CIE-II

Theory of computation (CS354TA)

1.1 Definition of left recursion - 1M

Elimination of left recursion - 1M

$S \rightarrow (L) \mid a$

$L' \rightarrow , SL' \mid \epsilon$, $L = SL'$

1.2 CFG to PDA.

$\delta(q_0, \epsilon, Z) = (q_1, SZ)$, $\delta(q_1, \epsilon, S) = \{(q_1, a), (q_1, aA), (q_1, B)$

$\delta(q_1, \epsilon, A) = \{(q_1, aB), (q_1, \epsilon)\}$, $\delta(q_1, \epsilon, B) = \{q_1, Aa)$

1.3 $S \rightarrow aSbb \mid A \mid B$ 1.4 one 1.5 L = R = P

$A \rightarrow aA \mid a$

$B \rightarrow bB \mid b$.

1.6 A PDA is deterministic iff

a) $\delta(q, a, X)$ has atmost one member for any $a \in \{\Sigma \cup \epsilon\}$ - 1M

b) If $\delta(q, a, X)$ is non empty for some $a \in \Sigma$, then $\delta(q, \epsilon, X)$ must be empty. - 1M

2a) PDA → 3 M, Graphical representation - 2M, ID-2M

$\delta(q_0, a, Z) = (q_0, aZ)$, $\delta(q_0, a, b) = (q_0, \epsilon)$ $\delta(q_0, b, a) = (q_0, \epsilon)$

$\delta(q_0, b, Z) = (q_0, bZ)$, $\delta(q_0, a, c) = (q_0, \epsilon)$ $\delta(q_0, b, b) = (q_0, bb)$

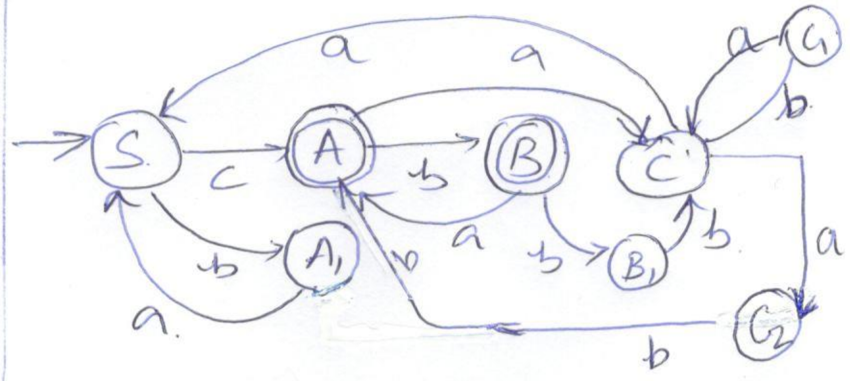
$\delta(q_0, c, Z) = (q_0, cZ)$, $\delta(q_0, a, a) = (q_0, aa)$ $\delta(q_0, b, c) = (q_0, \epsilon)$

$\delta(q_0, \epsilon, Z) = (q_f, Z)$

$\delta(q_0, c, a) = (q_0, \epsilon)$

$\delta(q_0, c, b) = (q_0, cb)$

$\delta(q_0, c, c) = (q_0, cc)$

2b) 

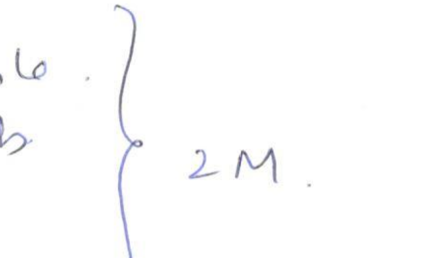
3a) (i) Reverse the automata. — 2M
(ii) Obtain right linear grammar - 2M
(iii) Reverse productions of right linear grammar - 2M.

3b) (i) Languages accepted by final state - 2M
(ii) Languages accepted by empty stack - 2M.

4a) Definition of CNF. — 2M
(i) NO useless production.
(ii) A and B are nullable variable.
$S \to abAB \mid abB \mid abA \mid ab$
$A \to a \mid Aa$
$B \to b$
} 2M.
(iii) No Unit productions.

Converting to CNF - 2M.
$S \to A_1A_2 \mid A_1B \mid A_1A \mid A_3B$.
$A_2 \to AB$, $A \to a \mid AA_3$.
$A_1 \to A_3B$. $B \to b$.
$A_3 \to a$.

4b) Definition of ambiguity - 1M.
Proving for ambiguity - 2M.
Unambiguous grammar - 1M.



LMD1
$S \to aaB$
$S \to aab$

LMD2
$S \to AB$
$A \to$ ~~AaB~~
$A \to aaB$
$A \to aab$

2 parse trees hence ambiguous.
Unambiguous grammar
$S \to AB$
$A \to a \mid Aa$, $B \to b$.

5a) Algorithm — 2 M
Conversion — 7M
Language accepted — 1M $L = \{n_0(w) = n_1(w) \mid w \in (0,1)^*\}$
Start symbol $q_0 Z q_1$ — 1M

$\delta(q_0, 0, Z) = (q_0, AZ)$ $4 * 1 = 4$ M

$q_0 Z q_0 \rightarrow 0(q_0 A q_0)(q_0 Z q_0) \mid 0(q_0 A q_1)(q_1 Z q_0)$
$q_0 Z q_1 \rightarrow 0(q_0 A q_0)(q_0 Z q_1) \mid 0(q_0 A q_1)(q_1 Z q_1)$

111^ly for other transitions writing CFG.

$\delta(q_0, 0, B) = (q_0, \epsilon)$ $q_0 B q_0 \rightarrow 0$
$\delta(q_0, 1, A) = (q_0, \epsilon)$ $q_0 A q_0 \rightarrow 1$
$\delta(q_0, \epsilon, Z) = (q_1, \epsilon)$ $q_0 Z q_1 \rightarrow \epsilon$
2M

6a) CFG for L_1 $S_1 \rightarrow a S_1 b \mid \epsilon$
$S_0 \rightarrow S_1 S_2$ $S_2 \rightarrow Cc \mid \epsilon$
1.5M

CFG for L_2 $S_4 \rightarrow S_5 S_6$
$S_5 \rightarrow a S_7 \mid \epsilon$
$S_6 \rightarrow b S_6 c \mid \epsilon$
1.5M

This proves that L1 & L2 are CFLs

L_3 $S \rightarrow S_0 \mid S_4$ — 1M

Proof for $L_1 \cap L_2 \notin$ CFL — 2M

6b) Applications of CFGs.

(i) In compiler design during parsing for syntactic checking.

(ii) XML and Document-type definition.

(iii) Markup languages.