INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

	Stamp / Signature of the Invigilator

EXAMINATION (Mid Semester)	SEMESTER (Spring)

Roll Number										Section		Name	
Subject Number	C	S	2	1	0	0	4	Subject Name	*Formal Languages and Automata Theory*				

Department / Center of the Student		Additional sheets	

Important Instructions and Guidelines for Students

1. You must occupy your seat as per the Examination Schedule/Sitting Plan.
2. Do not keep mobile phones or any similar electronic gadgets with you even in the switched off mode.
3. Loose papers, class notes, books or any such materials must not be in your possession, even if they are irrelevant to the subject you are taking examination.
4. Data book, codes, graph papers, relevant standard tables/charts or any other materials are allowed only when instructed by the paper-setter.
5. Use of instrument box, pencil box and non-programmable calculator is allowed during the examination. However, exchange of these items or any other papers (including question papers) is not permitted.
6. Write on both sides of the answer script and do not tear off any page. **Use last page(s) of the answer script for rough work.** Report to the invigilator if the answer script has torn or distorted page(s).
7. It is your responsibility to ensure that you have signed the Attendance Sheet. Keep your Admit Card/Identity Card on the desk for checking by the invigilator.
8. You may leave the examination hall for wash room or for drinking water for a very short period. Record your absence from the Examination Hall in the register provided. Smoking and the consumption of any kind of beverages are strictly prohibited inside the Examination Hall.
9. Do not leave the Examination Hall without submitting your answer script to the invigilator. **In any case, you are not allowed to take away the answer script with you.** After the completion of the examination, do not leave the seat until the invigilators collect all the answer scripts.
10. During the examination, either inside or outside the Examination Hall, gathering information from any kind of sources or exchanging information with others or any such attempt will be treated as **'unfair means'**. Do not adopt unfair means and do not indulge in unseemly behavior.

Violation of any of the above instructions may lead to severe punishment.

Signature of the Student

To be filled in by the examiner

Question Number	1	2	3	4	5	6	7	8	9	10	Total
Marks Obtained											

Marks obtained (in words)	Signature of the Examiner	Signature of the Scrutineer

CS21004 Formal Languages and Automata Theory, Spring 2019–2020

Mid-Semester Test

24–February–2020 NR-121/122/221, S-302 Maximum marks: 60

Instructions

- Write your answers in the question paper itself. Be brief and precise. Answer all questions.
- Write the answers in the respective spaces provided. Use the last two blank pages for rough work.
- If you use any theorem/example/formula/construction covered in the class, just mention it, do not elaborate.
- Write your proofs/constructions in mathematically precise language. Unclear, incomplete, and/or dubious statements would be severely penalized.

Do not write anything on this page.

Questions start from the next page.

1. Two regular expressions over the same alphabet are called *equivalent* if they generate the same language. Prove/Disprove the equivalence of the following pairs of regular expressions over the alphabet $\{a, b\}$.

(a) $(ab+a)^*a$ and $a(ba+a)^*$. **(5)**

Solution *Equivalent.* We have $(ab+a)^*a = (a(b+\varepsilon))^*a$, that is, this generates strings of the form

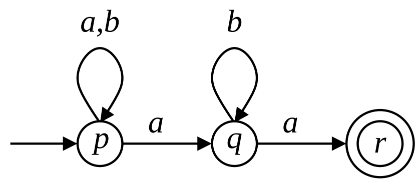
$$[a(b+\varepsilon)a(b+\varepsilon)\ldots a(b+\varepsilon)]a = a[(b+\varepsilon)a(b+\varepsilon)a\ldots(b+\varepsilon)a].$$

We finally have $a(ba+a)^* = a((b+\varepsilon)a)^*$.

(b) $(ab^*a+ba^*b)^*$ and $(ab^*a)^*+(ba^*b)^*$. **(5)**

Solution *Not equivalent.* For example, $aabb \in \mathscr{L}\Big((ab^*a+ba^*b)^*\Big)$, whereas $aabb \notin \mathscr{L}\Big((ab^*a)^*+(ba^*b)^*\Big)$.

2. Convert the following NFA to an equivalent DFA using the **subset-construction procedure** (*no credit for using any other method*). Take $\{a,b\}$ as the input alphabet. Draw the state-transition diagram of your DFA. Mark all unreachable states (that is, states inaccessible from the start state) in your DFA. **(10)**

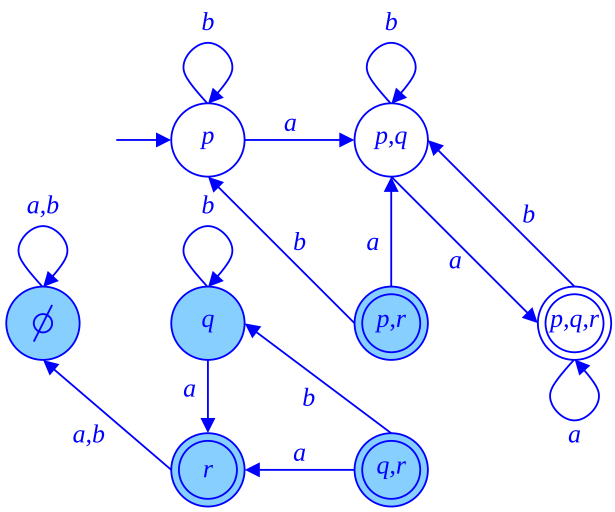


Solution The transition table for the DFA is first given.

