1. (Multiple Choice - 2 questions, 3 points each) Answer questions A. and B. below. There is no partial credit for this problem.

A. Consider the following TM T which has states, \{q_0, q_1, q_2, q_a, q_r\} (with q_0 = the start state, q_a = the accepting state, q_r = the rejecting state), tape symbols \{0, 1, B\}, and program d given by:

\[
\begin{align*}
    d(q_0, 0) &= (q_0, B, R) & d(q_1, 0) &= (q_1, 0, L) \\
    d(q_0, 1) &= (q_0, 1, R) & d(q_1, 1) &= (q_r, B, L) \\
    d(q_0, B) &= (q_1, 1, L) & d(q_1, B) &= (q_2, 0, L) \\
    d(q_2, 0) &= (q_2, 0, L) \\
    d(q_2, 1) &= (q_a, 1, R) \\
    d(q_2, B) &= (q_2, B, R)
\end{align*}
\]

Answer each of i, ii and iii.

i. On input 11 the TM T a. accepts, b. rejects, c. loops

ii. On input 00 the TM T a. accepts, b. rejects, c. loops

iii. On input 10 the TM T a. accepts, b. rejects, c. loops

Answer:

i. b

ii. c

iii. a

B. Which of the following statements is true about a universal TM U? (There is one correct answer here.)

i. U can decide any recognizable language

ii. U provides evidence that the Church/Turing thesis is false.

iii. U must have more than 1 work tape?

iv. There are many universal TM’s, but they tell us little about the Church/Turing thesis.

v. U can recognize any decidable language.

Answer: Both iv. and v. are true, you only needed to give one of them.
2. True or False (8 points)

Answer each of the following four True/False questions. Each is worth 2 points. There is no partial credit for problems in this section.

i. If S is any subset of a recognizable set then S is recognizable.

ii. The intersection of any decidable set S and any recognizable set T is recognizable.

iii. \{ S \mid S is a finite subset of N \} is a countable set.

iv. Let T be TM and let L(T) be the language that T accepts. If T loops on some input, then L(T) is undecidable.

Answer:

i. False

ii. True

iii. True

iv. False

3. Examples (6 points - 3 points each)

Give examples for each of a and b below.
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. There will be some partial credit given for these problems, so give a short explanation.

a. Two languages both of which are NOT decidable and whose union is decidable.

b. A subset of the halting language \( H_{TM} \) which is infinite and decidable. (Note: The answer to this problem should be specific for the set \( H_{TM} \). Recall that \( H_{TM} = \{ < M, w > \mid M(w) \text{ halts} \} \).

Answer:

a. Take any undecidable language \( A \). Then \( \overline{A} \) is also undecidable, but \( A \cup \overline{A} \) is always decidable (the TM which accepts every input decides it). Specifically, let \( A \) be \( H_{TM} \). \( H_{TM} \) and \( \overline{H_{TM}} \) are both undecidable, but \( H_{TM} \cup \overline{H_{TM}} \) is decidable.

b. \( \{ < M, w > \mid |M| \text{ is a TM which halts on } w \text{ within } |w| \text{ steps} \} \) is a decidable subset of \( H_{TM} \). The decider just runs \( M \) on \( w \) for \( |w| \) steps. If \( M \) has halted, it accepts. Otherwise, it rejects.
4. (4 points - 2 points each) Answer both i. and ii.

You are given a recognizable but undecidable set of natural numbers J, and a TM T which recognizes J.

i. Explain how to construct a proper subset S of J which is also not decidable.

Answer:

i. Since J is recognizable, it can be enumerated by an enumerator E. Build E' by running E, ignoring E's first output, and then printing everything else that E prints. The output of E' defines a proper subset S of J which is also not decidable.

ii. Explain how to construct an infinite subset R of J which is decidable.

Answer:

ii. Since J is recognizable, it can be enumerated by an enumerator E. Build E' which enumerates an infinite subset R of J by running E, printing the first output of E, and then printing E's next output x as long as the length of x is larger than the length of E's last output. E's output is an infinite subset of E's and is decidable since E enumerates in increasing order.

Part II. Do any two of the following three problems. (8 points each)

5. One of the following sets C or D is recognizable and one is not.

Pick which one of C or D is recognizable (3 points) and give an algorithm (or describe a TM) that recognizes it (5 points). Be sure that your algorithm is precise and clearly describes how the set is recognized.

C = \{ <M> | M is a Turing machine with input alphabet \{0,1\} and the language L(M) of the Turing machine M contains only binary strings of length 10 \}.

D = \{ <M> | M is a Turing machine with input alphabet \{0,1\} that halts on at least 3 different binary integer inputs between 1 and 20 \}.

Answer: D is recognizable, C is not.

Here is an algorithm A which recognizes D.

Algorithm A:

The input is a TM M. The goal of A is to accept input M if and only if M is in D.
What A does is to interweave all of the computations of M(x), for all x between 1 and 20.

It does by for i = 1, 2, 3, 4, ..., computing each of M(1), M(2), M(3), ..., M(20) each for i steps.

As it is carrying out all of these finite computations of M it keep track of and inputs x between 1 and 20 for which M(x) halts.

If it ever sees there are 3 such inputs x where M(x) halts then A stops and accepts M.

Otherwise it keeps going with the simulation of M(x).

Clearly A halts and accepts M if and only if M(x) halts on at least 3 different binary integer inputs between 1 and 20.

6. Let $F = \{ < M, w > \mid M$ is a Turing machine which takes binary inputs and M(w) halts within $2|w|$ steps of M(w)'s computation $\}$

Is the language F decidable? If yes, describe a TM (or algorithm) which decides F. If no, say why F is not decidable.

Answer: F is decidable.

An algorithm B to decide F goes as follows:

On input $< M, w >$, let $s = |w|$ and run the computation M(w) for 2s steps.

If during these 2s steps M(w) ever halts then B halts and accepts,

If during the 2s steps run of M(w) the computation does not halt, the B stops running M(w) and halts and rejects.

Clearly algorithm B is a decider (it never loops) and the language it decides is F.

7. Let L be a language recognized by a TM M. Prove that L is reducible to $A_{TM}$ ($L \leq_m A_{TM}$). You should first say what you have to prove to show this reduction then define the reduction itself.

Answer: This one is short and simple:

We show that L is reducible to $A_{TM}$ by the following mapping reduction f.

Define $f(x) = < M, x >$.

Clearly x is on L if and only if M(x) accepts if and only if $< M, x > \in f(x) \in M$. 
