INDIAN INSTITUTE OF TECHNOLOGY, KHARAGPUR

Date .... FN / AN *Time:* 2/3 Hrs. *Full Marks* .... *No. of Students* ....
Autumn / Spring Semester, 20 .... Deptt. .... Sub No. ....
...... Yr. B. Tech.(Hons.) / B. Arch. / M. Sc. Sub. Name ....

Instruction : **Answer Q1 and any Four(4) from the Remaining Questions.**

1. Answer with a short justification whether the following **claims** are **true** or **false**. No credit will be given for writing only *true* or *false*. Assume the alphabet $\Sigma = \{0, 1\}$ unless specified otherwise. $<A>$ means an encoding of the object A. CFL (CFG): context-free language (grammar), PDA: push-down automaton, r.e.: recursively enumerable, RE_Σ: the collection of all r.e. languages over Σ.

   (a) Claim: A deterministic finite state transition system over Σ with n states (the start state is fixed) can accept 2^n different languages for different choices of the set of final states, and these languages form a Boolean algebra.

   True: The set Σ^* is partitioned by the DFA $M = (Q, \Sigma, \delta, q_0, F)$ in $|Q| = n$ equivalence classes (Myhill-Nerode), where every state corresponds to an equivalence class - two strings $x, y \in \Sigma^*$ are said to be related (binary relation $\equiv_M$) if $\delta(q_0, x) = \delta(q_0, y)$. This can be shown to be an equivalence relation. If $F = Q$, Σ^* is accepted, if $F = \emptyset$, $\emptyset$ is accepted. Let $A, B \subseteq Q$, L_A or L_B are the languages accepted when $F = A$ or $F = B$ respectively. If $F = A \cup B$, the language accepted is $L_A \cup L_B$, if $F = A \cap B$, the language accepted is $L_A \cap L_B$, if $F = Q \setminus A$, the language accepted is $\Sigma^* \setminus L_A$. So is the claim.

   (b) Claim: There are only finite number of unambiguous CFGs for the language $L = \{0^n 1^n : n \geq 1\}$.

   False: There are unambiguous CFG's for every $k = 1, 2, 3, \cdots$. The production rules of the k^{th} CFG (k is fixed) are:

   $$S \to 01 \mid 0011 \mid \cdots \mid 0^k 1^k \mid 0^k S 1^k.$$

   Taking $k = 3$, the rulwes are $S \to 01 \mid 0011 \mid 000111 \mid 000S111$.

   (c) Let $\mathcal{C} = \{L \subseteq \Sigma^* : L \text{ is } \textit{co-finite}\}$.
   Claim: Each element of $\mathcal{C}$ is a CFL and intersections of any two of them is also a CFL.

   True: Each language $L \in \mathcal{C}$ is co-finite, so it is a regular language, and so it is CFL. If $L_1, L_1 \in \mathcal{C}$, then $L_1 \cap L_2$ is also a regular language, and so it is a CFL.

   (d) L is a CFL, $x \in L$, and a *proper prefix* of x is also in L.
   Claim: L cannot be accepted by a *deterministic push-down automaton* (DPDA) in *empty stack*.

   True: Let $x = uv \in L$, $u \in L$ and $v = \varepsilon$. The DPDA will empty the stack while accepting the string u and cannot make any move. But it is suppose to compute on the remaining portion of the string. As it is a DPDA, there is no other choice.

(e) Claim: If $L = \{1^p : p \text{ is a prime}\}$, then there is no *context-sensitive language* L' so that LL' is a *regular* language.

False: Take $L' = \{1^n : n \geq 1\}$, which is regular and so is context-sensitive. $LL' = \{1^n : n \geq 3\}$ is clearly regular.

(f) Claim: If L is a CFL and $x \in L$ is of length greater than or equal to the *pumping constant*, the number of strings of L is infinite.

True: Let the pumping constant be k and the string be w. By pumping theorem we can write $w = uvxyz$, so that $|vy| > 0$, $|vxy| \leq l$, and for each $i \geq 0$, $uv^ixy^iz \in L$. So the number of strings in the language are infinite.

(g) Claim: The collection of *decidable* or *recursive* languages over Σ is a Boolean algebra with countably infinite number of elements.

True: Recursive languages cannot supersede countability as they are decided by Turing machines. Both Σ^* and $\emptyset$ are recursive (they are regular). This class is also closed under union, intersection and completation. So is the claim.

