SCHOOL OF COMPUTER SCIENCE AND ENGINEERING (SCOPE)
FALL SEMESTER 2019
Continuous Assessment Test - II
September - 2019

Course Code: CSE2002 | **Max. Marks: 50**
Course Name: Th. of Computation and Comp. Design | **Duration: 90 Minutes**
Slot: A1 | **Date: 29-09-2019**

Answer ALL questions. You must justify your answers to get full marks.

1. Give a CFG for the language $\{a^n b^m a^{2n} : n, m \geq 0\}$. In your solution, explain what each of your variables generates. [4]

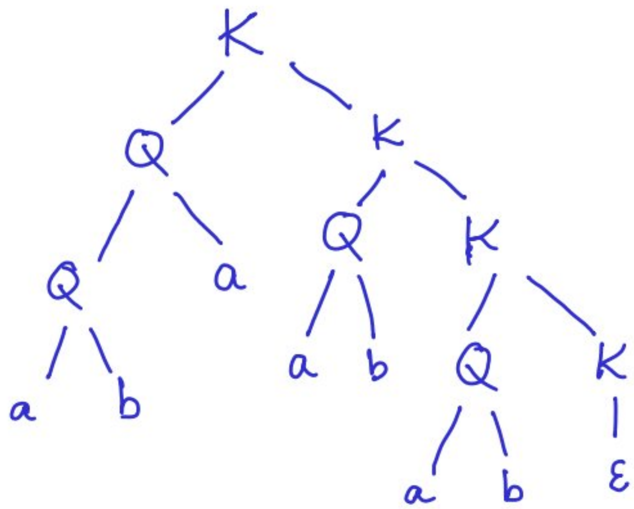
$$\boxed{\begin{array}{l} S \to aSaa \mid B \\ B \to bB \mid \epsilon \end{array}}$$

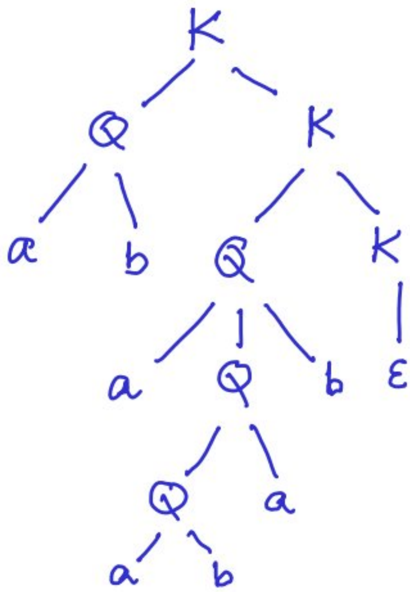
Here, S generates the language and B generates $L(b^*)$.

2. Prove that the grammar below is ambiguous by constructing at least two different parse trees for the input string *abaabab*.

$$K \to QK \mid \epsilon$$
$$Q \to Qa \mid aQb \mid ab$$

[4]





3. Construct a PDA for the language

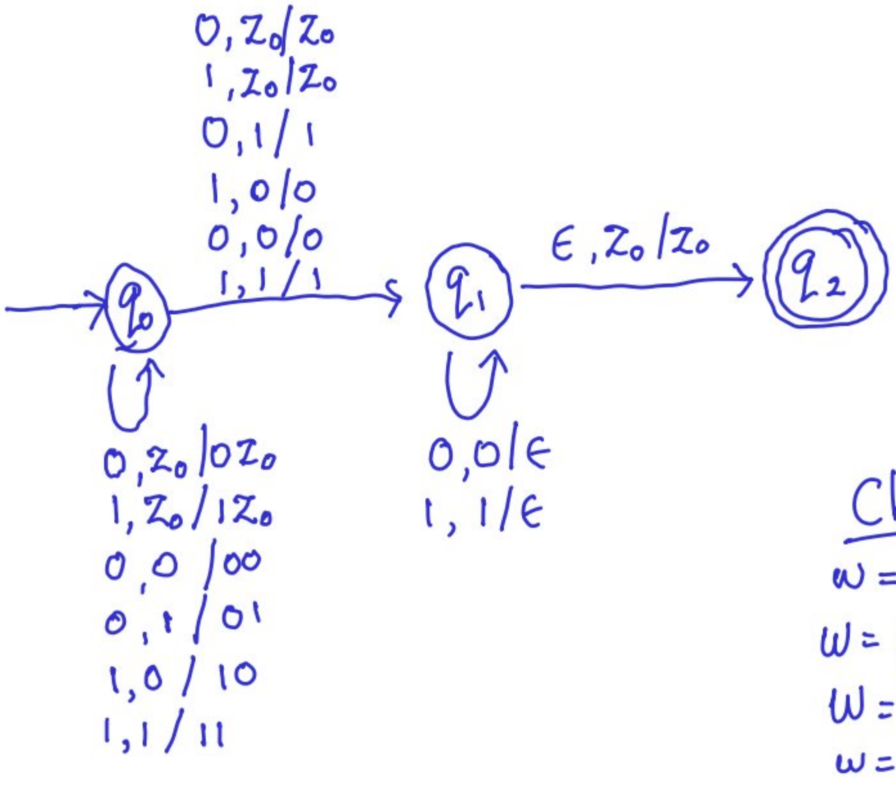
$$B = \{w \in \{0,1\}^* : w = w^{\mathcal{R}} \text{ and the length of } w \text{ is odd}\}.$$

