Solutions to CSE303 Final Exam Sample Questions

1. Consider the Turing machine $M = (K, \Sigma, \delta, s, \{h\}$, where $K = \{q_0, q_1, h\}$, $\Sigma = \{a, \sqcup, \triangleright\}$, $s = q_0$ and δ is given by the following table

q	σ	$\delta(q,\sigma)$
q_0	a	$(q_1, \sqcup)$
q_0	$\sqcup$	$(h, \sqcup)$
q_0	$\triangleright$	$(q_0, \rightarrow)$
q_1	a	(q_0, a)
q_1	$\sqcup$	$(q_0, \rightarrow)$
q_1	$\triangleright$	$(q_1, \rightarrow)$

(a) What is the string "M"?
Solution: We have $|K| = 3$ and $|\Sigma| + 2 = 5$, so $i = 2$ and $j = 3$ (refer to page 248 in the book). This leads to the following representations of states and symbols:

state/symbol	*representation*
q_0	$q00$
q_1	$q01$
h	$q10$
$\sqcup$	$a000$
$\triangleright$	$a001$
$\leftarrow$	$a010$
$\rightarrow$	$a011$
a	$a100$

Thus the representation of M is $q00, a000, q10, a000)$, $(q00, a001, q00, a011)$, $(q00, a100, q01, a000)$, $(q01, a000, q00, a011)$, $(a01, a001, a01, a011)$, $(q01, a100, q00, a100)$.

(b) What is the representation of the string aaa?
Solution: The encoding of aaa is $a100a100a100$.

(c) Suppose that the universal (3-tape) Turing machine U' simulates the operation of M on input aaa. What are the contents of the tapes of U' at the beginning of the simulation? At the beginning of simulation of the third step of M?
Solution: At the beginning of the simulation, the first tape contains $\triangleright \sqcup (q00, a000, q10, a000)$, $(q00, a001, a00, a011)$, $(q00, a100, q01, a000)$, $(q01, a000, q00, a011)$, $(q01, a001, q01, a011)$, $(q01, a100, q00, a100)$ the second tape contains $\triangleright \sqcup a001a000a100a100a100$, and the third tape contains $\triangleright \sqcup q00$.
M never reaches the third step of its computation - it halts on its first step.

2. Show that the class of recursively enumerable languages are closed under union and intersection.

Solution: Suppose we have Turing machines M_1 and M_2. Let M' be the nondeterministic Turing machine which, on input w, non-deterministically chooses to simulate the action of either M_1 or M_2 on w, accepting if the chosen machine accepts. Then $L(M') = L(M_1) \cup L(M_2)$. For if $w \in L(M_1) \cup L(M_2)$ then one of M_1 or M_2 must accept w and M', by choosing that machine and machine and seeing that accepting computation, also accepts, so that $w \in L(m'')$. On the other hand, if $w \in L(M')$ then there must be an accepting computation of M' on w, a computation which consists of a choice of M_1 or M_2 and then an accepting computation of that machine. In either case, one of M_1,M_2 accepts w, so $w \in L(M_1) \cup L(M_2)$.

Let L_1 and L_2 be semidecided by Turing machines M_1 and M_2. Construct a new two-tape Turing machine M' which does the following on input w. First, it copies w onto the second tape. It then runs M_1 on its first tape. If M_1 halts, M' runs M_2 on the second tape, halting if M_2 halts. Then $L(M') = L(M_1) \cap L(M_2)$. For let $w \in L(M_1) \cap L(M_2)$. In this case, both M_1 and M_2 will halt on input w, so that M' on input w will see both simulations terminate and will itself halt, so that $w \in L(M')$. On the other hand, let $w \in L(M')$. In this case, M' must have completed its algorithm on input w, in which it first saw M_1 halt on input w and then saw M_2 halt on w. Thus, since both M_1 and M_2 halt on input w, $w \in L(M_1) \cap L(M_2)$.

3. Which of the following problems about Turing machines are solvable, and which are undecidable. Explain your answers carefully.

   (a) To determine, given a Turing machine M, a state q, and a string $w$whether M ever reaches state q when started with input w from its initial state.

   Solution: This problem is undecidable. Suppose it were solvable; then some machine G would solve it. But given M and w, we could feed (M, w, h) ot G, where h is the halting state of M; if more than one, we can simply repeat our query several times, and return G's answer, and this would constitute an effective procedure for deciding the halting problem.

   (b) To determine, given a Turing machine M, whether M ever moves its head to the left when started with input w.

   Solution: The problem is solvable. Start by simulating M on input w and stop when either (1) M moves its head left, (2) M repeats a configuration, or (3) M moves its head to a position to the right of the end of the input and subsequently enters two distinct configurations C and C' with the same state and same symbol being currently scanned. If M has not moved left at this point, it never will. If (2) applies, then M is caught in an infinite stationary loop. If (3) applies but not (2), then M must periodically move to the right, but then from configuration C', M will show the same sequence of transitions that led it from C to C', and has thus entered a rightwards moving infinite loop.