Thursday, September 10, 2020
Context-free Automata

$L_1 = \{0^n 1^n \mid n \geq 0\}$

- Not Regular
0011 or 000111

DEFINITIONS
1). Production $a \Rightarrow b$
2). Variable holder for state A, B (capitals)
3). Terminals symbols of language 0, 1, a, b, c
4). Start State

Example $L = \{0^n \# 1^n \mid n \geq 0\}$

$A \Rightarrow 0A1$
$A \Rightarrow B$
$B \Rightarrow \#$

$A \Rightarrow 0A1 \mid B$
$B \Rightarrow \#$

$A \Rightarrow 0A1 \Rightarrow 00A11 \Rightarrow 000A111 \Rightarrow 000B111 \Rightarrow 000\#111$

Derivation

Example: Hopcroft pg 158
$A_1 = \{w \mid w \in \{0,1\}, w \text{ is a palindrome}\}$
$\varepsilon \mid 0 \mid 1 \mid 01110 \mid 0110$

$P \Rightarrow \varepsilon \mid 0 \mid 1 \mid 0P0 \mid 1P1$

$P \to \varepsilon$
$P \Rightarrow 0$
$P \Rightarrow 1$
$P \Rightarrow 0P0$
$P \Rightarrow 1P1$

Derivation of 01010
$P \Rightarrow 0P0 \Rightarrow 01P10 \Rightarrow 01010$

FORMAL DEFINITION $\to$ 4-tuple (V, Σ, R, S)
V is a finite set called the Variables
Σ is a finite set, disjoint from V called the terminals
R is a finite set of rules with a variable on the LHS & Variables and/or terminals on the RHS
$S \in V$ is the start variable.

Chomsky Normal Form

Two Forms For Rules:

1). $A \Rightarrow BC$

2). $A \Rightarrow a$

B, C MAY NOT BE THE START VARIABLE

$S \Rightarrow \varepsilon$ where S is the START VARIABLE

CFG ⇒ CNF CONVERSION

a) Add a new START VARIABLE S_0 AND A RULE $S_0 \Rightarrow S$

b). Eliminate ε rules

c). ELIMINATE UNIT Rules $A \Rightarrow B$

d). Add VARIABLES to MAKE proper form

Example: Sipser Ex 2.10

$S \Rightarrow ASA \mid aB$

$A \Rightarrow B \mid S$

$B \Rightarrow b \mid \varepsilon$

ab

a

ababab

a). $S_0 \Rightarrow S$

the same

b). $S_0 \Rightarrow S$

$S \Rightarrow ASA \mid aB \mid a$

$A \Rightarrow B \mid S \mid \varepsilon$

$B \Rightarrow b$

$S_0 \Rightarrow S$

$S \Rightarrow ASA \mid aB \mid a \mid SA \mid AS \mid S$

$A \Rightarrow B \mid S$

$B \Rightarrow b$

c). $S_0 \Rightarrow S$

$S \Rightarrow ASA \mid aB \mid a \mid SA \mid AS$

$A \Rightarrow B \mid S$

$B \Rightarrow b$

$S_0 \Rightarrow S$

$S \Rightarrow ASA \mid aB \mid SA \mid AS$

$A \Rightarrow b \mid S$

$B \Rightarrow b$

$S_0 \Rightarrow S$

$S \Rightarrow ASA \mid aB \mid SA \mid AS$

$A \Rightarrow b \mid ASA \mid aB \mid SA \mid AS$

$B \Rightarrow b$

$S_0 \Rightarrow ASA \mid aB \mid SA \mid AS$

$S \Rightarrow ASA \mid aB \mid SA \mid AS$

$A \Rightarrow b \mid ASA \mid aB \mid SA \mid AS$

$B \Rightarrow b$

d). $S_0 \Rightarrow AA_1 \mid aB \mid SA \mid AS$

$A_1 \Rightarrow SA$

replace all other SA with A_1

$S_0 = AA_1 \mid L_a B \mid SA \mid AS$

$S = AA_1 \mid L_a B \mid SA \mid AS$

$A = b \mid AA_1 \mid L_a B \mid SA \mid AS$

$A_1 = SA$

$B = b$

$L_a = a$

Let A = 10;

Let AR = &A;

