MIDTERM EXAM - Answers

Do each if the following 5 problems. Write all of your answers in your blue book. 20 points total.

1. (4 points)
   Say whether each of i. - iv. is True or False (1 points each, no partial credit)
   i. If the Church/Turing thesis is true then any language which is recognized by
      a Turing machine can be decided by a TM.
   ii. There is a languages L which can be decided by a 3 tape TM but not
       recognized by any 1 tape TM.
   iii. If a set A is decidable and B is recognizable then A × B is decidable. (Note:
       A × B is the set of ordered pairs (a, b) where a ∈ A and b ∈ B.)
   iv. Every Turing machine recognizes exactly one language.

   Answers: i. False ii. False iii. False iv. True

2. EXAMPLES (6 points - 2 points each) Give examples for each of a,b and c
   below. (Two points each.)
   You need not justify your answers, but you should be precise about the example
   you are defining. It may be helpful to use set theory notation or to explain your
   terms when you give the example.
   a. Two infinite languages K ⊆ N and L ⊆ N whose intersection contains
      exactly 5 elements.
      Answer: Let K = {0, 2,4,6,8,...} = all even natural numbers
      Let L = {0,2,4,6,8, 9,11,15,...} = all even natural numbers less than 10 and all
      odd natural numbers greater than 10.
      The intersection is {0, 2,4,6,8}.
   b. One example of evidence that Church/Turing’s thesis is true.
      By evidence I mean some statement which is known to be true and which
      supports the thesis.
      (Or said another way, if the evidence you state was false then this would
      contradict the thesis or at least make it less believable.)
Answer: One is "You can simulate a k-tape TM with a 1-tape TM", for any k. This statement is true an supports the C-T thesis because it is an example of how you can strengthen the standard TM definition but not change which languages can decided or recognized. ANd it lends support to the notion that any resonable algorithmic computation can be carried out on a TM.

c. A subset of the accepting problem $A_{TM}$ which is infinite and decidable. (Note: No general procedure or proof needs to be given her. The answer to this problem should be specific for the set $A_{TM}$.)

Answer: Let N be a TM which accepts all its inputs in 1 step sby just always going into the $q_a$ state. We wrote such a TM in the first examples given in class. Say N has tape tape alphabet \{0,1,B\}.

Now the language $\{< N, s > | s \text{ is a finite binary string}\}$ is an infinite set which is decidable and a subset of $A_{TM}$.

3. (2 points) Which one of the following 4 statements is NOT true about a universal Turing machine (UTM).

a. The set of inputs for which the UTM halts is recognizable.
b. You could actually write the program of such a machine.
c. A UTM can be used to decide an undecidable problem.
d. The UTM can simulate the computation of any Turing machine on any input.

Answer: c.

4. (4 points)

One of the following sets B or C is recognizable and one is not. Pick which one is recognizable and briefly give an algorithm (or describe a TM) that recognizes it.

$B = \{ < M > | M \text{ is a Turing machine that accepts no more than 5 different inputs} \}$

$C = \{ < M > | \text{the Turing machine } M \text{ halts on at least 5 different inputs} \}$

Answer: C is recognizable.
Algorithm: We describe a TM that accepts exactly the language C.

The TM is not the universal TM U, but it runs similarly to how U does on its various inputs \(< M, w >\).

Let \( T_l \) be the finite set of all TM’s which when written as a finite string has length at most \( l \). And for each TM \( M \) in \( T_l \) let \( W_{M,l} \) be the set of all input strings to \( M \) which have length at most \( l \).

We now explain the steps of an algorithm A. A runs in infinitely many steps 1,2,3,... During these steps A may recognize some of the TM’s machine which it is simulating during that step. If it does so it accepts this machine \( M \) and never needs to refer to \( M \) in later steps.

Here is step 1 of A:
For each TM \( M \) in \( T_l \) run the machine \( M \) on all strings of \( W_{M,l} \) for \( l \) steps.
During the running of each of the \( M \)’s on its string of length \( l \) count how many string \( M \) accepts. If A sees that 5 strong have been executed as it simulates \( M \) on its input in \( W_{M,l} \), then A accepts \(< M >\) and does not run it in any of the later steps.

Claim: The machines accepted by this algorithm A are exactly the machine in C.

The proof is simple. If \( M \) is in C then at some step of A, the algorithm will simulate \( M \) on 5 strong which \( M \) eventually accepts for enough steps so that it can recognize that 5 inputs to \( M \) are accepted by \( M \), and at this point A accepts \(< M >\). and if \( M \) is not in C then this never will happen and so A will not accept \( M \).

Finally, A is discussed as an algorithm here. But Church’s thesis tells us that the algorithm A can be carried out by a TM. And so there is some TM which recognizes the set C.

5. (4 points) Assume a TM T has input alphabet \{1,2\} and tape alphabet \{1,2,B\}. Its state set consists of q0, q1, q2, qa, qr. q0 is the start state of T.

   The program of T consist of the 5 tuples
   (q0,1,q2,1,R) (q1,1,q2,1,R) (q2,1,q0,2,R)
   (q0,2,q1,2,R) (q1,2,q0,1,R) (q2,2,q0,2,L)
   (q0,B,qr,- ,-) (q1,B,qr,-,-) (q2,B,qa,-,-)

   i. Give an example of a string x which T accepts

   Answer: The string 1 is accepted. The computation of T(1) is q0 1BBB..., 1 q2 BBB..., 1 bqa BBB...
ii. Give an example of a string $y$ which $T$ rejects.

Answer:
2 is rejected. The computation is $q_0$ 2BBB..., 1 $q_1$ BBB..., 1 Bqr BBB...

iii. Define an infinite set of strings $S$ where for every $x$ in $S$, $T(x)$ does not halt.

Answer: the strings 12, 122, 1222, 12222,...

The computation of $T(12)$ is $q_0$ 12BBB..., 1 $q_2$ 2BBB..., 1 $q_0$ 12BBB, 1 $q_2$ 2BBBB, ..... and keeps going this way alternating between $q_0$ and $q_2$. And $T(122)$, $T(1222)$... would all do the same.

iv. What is the language recognized by $T$?

Answer: This one is a bit harder: we first see that $T$ accepts $2^k1$ for any $k \geq 0$. $T$ also accepts any sequence of an odd number of 1’s.