CSE 2001: Introduction to the theory of computation
Assignment 4 – Solutions

Question 1

Describe a Turing Machine that accepts (decides) the non-context-free language $\{a^n b^n c^n | n \geq 0\}$. While you need not draw a state diagram, you should describe the machine in detail.
Solution: The strategy is very simple. The TM crosses off each a and matching b, cs. If at some point matching b or c's are not found, it enters the reject state and halts. When all the a's are crossed off it scans right to check for uncrossed b or c's. If none are found, it halts in the accept state, else it halts in the reject state.

Question 2

Suppose Turing Machines T_1, T_2 compute the functions f_1, f_2 respectively. Describe how to construct a Turing Machine that computes the function $f_1 + f_2$.
Solution: The basic idea is : given an input x, the machine first simulates f_1 and writes down $f_1(x)$ and then it simulates f_2 and writes down $f_2(x)$. Finally, it adds these two numbers to generate the final output. Some details need to be provided,
We will use a 5 tape machine for this problem. Assume that the machine descriptions for T_1, T_2 are given on tapes 2, 3, respectively (if not copy them to these two tapes as a first step) and that the input x is given on tape 1. Then simulate T_1 on x and write the output on tape 4. Simulate T_2 on x and write the output on tape 5. The final step is addition of the numbers on tapes 4 and 5. If the numbers are given in unary, we have seen in class how to add them. If they are given in binary or decimal, use the algorithm for longhand addition taught in elementary school. Showing that addition can be done by TM's is trivial.

Question 3

For each of the following decision problems, state whether the problem is decidable and prove your answer.

1. Given a TM T are there any input strings on which T loops forever?

   Solution: Let us assume that this language is decidable. Then there exists a TM R that decides this language. Suppose R accepts if the input machine never loops forever and rejects otherwise.

   We reduce the acceptance problem to this problem. Given an input $\langle M, w\rangle$ for the acceptance problem, we build a new TM M_1 as follows:

   M_1: On input x,
   1. if $x = w$, accept if M accepts w, loop forever otherwise.
   2. Else if $x \neq w$ reject.

   Clearly this machine halts on all inputs if M accepts w and loops forever on input w if M rejects w.

Feed M_1 as input to R. If R rejects M_1 then we know that M rejects w. Otherwise M accepts w. Thus we have built a TM that decides the acceptance problem. This is a contradiction. Therefore the given problem is undecidable.

2. Given a TM T are there any input strings not accepted by T?

   Solution: Let us assume that this language is decidable. Then there exists a TM R' that decides this language. Suppose R' accepts if the input machine accepts all strings and rejects otherwise.

   We reduce the acceptance problem to this problem. Given an input $\langle M, w \rangle$ for the acceptance problem, we build a new TM M_2 as follows:

   M_2: On input x,
   1. if $x = w$, accept if M accepts w, reject otherwise.
   2. Else if $x \neq w$ accept.

   Clearly this machine accepts all inputs if M accepts w and rejects w if M rejects w.

   Feed M_2 as input to R'. If R' rejects M_2 then we know that M rejects w. Otherwise M accepts w. Thus we have built a TM that decides the acceptance problem. This is a contradiction. Therefore the given problem is undecidable.