Assignment 4

SE4I03

due on Nov 30 (Friday)
Each question 10 marks, total 30 marks

1. Question 3 on Test 2 asked for an informal description of a Turing Machine that recognizes the language L_{PAL}. Now give a formal description of your[1] Turing Machine: call this Turing Machine M_{PAL}, and let $M_{\text{PAL}} = (Q \cup \{q_0, q_{\text{accept}}, q_{\text{reject}}\}, \Sigma, \Gamma, \delta)$. Specify the following:

   $Q =$ ____________________________________

   $\Sigma = \{0, 1\}$ (nothing to do here)

   $\Gamma = \Sigma \cup$ ____________________________________

   and finally provide the table for the transition function δ:

	state	symbol	action
(example)	q_0	0	(q_0, B, R)
	$\vdots$	$\vdots$	$\vdots$

   The above example means the following: in state q_0, and with the head reading the symbol 0, M performs the action (q_0, B, R), which means that the new state is q_0, the head writes B on the square that contained 0, and the head moved to the right (R). Let L denote a head movement to the left.

2. Give your own encoding of Turing Machines. That is, show how to associate every Turing Machine uniquely with a number, and how to reconstruct a Turing Machine from any number that encodes one, and how to tell if a given number encodes a valid Turing Machine.

   Recall that a TM is a finite object given by: $(Q \cup \{q_0, q_{\text{accept}}, q_{\text{reject}}\}, \Sigma, \Gamma, \delta)$. Also note that every sequence of numbers $\{a_1, a_2, \ldots, a_n\}$ can be encoded by a single number as

[1] You can give a formal description of the Turing Machine given in the solutions to Test 2, if you prefer.

follows: $p_1^{a_1} p_2^{a_2} \cdots p_n^{a_n}$, where $p_1 < p_2 < \cdots < p_n$ is the list of the first n prime numbers. Since prime decomposition is unique, given any number x, we can write it as a product of powers of primes, and get back the sequence $a_1, a_2, \ldots, a_n$.

The point of this exercise is to show that numbers can encode Turing Machines, and that once we decide on a scheme to encode Turing Machines, we can speak of languages consisting of encodings of Turing Machines.

3. Prove the following theorem:

Theorem: Let L be a language. Then, L is decidable iff both L and L^c are semidecidable. (Here L^c denotes the complement of L; that is, $x \in L$ iff $x \in \Sigma^*$, but $x \notin L$.)