\}$

if $(S, \varepsilon) \in R,\ \delta(q_{loop}, \varepsilon, S) = \{(q_{loop}, \varepsilon)\}$

$\forall V, a.\ (V, a) \in R,\ \delta(q_{loop}, \varepsilon, V) \ni \cancel{\{}(q_{loop}, a)$

$\forall A, B, C.\ (A, BC) \in R,\ \delta(q_{loop}, \varepsilon, A) \ni (q_{addB}, C)$

$\wedge\ \delta(q_{ADDB}, \varepsilon, \cancel{B}) \ni ((q_{loop}, B))$