HW 10 CMSC 452. Morally Due May 7
THIS HW IS TWO PAGES LONG!!!!!!!!!
SOLUTIONS

Throughout this HW $M_1, M_2, \ldots$ is a standard list of Turing Machines. Can also view as a list of all partial computable functions.

1. (60 points — 15 points for each part)

   (a) Let M be a Turing machine. Show that the following set is Σ_1:

   $$\{x \mid M(x)\downarrow\}$$

   (b) Describe an algorithm M such that

   $$\{x \mid M(x)\downarrow\}$$

   is undecidable.
   (HINT- Write an M such that the set

   $$\{x \mid M(x)\downarrow\}$$

   is HALT. Recall that HALT is

   $$\{e \mid M_e(e)\downarrow\}$$

)

   (c) Let M be a Turing machine. Show that the following set is Σ_1:

   $$\{y \mid \text{ there is some } x \text{ such that } M(x) = y\ \}$$

   (d) Describe an algorithm M such that

   $$\{y \mid \text{ there is some } x \text{ such that } M(x) = y\ \}$$

   is undecidable.
   (HINT- Write an M such that the set

   $$\{y \mid \text{ there is some } x \text{ such that } M(x) = y\ \}$$

   is HALT.
)

SOLUTION TO PROBLEM ONE

(a) Let M be a Turing machine. Show that the following set is Σ_1:

$$\{x \mid M(x)\downarrow\}$$

ANSWER:

$$\{x \mid M(x)\downarrow\} = \{x \mid (\exists s)[M(x)\downarrow \text{ within } s \text{ steps }]\}$$

(b) Describe an algorithm M such that

$$\{x \mid M(x)\downarrow\}$$

is undecidable.
ANSWER:

- Input e
- Run $M_e(e)$

The set of inputs this halts on IS HALT!

(c) Let M be a Turing machine. Show that the following set is Σ_1:

$$\{y \mid \text{ there is some } x \text{ such that } M(x) = y\ \}$$

ANSWER:

$$\{y \mid \text{ there is some } x \text{ such that } M(x) = y\ \} =$$
$$\{y \mid (\exists x, s)[M(x)\downarrow = y \text{ within } s \text{ steps }]\}$$

(d) Describe an algorithm M such that

$$\{y \mid \text{ there is some } x \text{ such that } M(x) = y\ \}$$

is undecidable.
ANSWER:

- Input e
- Run $M_e(e)$
- If you get to this step then output e.

The set of outputs of this M IS HALT!

END OF SOLUTION TO PROBLEM ONE

2. (40 points — 20 points each) A NATHAN program is a program that can, on each input, make 10 queries to HALT.

   (a) Is there a NATHAN program for the following problem: on input $(e_1, \ldots, e_{100})$ determine EXACTLY which e_i are such that $M_{e_i}(0) \downarrow$? (Formally the output is a bit string $(b_1, \ldots, b_{100})$ such that, for all $1 \le i \le 100$,

   $$M_{e_i}(0) \downarrow \text{ iff } b_i = 0.$$

)

   (b) Is there a NATHAN program for the following problem: on input n viewed as a number written in binary, output some string y such that $C(y) \ge n$ ($C(y)$ is the Kolmogorov complexity of y — the size of the smallest Turing Machine that prints out y on input 0.)

SOLUTION TO PROBLEM TWO

(a) Is there a NATHAN program for the following problem: on input $(e_1, \ldots, e_{100})$ determine EXACTLY which e_i are such that $M_{e_i}(0) \downarrow$?

YES

Note that the query *do at least i of the the machines halt on 0?* can be phrased as a question to HALT. Write a machine that runs ALL OF THEM at the same time until i of them halt -if that happens then stop. If not then of course you won't stop. So asking if this machine is in HALT is equiv to asking if at least i of the machines halt on 0.

Using this, here is the NATHAN algorithm:

   i. Input $(e_1, \ldots, e_{100})$.
   ii. Do a binary search using queries to HALT to find out EXACTLY how many of M_{e_i} halt on 0. This will take roughly $\lg(100) \approx 7 < 10$ queries to HALT.
   iii. Once you know how many of them halt RUN all of them UNTIL that many halt (this is guaranteed to happen). You then know exactly which ones halt.

(b) Is there a NATHAN program for the following problem: on input n viewed as a number written in binary, output some string y such that $C(y) \geq n$ ($C(y)$ is the Kolmogorov complexity of y — the size of the smallest Turing Machine that prints out y on input 0.)

NO

Assume, by way of contradiction, that there is such a NATHAN program. It is an Oracle TM $M^{()}$ and has a size which we call s. Let n be large (we will pick how large later). We will use M and n and a bit more (actually 10 bits more) to describe a string of high Kolm complexity.

$M^{HALT}(n)$ outputs a string y such that $C(y) \geq n$.

We cannot quite use this since WE do not have access to HALT.

But THERE EXISTS a sequence $b_1 b_2 \cdots b_{10}$ of the right answers.

We use these to describe y:

y is the output produced if $M^{()}(n)$ is run using $b_1 \cdots b_{10}$ as the answers to queries.

This description of y needs:

$M^{()}$: which is of length s

n: which is of length $\lceil \lg n \rceil$

$b_1 \cdots b_{10}$ which is of length 10

So the total length of the description is

$s + \lceil \lg n \rceil + 10 + O(1)$

Now choose n large enough such that

$$s + \lceil \lg n \rceil + 10 + O(1) < n$$

We now have a contradiction since we described a string y whose shortest description takes at least n bits in LESS THAN n bits.