(h) Claim: The length of encoding (using $0, 1$) of a *deterministic Turing machine* over $\{0, 1\}$, with the tape alphabet $\{0, 1, \not{b}\}$, and the number of states n, is $O(n)$.

False: We cannot go for binary encoding as 0 is to be used as separator. So $Q = \{1, 11, 111, \cdots, \overbrace{111\cdots 1}^{n}\}$ is of length $O(n^2)$.

(i) Claim: Every *r.e.* language over $\{0, 1\}$ is not reducible to $L_{HALT} = \{< M, w >: M \text{ is a Turing machine that halts on input } x\}$.

False: Every r.e. language is mapping-reducible to L_{HALT}. Let L be a r.e. language recognised by a Turing machine M. We can always modify M is such a way that it halts only when it accepts a string, otherwise it runs forever. Let us call this machine to be M_L. Now the reduction mapping f (Turing computable) is $x \mapsto < M_L, x >$. Note that M is known and the Turing machine that computes f modifies it.
It is clear that $x \in L$ if and only if $< M_L, x > \in L_{HALT}$.

[9 × 3]

2. (a) Use pumping theorem to prove that $L = \{0^p1^q : p + q \text{ is } \underline{not} \text{ a perfect square}\}$ is not a regular language.

Ans. Let $\overline{L} = \{0^p1^q : p+q \text{ is a perfect square}\}$ is regular and the pumping constant be k. Naturally $0^{k^2} \in \overline{L}$ and by pumping theorem we can write $0^{k^2} = xyz$ so that (i) $|y| > 0$, (ii) $|xy| \leq k$, (iii) for all $i \geq 0$, $xy^iz \in \overline{L}$. Let $y = 0^l$, where $l \leq k$. Now $xy^2z = 0^{k^2+l} \in \overline{L}$. But then the next similar string is $0^{(k+1)^2} = 0^{k^2+2k+1}$. $l \neq 0^{2k+1}$ as $l \leq k$. So it is impossible for $\overline{L}$ to be regular. Now, $\overline{L} = (\Sigma^* \setminus L) \cap L(0^*1^*)$. L is regular implies $\overline{L}$ is regular - a contradiction.

(b) Use Myhill-Nerode theorem and other closure properties of regular languages to show that $L = \{0^m1^n : \text{hcf}(m, n) > 1\}$ is not regular.

Ans. If possible $\overline{L} = \{0^p1^q : \text{hcf}(m, n) = 1\}$ is accepted by the DFA $M = (Q, \{0, 1\}, \delta, q_0, F)$.
But we claim that 0^p and 0^q, where p, q are two distinct primes can not be equivalent i.e. $\delta(q_0, 0^p) \neq \delta(q_0, 0^q)$. Otherwise, $0^p1^q \in \overline{L}$ implies that $0^q1^q \in \overline{L}$ which is not the case.

But then there are infinite number of primes, so it is not possible to have a DFA. $\overline{L} = (\Sigma^* \setminus L) \cap L(0^*1^*)$. If L is regular, then so is $\overline{L}$ - contradiction. So L is not regular.

[6 + 6]

3. (a) Design a PDA (state transition diagram) that recognises the language $L = \{x \in \{0,1\}^* : x \neq ww\}$. \$ is the bottom marker of the stack.

Ans. Design the CFG with the start symbol \$ and use standard construction of one state PDA. The CFG $G = (\{\$, A, B, D\}, \{0, 1\}, P, \$)$, where profuction rules are

$$\begin{aligned} \$ &\rightarrow AB \mid BA \mid A \mid B \\ A &\rightarrow DAD \mid 0 \\ B &\rightarrow DBD \mid 1 \\ D &\rightarrow 0 \mid 1 \end{aligned}$$

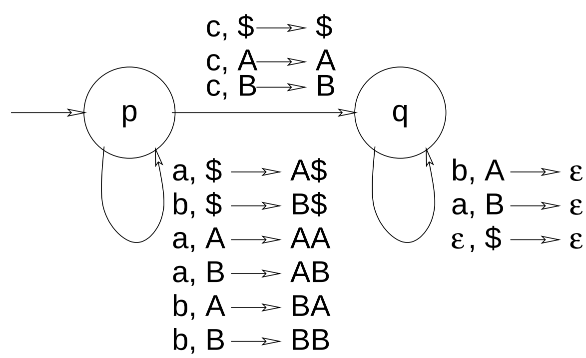
The corresponding PDA is $M = (\{q\}, \{0, 1\}, \{0, 1, \$, A, B, D\}, \delta, q, \$, \emptyset)$. The transition functions are,

i. If $A \rightarrow \alpha \in P$, then $(q, \alpha) \in \delta(q, \varepsilon, A)$ e.g. $\delta(q, \varepsilon, A) = \{(q, DAD), (q, 0)\}$.

ii. $\delta(q, 0, 0) = \{(q, \varepsilon)\}$ and $\delta(q, 1, 1) = \{(q, \varepsilon)\}$.