State	Input symbol	Next state
$\emptyset$	a	$\emptyset$
$\emptyset$	b	$\emptyset$
$\{p\}$	a	$\{p,q\}$
$\{p\}$	b	$\{p\}$
$\{q\}$	a	$\{r\}$
$\{q\}$	b	$\{q\}$
$\{r\}$	a	$\emptyset$
$\{r\}$	b	$\emptyset$

State	Input symbol	Next state
$\{p,q\}$	a	$\{p,q,r\}$
$\{p,q\}$	b	$\{p,q\}$
$\{p,r\}$	a	$\{p,q\}$
$\{p,r\}$	b	$\{p\}$
$\{q,r\}$	a	$\{r\}$
$\{q,r\}$	b	$\{q\}$
$\{p,q,r\}$	a	$\{p,q,r\}$
$\{p,q,r\}$	b	$\{p,q\}$

The transition diagram is shown below. The unreachable states are shaded.



3. Consider the languages over the alphabet $\{a, b, c\}$, generated by the context-free grammars of the following two parts. For both these grammars, S is the start (and the only non-terminal) symbol. Which of these languages is/are regular? (*No credit without proper justification*)

(a) $S \to a \mid b \mid cSc$. **(5)**

Solution *Not regular.* The language of this grammar is $L_{(\mathrm{a})} = \{c^n a c^n \mid n \geqslant 0\} \cup \{c^n b c^n \mid n \geqslant 0\}$. Suppose that $L_{(\mathrm{a})}$ is regular, and let k be a pumping-lemma constant for $L_{(\mathrm{a})}$. Take $u = \varepsilon$, $v = c^k$, and $w = ac^k$. The pumping lemma gives the decomposition $v = xyz$ with $l = |y| > 0$ such that $uxy^izw \in L_{(\mathrm{a})}$ for all $i \geqslant 0$. Take $i = 2$, and note that $c^{k+l}ac^k \in L_{(\mathrm{a})}$, a contradiction.

(b) $S \rightarrow a \mid b \mid ScS$. **(5)**

Solution *Regular.* Repeated applications of the production $S \rightarrow cSc$ give sentential forms $ScScSc\ldots cS$. Eventually, all occurrences of S must vanish by becoming a or b. Therefore $L_{\text{(b)}} = \mathscr{L}(S)$ is the language of the regular expression $(a+b)(c(a+b))^*$.

4. (a) Design a context-free grammar for the language $\left\{a^i b^j c^k d^l \mid i,j,k,l \geqslant 0, \text{ and } i+k=j+l\right\}$ over the alphabet $\{a,b,c,d\}$. Clearly specify the role played by each non-terminal symbol of your grammar. (*No credit for writing only a grammar without any explanation*) **(7)**

Solution We consider two cases.

Case 1: $i \geqslant j$

Write $i = j+t$ for some $t \geqslant 0$. Since $i+k = j+l$, this implies $l = k+t$, that is, $a^i b^j c^k d^l = a^t a^j b^j c^k d^k d^t$, where $j,k,t \geqslant 0$ are independent.

Case 2: $j \geqslant i$

Write $j = i+t$ for some $t \geqslant 0$. Since $i+k = j+l$, this implies $k = l+t$, that is, $a^i b^j c^k d^l = a^i b^i b^t c^t c^l d^l$, where $i,l,t \geqslant 0$ are independent.

In view of this, we first introduce three new non-terminal symbols and the following productions for them.

$$
\begin{array}{lcll}
U & \rightarrow & \varepsilon \mid aUb & \text{[Generates } a^n b^n \text{ for } n \geqslant 0] \\
V & \rightarrow & \varepsilon \mid bVc & \text{[Generates } b^n c^n \text{ for } n \geqslant 0] \\
W & \rightarrow & \varepsilon \mid cWd & \text{[Generates } c^n d^n \text{ for } n \geqslant 0]
\end{array}
$$

If S is the start symbol, Case 2 is easy to handle:

$$S \rightarrow UVW$$

In Case 1, S generates equally many a's and d's at the two ends, and then converts to T to generate the central part.

$$
\begin{array}{lcl}
S & \rightarrow & aSd \mid T \\
T & \rightarrow & UW
\end{array}
$$

Note: We can rephrase Case 2 to have $j > i$ (in order to avoid ambiguity). In this case, $t > 0$, so the productions for V would be $V \rightarrow bc \mid bVc$.

