

CS 3240 : Languages and Computation Practice Test : Finals

Question I: Design a Turing Machine that will pick the greater of two positive integer numbers which are represented by a sequence of 0's. Consider the following example:

0000 # **00000** # **00000**

num1 = 4 num2 =5 result=5

1. Give the Tape Alphabet, meaning the set of all characters appearing on the TM's tape (include any character that you might write on the tape)
2. Design the state diagram of the Turing Machine.

Question II : Design a Turing Machine (including tape alphabet, state transition diagram) to recognize the language $L = \{a^n b^n c^n, n > 0\}$

Question III : Let's A be the language over $\{ (,) \}$ where the number of “(“ is equal to the number of “)”. Give a deterministic PDA that will recognize A

Question IV. Let's A be the language over $\{0,1\}$ where the 0's and the 1's can occur in any order and the number of 0's is greater than the number of 1's

1. Is language A acceptable a Non-deterministic PDA? If it is then provide a NPDA otherwise reason why not.
2. Is A acceptable by a Turing Machine? If it is give a TM otherwise reason why not.

Question V A given grammar is found to be LL(2) ie, one token lookahead is not sufficient. Answer the following:

0. First give an example of a small grammar that is not LL(1) but is LL(2).
1. How will you modify the LL(1) parsing algorithm for 2 token lookahead?
2. In order to support the parsing algorithm in (1), how will you redefine the parsing table?
3. In order to generate the parsing table in (2) how will you extend the concept of First() and Follow() sets? Define the new sets and show how will you compute them using the grammar of part (0).
4. Show the parsing table using 2-token look-ahead and show a sample parse on some input.