(b) Use pumping theorem to prove that $L = \{x \in \{0,1\}^* : x = ww\}$ is not a context-free language.

Ans. Let L be context-free and k is the pumping constant. We consider the string $1^k0^k1^k0^k \in L$ and argue that pumping is impossible. **[7 + 5]**



4. (a) Consider the *push-down automaton* $P = (\{p, q\}, \{a, b, c\}, \{\$, A, B\}, \delta, p, \$, \phi)$ and <u>*formally*</u> construct an equivalent *context-free grammar*. The acceptance is by *empty-stack*. Clearly explain the non-terminals and the production rules.

Ans. The meaningful non-terminals and productions of the grammar are $N = \{S, (p\$q), (pAq), (pBq), (q\$q), (qAq), (qBq)\}$ and

$$\begin{aligned} S &\rightarrow (p\$q) \\ (p\$q) &\rightarrow a(pAq)(q\$q) \mid b(pBq)(q\$q) \mid c(q\$q) \\ (pAq) &\rightarrow a(pAq)(qAq) \mid b(pBq)(qAq) \mid c(qAq) \\ (pBq) &\rightarrow a(pAq)(qBq) \mid b(pBq)(qBq) \mid c(qBq) \\ (qAq) &\rightarrow b \\ (qBq) &\rightarrow a \\ (q\$q) &\rightarrow \varepsilon \end{aligned}$$

If we give 'better' names to the non-terminals, we get:

$$\begin{array}{rcl} S & \to & S' \\ S' & \to & aAD \mid bBD \mid cD \\ A & \to & aAX \mid bBX \mid cX \\ B & \to & aAY \mid bBY \mid cY \\ X & \to & b \\ Y & \to & a \\ D & \to & \varepsilon \end{array}$$

(b) Let L be a prefix closed infinite context-free language. Prove that there is an infinite regular language $L' \subseteq L$.

Ans. Let the pumping constant be k. Consider a string $w \in L$ of length $\geq k$. We can divide the string as $w = uvxyz$ so that (i) $|vxy| \leq k$, (ii) $|vy| > 0$, and for all $i \geq 0$, $uv^ixy^iz \in L$.

But then the language is prefix-closed, so for all $i \geq 0$, (i) $uv^i \in L$, if $v \neq \varepsilon$, or (ii) $uxy^i \in L$, if $v = \varepsilon$. In the first case the regular subset is uv^* an in the second case the regular subset is $uxy*$.

[8 + 4]

5. (a) Prove that $L_d = \{x_i$: the Turing machine M_i does not accept $x_i\}$ is not Turing recognisable.

Ans. If L_d is Turing recognisable, there is a Turing machine M_d that recognises L_d. Let the encoding of the Turing machine be the k^{th} string i.e. $< M_d >= x_k$ and M_d is M_k in our enumeration.

The question is whether $x_k \overset{?}{\in} L_d$. If $x_k \in L_d$, then $M_d = M_k$ does not accept x_k i.e. $x_k \notin L_d$. If $x_k \notin L_d$, then $M_d = M_k$ does not accept x_k i.e. $x_k \in L_d$. So, $x_k \in L_d$ if and only if $x_k \notin L_d$ - a contradiction. So L_d is not Turing recognisable.

(b) Prove that $L_\emptyset = \{< M >: M$ is a Turing machine and $L(M) = \emptyset\}$ *mapping reducible* to $L_= = \{< M_1, M_2 >: M_1$ and M_2 are equivalent Turing machines$\}$ as well $L_{\neq} = \{< M_1, M_2 >: M_1$ and M_2 are not equivalent Turing machines$\}$.
What is your conclusion from this result?

Ans. We construct a Turing machine M_e so that $L(M_e) = \emptyset$. The mapping reduction is $< M > \mapsto < M, M_e >$. It is clear that $< M > \in L_\emptyset$ if and only if $< M, M_e > \in L_=$.

The answer to the remaining portion is not known to me!!