[4]

Idea: Similar to the NPDA for $\{w : w = w^R\}$, where for next input symbol w_i

- push w_i onto stack (state q_0)
- another copy of machine guesses w_i is start of second half and matches / pops. (in q_1)

But add a transition guessing w_i is the middle bit. (ignore this middle bit) from q_0 to q_1.



Check:

$w = 0110$ reject ✓

$w = 01010$ accept ✓

$w = 0$ accept ✓

$w = \varepsilon$ reject ✓

4. Let G be the grammar

$$
\begin{aligned}
E &\to XY \\
X &\to ZZ \mid + \mid t \\
Y &\to YZ \mid * \\
Z &\to ZY \mid YX \mid +
\end{aligned}
$$

$+**t$

Let the input string be $w = abbc$. Use the CYK algorithm to determine whether the string w is in the language $L(G)$. Show the table computed by the CYK algorithm. [4]

$\{E,Z,X\}$			
$\{Z\}$	$\{Y\}$		
$\{E,Z\}$	ϕ	$\{Z\}$	
$\{X,Z\}$	$\{Y\}$	$\{Y\}$	$\{X\}$
$+$	$*$	$*$	t

ans :

Yes, $w \in L(G)$

b/c

$E \in \{E,Z,X\}$

5. The symmetric difference $A \triangle B$ of two sets A and B is the set of elements which belong to exactly one of the two sets. For example, if $A = \{a, b\}$ and $B = \{b, c\}$, then $A \triangle B = \{a, c\}$. Either prove the following statement or give a counterexample: if L and M are CFL's, then $L \triangle M$ is a CFL. [4]

Counterexample:

Let $M = 0^* 1^* 2^*$

$L = \{0^i 1^j 2^k : i \neq j \text{ or } j \neq k\}$

Then, $L \triangle M = \{0^n 1^n 2^n : n \geq 0\}$ is not a CFL.

6. Let L be the language $\{a^i : i \text{ is a perfect square}\}$. That is, L contains strings a, a^4, a^9, a^{16}, and so on. Prove that L is not context free. [6]

Pf. Suppose L is a CFL. Then, by the CFL pumping lemma, $\exists p \in \mathbb{N}$ s.t. $\forall s \in L$ with $|s| \geq p$, $\exists$ decomposition $s = uvwxy$ s.t.

(1) $uv^iwx^iy \in L, \forall i \geq 0$, (2) $|vx| \geq 1$, (3) $|vwx| \leq p$

Given any p (chosen by adversary),
take $s = a^{p^2}$. For any decomposition
$s = uvwxy$ (chosen by adversary, obeys (1)-(3)),
we have that

$$p^2+1 \leq |uv^2wx^2y| \leq p^2+p. < (p+1)^2$$

Hence, $uv^2wx^2y \notin L$ (b/c it's length is not a perfect square).

QED

7. Consider the following grammar, which has the set of terminals $\Sigma = \{a, b, c, d, e, f, g\}$.

$$
\begin{aligned}
S &\to AB \\
B &\to dS \mid \epsilon \\
A &\to DC \\
C &\to A \mid \epsilon \\
D &\to PE \\
E &\to eE \mid \epsilon \\
P &\to fSg \mid a \mid b \mid c
\end{aligned}
$$

(a) Obtain the first and follow sets for each non-terminal.