(b) Show **leftmost derivations** of the strings *ad*, *bc*, and *abcd* by your grammar. **(3)**

Solution

$S \to aSd \to aTd \to aUWd \to aWd \to ad$

$S \to UVW \to VW \to bVcW \to bcW \to bc$

$S \to T \to UW \to aUbW \to abW \to abcWd \to abcd$ [Using Case 1]

$S \to UVW \to aUbVW \to abVW \to abW \to abcWd \to abcd$ [Using Case 2]

5. In the following context-free grammar, the set of terminal symbols is $\Sigma = \{a, b, c\}$, the set of non-terminal symbols is $N = \{S, U, V\}$, and the start symbol is S. Convert the grammar to the Chomsky normal form. Show all the steps of your conversion. **(10)**

$$\begin{array}{lcl} S & \rightarrow & U \mid V \\ U & \rightarrow & \varepsilon \mid c \mid aVa \\ V & \rightarrow & \varepsilon \mid c \mid bUb \end{array}$$

Solution To get rid of the unit productions $S \rightarrow U$ and $S \rightarrow V$, we introduce the new productions:

$$S \rightarrow \varepsilon \mid c \mid aVa \mid bUb$$

Now, in order to avoid the ε productions $U \rightarrow \varepsilon$ and $V \rightarrow \varepsilon$, we introduce the new productions:

$$\begin{array}{lcl} U & \rightarrow & aa \\ V & \rightarrow & bb \end{array}$$

S does not appear on the right side of any production, so the production $S \rightarrow \varepsilon$ introduces no new productions.

Now, we are ready to throw out all unit and ε productions. Thus, our grammar is now of the form:

$$\begin{array}{lcl} S & \rightarrow & c \mid aVa \mid bUb \\ U & \rightarrow & c \mid aa \mid aVa \\ V & \rightarrow & c \mid bb \mid bUb \end{array}$$

Since c does not appear with any other symbol on the right side of any production, it suffices to introduce new non-terminal symbols A and B for a and b only. This gives us:

$$\begin{array}{lcl} S & \rightarrow & c \mid AVA \mid BUB \\ U & \rightarrow & c \mid AA \mid AVA \\ V & \rightarrow & c \mid BB \mid BUB \\ A & \rightarrow & a \\ B & \rightarrow & b \end{array}$$

Finally, the productions with three symbols on the right side are handled.

$$\begin{array}{lcl} S & \rightarrow & c \mid AX \mid BY \\ U & \rightarrow & c \mid AA \mid AX \\ V & \rightarrow & c \mid BB \mid BY \\ X & \rightarrow & VA \\ Y & \rightarrow & UB \\ A & \rightarrow & a \\ B & \rightarrow & b \end{array}$$

6. Use the **pumping lemma** to prove that the language $\left\{x\#y \mid x,y \in \{a,b\}^*, \text{ and } x \text{ is a substring of } y\right\}$ over the alphabet $\{a,b,\#\}$ is not context-free. **(10)**

Solution Suppose that the language—call it L_6—is regular. Let k be a pumping-lemma constant for L_6. Consider the string $z = a^k b^k \# a^k b^k \in L_6$. Since $|z| \geqslant k$, the pumping lemma for context-free languages supplies a decomposition $z = uvwxy$ with the following conditions:

(i) $|vwx| \leqslant k$,
(ii) $vx \neq \varepsilon$, and
(iii) $z_i = uv^i wx^i z \in L_6$ for all $i \geqslant 0$.

Let us call the blocks of a and b in z (or z_i) A_L, B_L, A_R, B_R (from left to right). Now, consider several cases.

***v* or *x* contains #:** In this case, z_i contains a wrong number of #'s for all $i \neq 1$.

Both *v* and *x* belong to the left of #: Now, z_2 contains more symbols to the left of # than to the right of #, so the string to the left of # in z_2 cannot be a substring of the string to the right of # in z_2.

Both *v* and *x* belong to the right of #: Consider z_0, and the symbol counts again indicate that the string to the left of # in z_0 cannot be a substring of the string to the right of # in z_0.

***v* is to the left of #, and *x* is to the right of #:** By Condition (i), v and x cannot belong to non-consecutive blocks, so v must belong to B_L, and x to A_R. By Condition (ii), either v or x is non-empty (or both are). If v is non-empty, consider z_2. If x is non-empty, consider z_0.

In all the cases, we see that either z_2 or z_0 does not belong to L_6, a contradiction to Condition (iii).