(c) Prove that the universal language $L_u = \{< M, x >:$ the Turing machine M accepts $x\}$ is not *mapping reducible* to $L_\emptyset$.

Ans. We know that $\overline{L_\emptyset}$ is recursively enumerable. Following Turing machine recognises this language.
$M_{\overline{\emptyset}}$:
Input: y

i. If $y \neq < M >$, reject y as such a string by definition encodes a machine whose language is empty,

ii. for $i = 1, 2, 3, \cdots$ do the following steps

iii. enumerate $x_1, x_2, \cdots, x_i \in \Sigma^*$

iv. Simulate M $(< M >= y)$ on each of these strings for i steps.

v. If one of the simulation on some x_k comes to accept halt, accept $y =< M >$ as $x_k \in L(M) \neq \emptyset$.

If $L_u \leq_m L_\emptyset$, $\overline{L_u} \leq_m \overline{L_\emptyset}$. But we know that $\overline{L_\emptyset}$ is recursively enumerable. That makes $\overline{L_u}$ recursively enumerable - but that is impossible.

[3 + 6 + 3]

6. Give proper justification for the following statements.

(a) Context-free languages are closed under inverse-homomorphism.

Ans. Let $h : \Sigma_1^* \to \Gamma_2^*$ be a homomorphism and L be a CFL over Σ_2. The claim is that $h^{-1}(L) \subseteq \Sigma^*$ is a CFL.

Let the PDA $M_2 = (Q_2, \Sigma_2, \Gamma_2, \delta_2, q_{20}, \$, F_2)$ recognises L_2. The PDA $M_1 = (Q_1, \Sigma_1, \Gamma_1, \delta_1, q_{10}, \$, F_1)$ first translates the input from Σ_1 to its homomorphic image and then runs M_2 on it. So the whole input over Σ_1^* is translated to its image over Σ_2^* and M_2 is run on it. If M_2 accepts, then so is M_1.

The state set takes care of the translation:

- $Q_1 = Q_2 \times \{x \in \Sigma_2^* : x \text{ is a suffix of } h(a) \text{ for all } a \in \Sigma_1\}$. The second component of the state keeps the translation.
- $q_{10} = (q_{20}, \varepsilon)$.
- $\Sigma_1 = \Sigma_2$, $\Gamma_1 = \Gamma_2$.
- $F_1 = F_2 \times \{\varepsilon\}$.

The translation takes place as follows: $\delta_1((p, \varepsilon), a, X) = \{(p, h(a)), X)\}$.

(b) If L_1 and L_2 are recognised by deterministic Turing machines (DTMs) M_1 and M_2, then there is a DTM that recognises L_1L_2.

Ans. The DTM for L_1L_2 works as follows:

M :

Input: x

i. Split the input x in two parts $x = x_1x_2$ in all possible ways. If the length of x is n, there are $n + 1$ split.

ii. for $i = 1, 2, 3, 4, \cdots$ do the following steps.

iii. Simulate M_1 on the first part, x_1, and simulate M_2 on the second part x_2 for i steps.

iv. If both the machines accept some split within some i steps, accept x.

(c) Any *context-free language* over a one-letter alphabet is a regular language.

Ans. If the language L is finite, there is nothing to prove.

Consier an infinite language L and let the pumping constant be k. So each w of length $\geq k$ in the language can be written as $w = uvxyz$ so that $|vxy| \leq k$, $|vy| > 0$ and for all $i \geq 0$, $uv^ixy^iz \in L$. As there is only one alphabet, we write the last

clause as $uxz(vy)^* \subseteq L$. $|vy| = p$, so $uxz(a^p)^i \in L$ for all $i \geq 0$.
Let $\alpha = k!$, we calim that $w(a^\alpha)^m \in L$ for all $m \geq 0$, as $\alpha \times m = p \times \frac{m \times \alpha}{p}$. Note that α does not depend on w.
So for each word $w \in L$ and $|w| \geq k$, $w(a^\alpha)^m \in L$, for all $m \geq 0$.
We observe that each $w \in L$ and $|w| \geq k$ is an element of $a^{k+i}(a^\alpha)^*$ where $0 \leq i < \alpha$. Consider the least element w_i of $L \cap a^{k+i}(a^\alpha)^*$. So the language $L = L_1 \cup L_2$, where L_1 is the finite collection of strings of length $< k$ and $L_2 = \bigcup_{0 \leq i < \alpha} w_i(a^\alpha)^*$. So L is regular. **[3 × 4]**

Sig.of the Paper-Setter ....