(b) Construct an LL(1) parsing table for the grammar. [8]

(a)

X	First(X)	Follow(X)
S	f, a, b, c	\$, g
A	f, a, b, c	d, \$, g
B	d, ε	\$, g
C	ε, f, a, b, c	d, \$, g
D	f, a, b, c	f, a, b, c, d, \$, g
P	f, a, b, c	e, f, a, b, c, d, \$, g
E	e, ε	f, a, b, c, d, \$, g

(b)

	f	g	a	b	c	d	e	\$
S	$S \to AB$		$S \to AB$	$S \to AB$	$S \to AB$			
B		$B \to \varepsilon$				$B \to dS$		$B \to \varepsilon$
A	$A \to DC$		$A \to DC$	$A \to DC$	$A \to DC$			
C	$C \to A$	$C \to \varepsilon$	$C \to A$	$C \to A$	$C \to A$	$C \to \varepsilon$		$C \to \varepsilon$
D	$D \to PE$		$D \to PE$	$D \to PE$	$D \to PE$			
E	$E \to \varepsilon$	$E \to \varepsilon$	$E \to \varepsilon$	$E \to \varepsilon$	$E \to \varepsilon$	$E \to \varepsilon$	$E \to eE$	$E \to \varepsilon$
P	$P \to fSg$		$P \to a$	$P \to b$	$P \to c$			

8. Let G be the grammar

$$
\begin{array}{ll}
1. & S \to X\$ \\
2. & X \to a \\
3. & X \to ab
\end{array}
$$

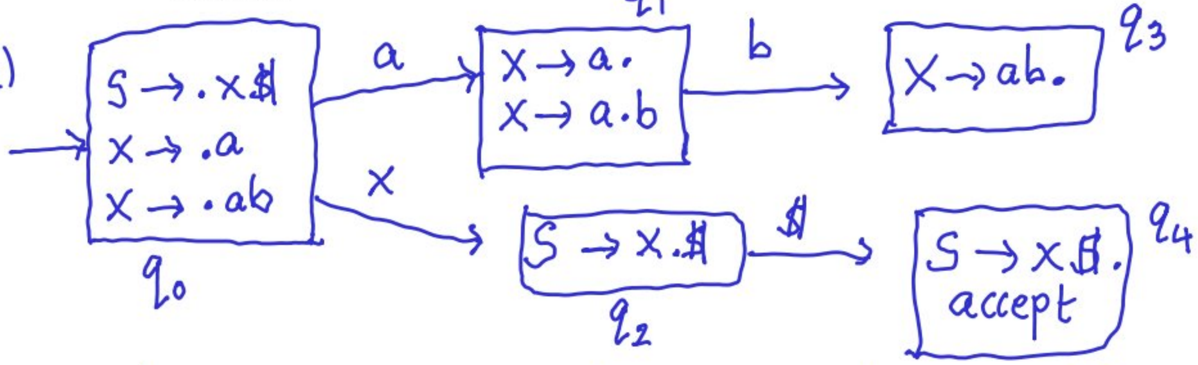
(a) Construct the parsing automaton for the grammar".

(b) Write the LR(0) parse table (the ACTION and GOTO) tables. Are there are conflicts?

(c) Write the SLR(1) parse table which uses a lookahead of one token. Are there any conflicts?

[8]

(a)



(b)

	ACTION	GOTO: a	b	\$	X
q_0	shift	q_1	e	e	q_2
q_1	shift/reduce $X \to a$		q_3		
q_2	shift	e	e	q_4	
q_3	reduce $X \to ab$				
q_4	reduce $S \to X\$$, accept				

=

LR(0) table:

	a	b	\$	X
0	s1	e	e	s2
1	r2	r2/s3	r2	r2
2	e	e	s4	
3	r3	r3	r3	r3
4	r1	r1	r1	r1

(c) Since $b \notin$ Follow(X), in state q_1, we don't reduce by $X \to a$ if next symbol is b. So, replace "r2/s3" by "s3" in LR(0) table to get SLR(1) table. No conflicts.

M[1,b] = r2/s3 is a conflict: if in state q_1, next symbol is b, then reduce by production #2 $X \to a$, or shift and change state to q_3.

