Theory of Computation(60-354)
Sample Final
Total Marks : 90
Time: 3 hours

Instructions:

- This test is open-book (course text-book only), open-notes.
- There are a total of 6 questions. Answer all of them.

Name: **SID:**

Qn.1 We showed that PCP is undecidable for an arbitrary alphabet Λ. Show that PCP is undecidable even if we restrict the alphabet to $\Sigma = \{0, 1\}$ by reducing PCP to this special case of PCP. (*Hint*: Consider fixed-length encoding of symbols in Λ over Σ).

[15 marks]

Ans:

Let Λ have n symbols. We can encode each one of these over Σ using $\lceil(\log_2(n))\rceil$ bits.

If w_i and $x_i, i = 1, ..., k$ is a PCP instance over Λ, we reduce this to a PCP instance over w_i' and ${x_i}', i = 1, ..., k$ by replacing each pair (w_i, x_i) by the corresponding pair (w_i', x_i'), where each w_i' (resp x_i') is obtained from $w_i(x_i)$ by using the above encoding scheme so that the w_i's and the x_i's are strings over Σ.

Thus if there is a solution $i_1, i_2, ..., i_l$ to the PCP instance over Λ iff there is a solution to the corresponding PCP instance over Σ.

Qn.2 A balanced parentheses sequence over $\Sigma = \{(,)\}$ is one in which every closing parenthesis) is matched by the closest unmatched opening parenthesis (to its left. For example, $(), (()())$ are balanced parentheses sequences. Use this definition to design a Turing machine to accept a balanced parentheses sequence. Provide explanations of the different states that you introduce and what their functions are.

[15 marks]

Ans: The algorithm is this: For each closed parenthesis, ")", we find the nearest open parenthesis, "(", to its left. We do this in a loop as we find the closed parentheses in a left to right scan. If for a given closed parenthesis we cannot find an open parenthesis to its left then the parentheses sequence is not balanced.

The TM we design has three states: a start state q_0, two left-moving states $q_)$, q_B and a final state q_f.

The transition table is this.

	(	)	X	B
q_0	$(q_0, (, R)$	$(q_), X, L)$	(q_0, X, R)	(q_B, B, L)
$q_)$	(q_0, X, R)		$(q_), X, L)$	
q_B			(q_B, X, L)	(q_f, B, R)

Qn.3 Show that the languages

$L_1 = \{0^n 1^m 2^{2m} | n, m \geq 0\}$ and $L_2 = \{0^n 1^{2n} 2^m | n, m \geq 0\}$

over $\Sigma = \{0, 1, 2\}$ are context-free by generating context-free grammars for these.

What is $L = L_1 \cap L_2$? Can you prove that L is context-free by using the pumping-lemma for CFLs ? Justify your answer.

[15 marks]

Ans:

L_1 ia generated by the grammar:

$S \to AB$
$B \to 1B22 | \epsilon$
$A \to 0A | \epsilon$

L_2 ia generated by the grammar:

$S \to AB$
$S \to 0A11 | \epsilon$
$B \to 2B | \epsilon$

$L = L_1 \cap L_2 = \{0^i 1^{2i} 2^{4i} | i \geq 0\}$

L is not context-free.

Let $z = 0^n 1^{2n} 2^{4n}$, where n is the PL constant. Let $z = uvwxy$ be an adversarial decomposition.

1. Case 1: vwx spans $0^n 1^{2n}$. Setting $i = 0$ in $uv^i wx^i y$, gives us uwx which has fewer $0's$ or $1's$ as a result of which the 1:2:4 ratio of the 0's, 1's and 2's is not maintained.
2. Case 1: vwx spans $1^2 n 2^{4n}$. Setting $i = 0$ in $uv^i wx^i y$, gives us uwx which has fewer $1's$ or $2's$ as a result of which the 1:2:4 ratio of the 0's, 1's and 2's is not maintained.

Qn.4 Give an *informal* argument as to why the language $L(G)$ generated following grammar G

$$S \to 0|0S|1SS|S1S|SS1$$

is contained in the language $L = \{w \in \{0,1\}^* | \#0's > \#1's\}$ (It is a lot more difficult to show that every string in L can be generated by this grammar).

Ans:

Clearly, the string 0 satisfies the property. If S generates a string w with the property, then the string $0w$ has the property too. If S generates strings x and y with the property then each of the strings $1xy$, $x1y$, $xy1$ has the same property, since in each of the cases x and y between them generates at least 2 more 0's than 1's.

The moves of the PDA derived from the above grammar is :

$\delta(q, \epsilon, S) = \{(q, 0), (q, 0S), (q, 1SS), (q, S1S), (q, SS1)\}$

$\delta(q, 0, 0) = \{(q, \epsilon)\}$

$\delta(q, 1, 1) = \{(q, \epsilon)\}$

Qn.5 Define a suitable homomorphism from $\{a, b, c\}^* \to \{0, 1\}^*$ to show that the language $L = \{a^n b^k c^{n+k} | n \geq 0, k \geq 0\}$ is not regular.

Also, give an alternate proof using the pumping-lemma for regular languages.

[15 marks]

Ans: Let $h : \{a, b, c\}^* \to \{0, 1\}^*$ be the homomorphism defined by $h(a) = 0$, $h(b) = 0$, $h(c) = 1$. Then $h(L) = \{0^{n+k} 1^{n+k} | k \geq 0, n \geq 0\} = \{0^m 1^m | m \geq 0\}$, letting $m = n + k$.

Since $h(L)$ is not regular and a homomorphism preserves regularity, L is not regular.

For the alternate argument choose the string $z = a^n b^n c^{2n}$, where n is the pumping lemma constant.

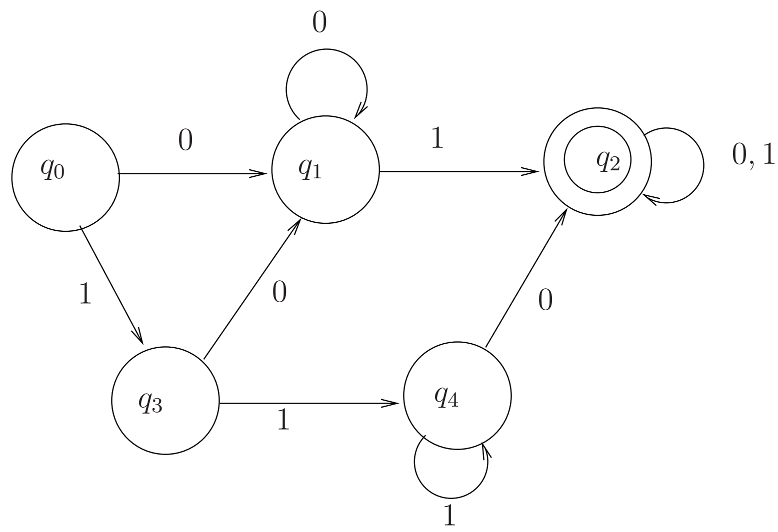

Figure 1: *DFA for Qn. 6*

Qn.6 Let L be a language over $\Sigma = \{0, 1\}$, that consists of all strings containing the substring 01 or 110; construct a regular expression $\mathbf{R}$ corresponding to L. Also, design a DFA that accepts L. Briefly explain your design. (*Hint*: For the DFA construction try a direct approach).

[15 marks]

Ans: The regular expression is : $(0{+}1)^*(01{+}110)(0{+}1)^*$. A DFA is shown in the figure below.