} - - - - -

Life of A

Life of AR

← Invalid

Pushdown Automata

NFA + Stack

Push & Pop from stack as part of transitions

Example: $L_{ww_R} = \{ww_R \mid w \in \{0,1\}^*\}$ 0110 00 110011

$q_0 \rightarrow$ read & store symbols on stack

$q_1 \Rightarrow$ compare current symbol to top of stack, if match pop & continue

if stack is empty & out of symbols $\rightarrow$ accept

Formal Definition: 7-tuple $P = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$

Q: Set of states

Σ: Set of symbols

Γ: Stack Alphabet $\rightarrow$ Set of symbols we can put on stack

δ: Transition Function $\delta(q, a, X)$ where $X \in \Gamma$

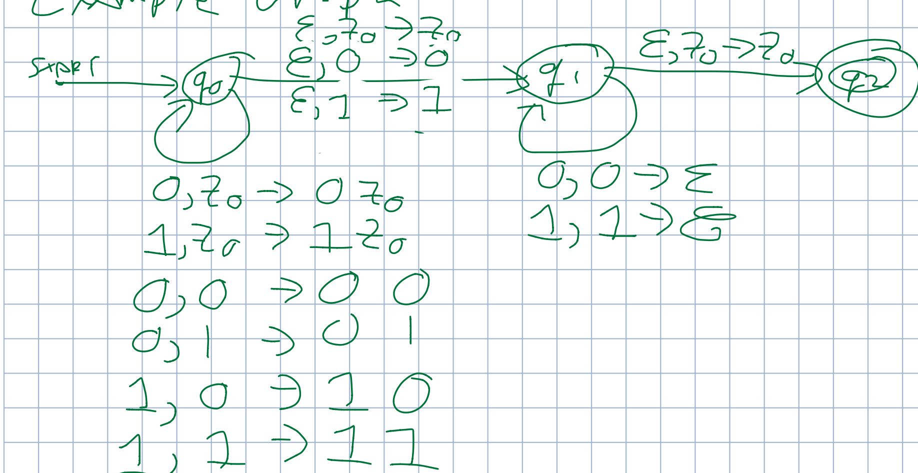
$\Rightarrow (q_{next}, Z)$

q_0: Start state

z_0: Stack start symbol, frequently '\$'

F: Accepting States

Example Graph



Sipser Ex 2.14

$L = \{0^n 1^n \mid n \geq 0\}$

ε
01
0011
000111

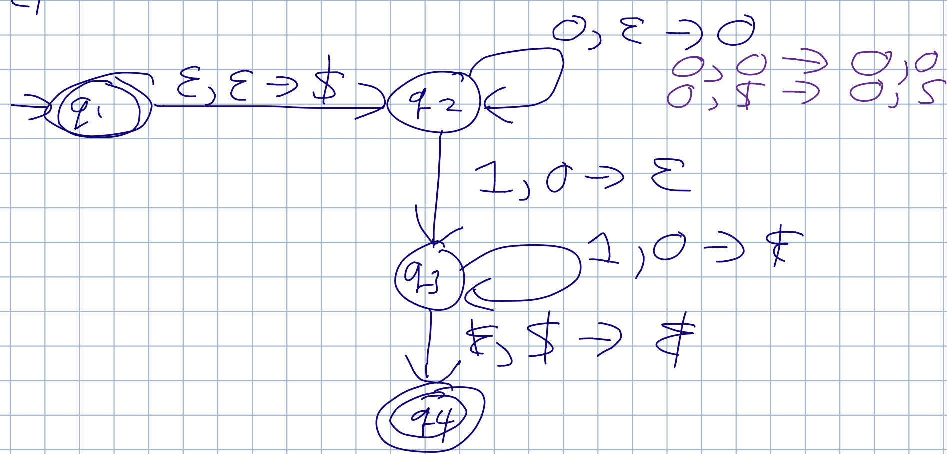
$Q = \{q_1, q_2, q_3, q_4\}$

$\Sigma = \{0, 1\}$

$\Gamma = \{0, \$\}$

$F = \{q_1, q_4\}$

δ Input / Stack	0: 0	0: \$	0: ε	1: 0	1: \$	1: ε	ε: 0	ε: \$	ε: ε
S_1									$(q_2, \$)$
S_2			$(q_2, 0)$	(q_3, ε)					
S_3				(q_3, ε)				(q_4, ε)	
S_4									



CFG ⟺ PDA