COSC/MATH 4P61 - Theory of Computation

Example Questions Test 3

Question 1: Use the Pumping Lemma to show that the language

$$L = \{a^i b^j c^k \mid i < j < k\}$$

is not context-free.
Hint: Use the word $a^n b^{n+1} c^{n+2}$ *where* $n \geq 2$ *is the constant of the Pumping Lemma and distinguish the cases that* vwx *contains or does not contain* c*.*

Question 2: Consider the Turing machine

$$M = (\{q_1, q_2, q_3\}, \{0, 1\}, \{0, 1, B\}, \delta, q_0, B, \{q_2\})$$

with $\delta(q_1, 0) = (q_3, 1, R)$, $\delta(q_3, 1) = (q_1, 0, R)$ and $\delta(q_3, B) = (q_2, B, R)$. What is the language $L(M)$ accepted by M? Justify your answer.

Question 3: Construct a Turing machine that adds two numbers, i.e., it transforms an initial tape of the form $0^m 1 0^n$ (m and n 0's separated by a 1) into 0^{m+n} ($m + n$ 0's, no 1). The tape head should be at the left-most 0 before and after the computation. Run your machine on the input 0010.
Hint: Make sure that you consider the cases $m = 0$ *and* $n = 0$*.*

Question 4: Write a possible code s for the Turing machine in Question 2. What is the number of this string, i.e., for which n is $w_n = s$? (You can provide the number in binary)

Question 5: Consider the language

$$L = \{0w \mid w \in L_u\} \cup \{1w \mid w \notin L_u\}.$$

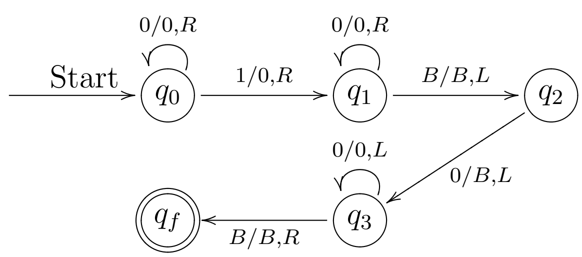
Is this language recursively enumerable? Justify your answer.

Solutions

Question 1: Assume L is context-free and let $n \geq 2$ be the constant from the Pumping Lemma. Pick the word $z = a^n b^{n+1} c^{n+2}$. Using the Pumping Lemma we can write z as $z = uvwxy$ with $|vwx| \leq n$ and $vx \neq \epsilon$. If vwx does not have a c, then uv^3wx^3y has at least $3n > n+2$ a's or $b's$ and is, therefore, not in L. If vwx has a c, then it cannot have an a because $|vwx| \leq n$. In this case the word uwy has n a's and at most $(n+1)+(n+2)-1 = 2n+2$ b's and c's showing that $uwy \notin L$.

Question 2: The language accepted by this Turing machine is given by the regular expression (01)*0. The machine moves in every step of its computation to the right. It only accepts a word if it is in state q_3 and sees a blank. The only way of getting in state q_3 is by reading a 0 in state q_1. Therefore, the last symbol of the input must be a 0. The machine can get from q_1 to q_1 only by reading 01, which can be repeated.

Question 3: We define $M = (\{q_0, q_1, q_2, q_3, q_f\}, \{0, 1\}, \{0, 1, B\}, \delta, q_0, B, \{q_f\})$ by:



Tape						State
_	0	0	1	0	_	q_0
	↑					
_	0	0	1	0	_	q_0
		↑				
_	0	0	1	0	_	q_0
			↑			
_	0	0	0	0	_	q_1
				↑		
_	0	0	0	0	_	q_1
					↑	
_	0	0	0	0	_	q_2
				↑		
_	0	0	0	_	_	q_3
			↑			
_	0	0	0	_	_	q_3
		↑				
_	0	0	0	_	_	q_3
	↑					
_	0	0	0	_	_	q_3
↑						
_	0	0	0	_	_	q_f
	↑					

Question 4: The three instruction can be encoded by:

$$\delta(q_1, 0) = (q_3, 1, R) \qquad 0101000100100$$
$$\delta(q_3, 1) = (q_1, 0, R) \qquad 0001001010100$$
$$\delta(q_3, B) = (q_2, B, R) \qquad 00010001001000100$$

so that

01010001001001100010010101001100010001001000100

is a possible code for M. This string has number

101010001001001100010010101001100010001001000100 in binary.

Question 5: Suppose L would be recursively enumerable, i.e., there is a Turing machine M that accepts L. Then we can construct a Turing machine M' that accepts $\overline{L_u}$ as follows. Given an input w, M' modifies the input to $1w$ and then simulates M. If M accepts, then w is in $\overline{L_u}$, and our machine M' accepts as well. If M does not accept $1w$ or runs forever, then M' does the same. The existence of M' is a contradiction to the fact that L_u is recursively enumerable but not recursive, i.e., $\overline{L_u}$ is not recursively enumerable.