9. Design a Turing machine that recognizes the language

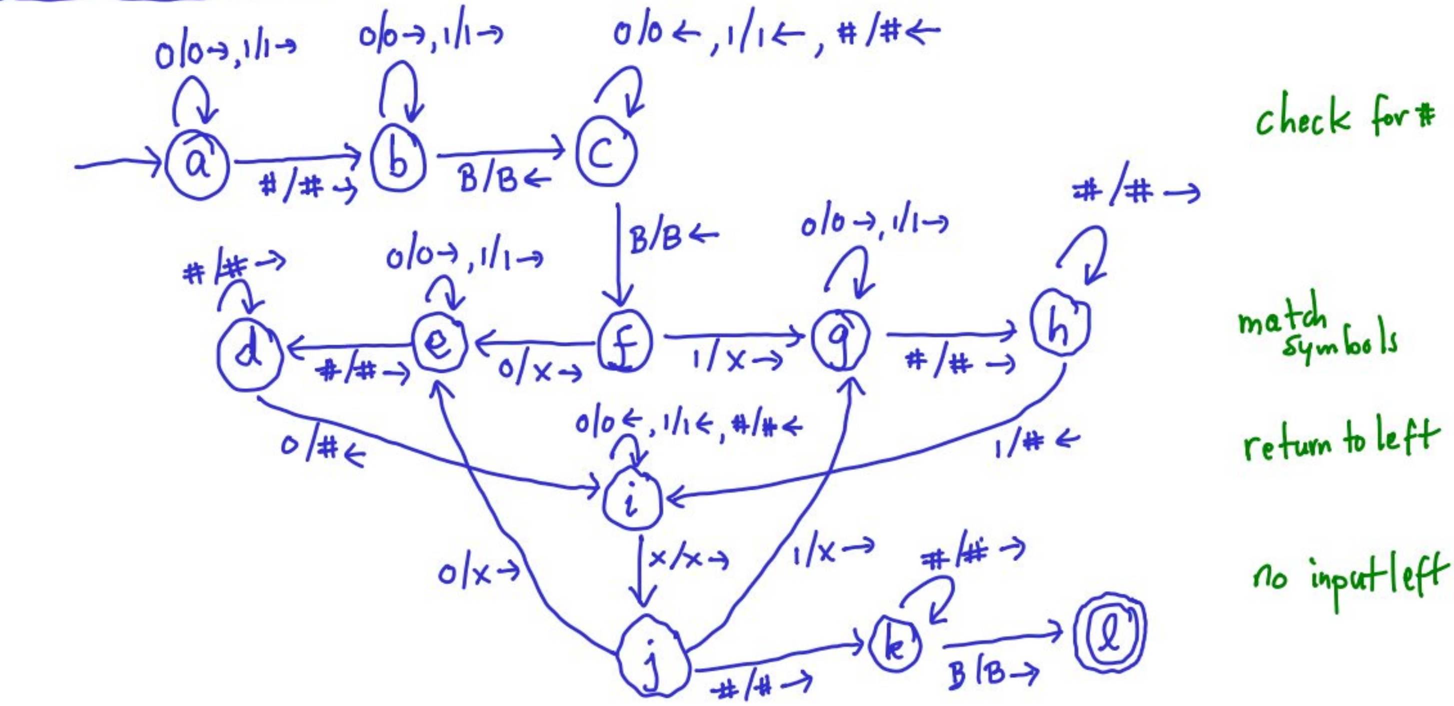
$$L = \{w\#w : w \in \{0,1\}^*\}.$$

Give your answer in the form of a state transition diagram, and explain the strategy you used to create the Turing machine. Simulate the machine on input 010#010 by giving the sequence of ID's of the machine. [8]

Strategy: 1) First check input contains a #

2) Start with leftmost symbol. Replace it with X. Jump over 0's, 1's till we hit #. Jump over #'s. If bit matches, replace it with #.

3) Move leftwards till we hit X. If next symbol is not #, repeat 2).



Sequence of ID's: a010#010 $\vdash$ f010#010 $\overset{*}{\vdash}$ Xj10##10 $\overset{*}{\vdash}$ XXj0###0 $\overset{*}{\vdash}$ XXXj####B $\vdash$ XXX####BlB.