CSEP531 Homework 2 Solution

1. (a) The following TM M' takes w as an input and accepts iff it is enumerated by M

```
simulate M step by step:
  if M enumerates w, halt and accept
  if M enumerates a string longer than w, halt and reject
```

First, since M enumerates strings in the increasing order of length, it generates at most $|w|+1$ strings before writing out some string longer than w. Thus, M' always halts.

Second, if w is enumerated by M, it is generated before any string longer than w. Therefore, w is accepted by M'. On the other hand, M' never accepts a string not generated by M. Therefore, M' accepts exactly the language generated by M.

Since M' always halts and accepts exactly the language enumerated by M, it *decides* this language. Therefore, this language is decidable.

(b) Suppose that L is a recognizable language. Then L is enumerable and there is a TM M that enumerates L. We will construct another TM M' that enumerates strings in an infinite subset L' of L in increasing order of their lengths. Then, by the previous part, L' is decidable.

The machine M' works as follows;

```
set t = 0
simulate M step by step:
  if M generates a string w longer than t:
    output w
    set t = |w|
```

First, it is clear that M' enumerates the strings in L' in the increasing order of their length.

Second, we claim that L' is infinite. It suffices to argue that for any t, M eventually (i.e. after a finite number of steps) generates a string longer than t; which is true since L is infinite while the set of strings of length at most t is finite.

2. (a) We will prove that the problem is not decidable. Assume for the sake of contradiction that it is. Then there is a TM $COMP$ that decides it. We design a TM ETM that decides the language $E_{TM} = \{< P > |L(P) \text{ is empty }\}$ as follows.

Given as input the description a machine P, ETM simulates $COMP$ on the pair of machines (P, Q) where Q is a machine that accepts every string (e.g. Q accepts immediately on any input). If $COMP(P, Q)$ accepts, ETM accepts; otherwise, it rejects.

Now, $P \in E_{TM}$ iff $L(P)$ and $L(Q)$ are complement, iff $COMP(P, Q)$ accepts, iff $ETM(P)$ accepts. Thus, ETM decides the language E_{TM}, which is impossible as E_{TM} is undecidable.

(b) It is decidable. A TM QH that decides this problem first simulates M on x for at most $a|x|^2 + b$ steps. If M halts before that number of steps, QH says "yes"; otherwise, QH says "no".

(c) We will prove that the problem is not decidable. Assume for the sake of contradiction that it is. Then there exists a TM REV that decides it. We will use REV to design a TM ATM that decides the language $A_{TM} = \{\langle M, w\rangle | M \text{ accepts } w\}$.

Given as input the pair (M, w), ATM constructs a TM N that behaves as follow:

> N rejects all strings except 10 and 01. It accepts 10 immediately. On 01, however, N first simulates M on w, then accepts 01 if M accepts w.

ATM then simulates REV on N. If REV accepts N then ATM accepts $\langle M, w\rangle$; otherwise, it rejects.

Now, $\langle M, w\rangle \in A_{TM}$ iff M accepts w, iff N accepts 01, iff $L(N) = \{01, 10\}$, iff REV accepts N, iff ATM accepts $\langle M, w\rangle$. Thus ATM decides the language A_{TM}, which is impossible because A_{TM} is undecidable.

3. A TM that has a finite size tape has a finite number of configurations, where a configuration is defined by the machine's tape's content, its state and head position. Thus, the only way for it to runs forever is to repeat some configurations. Furthermore, the converse is true, i.e. if the machine repeats a configuration, it will run forever. Thus, one way to answer the halting problem is to simulate the machine until either a halting configuration is reached (in which case we say "yes") or a configuration is repeated (in which case we say "no"). In the worst case, we have to go through all possible configurations before either of them happens. Thus, the number of steps we might need is the number of possible configurations, which is

$$T = 3^{2^{34}} 2^{10} 2^{34}$$

The $3^{2^3 4}$ part is an upper bound of the number of possible tape content, since each cell contains either 1,0 or blank. The second part, 2^{10}, is the number of states. Finally, the third part, 2^{34} is a number of head position.

The time it takes to perform the check is $T/10$, which is at least $3^{2^{34}}$. Some students claimed that it is thousands time the age of the universe. Also, note that we need to remember the configurations we have seen in order to check for repeats. This would require a memory millions time as large as the number of particles in the universe.

4. Assume for the sake of contradiction that $A = \{0^n 1^n | n \geq 1\}$ is regular. Then there is a DFA D accepts it. Assume that D has k, which is a constant, states. Then for $n > k$, by the pigeon hole principle, D must repeat some states scanning the 0^n parts. Thus, there exist some $i < j < n$ such that D is in the same state after scanning 0^i and 0^j. Thus, it either accepts both $0^i 1^i$ and $0^j 1^j$ or rejects both of them. In both cases we have a contradiction.

Note that if you are familiar with the pumping lemma, you can also use it.

5. (a) Note that A_{TM} is Turing recognizable. A machine ATM that recognizes A_{TM} can be described as follow. On input (P, x), ATM simulates P on x and accepts if P accepts x.

First, we claim that *if A is Turing recognizable then A is mapping reducible to A_{TM}*. In order to do so, we need to construct a *computable* function $f : \Sigma^* \rightarrow \Sigma^*$ such that $w \in A$ iff $f(w) \in A_{TM}$. Since A is Turing recognizable, there exists a machine M_A that accepts all strings in A and nothing else. Let $f(w) = \langle M_A, w\rangle$.

Clearly, f is computable. Furthermore, $w \in A$ iff M_A accepts w, iff $f(w) = \langle M_A, w\rangle \in A_{TM}$. This proves the claim.

Second, we claim that *if A is mapping reducible to A_{TM} then A is Turing recognizable.* Since A is mapping reducible to A_{TM}, there exists a computable function f such that $w \in A$ iff $f(w) \in A_{TM}$. We construct a machine M_A that recognizes A using the machine ATM that recognizes A_{TM} as follow.

On input w, M_A first computes $f(w)$ - this can be done since f is computable. It then simulates ATM on $f(w)$. If ATM accepts $f(w)$, M_A accepts w.

Note that M_A accepts w iff ATM accepts $f(w)$. Thus for any string x, $x \in A$ iff $f(w) \in A_{TM}$, iff ATM accepts $f(w)$, iff M_A accepts w. Thus, M_A recognizes A.

(b) Note that $C = \{0^n 1^n | n \geq 1\}$ is decidable. One machine that decides it works as follow. First, it counts the number of 0's before seeing a 1. Then it counts the number of 1's and verifies that no more 0 appears. Finally, it check if the numbers of 0's and 1's are the same.

Similar to above, we first claim that *if a language B is decidable then it is mapping reducible to C*. Since B is decidable, there exists a machine M_B that decide B. We construct the function f as follows:

$$f(w) = \begin{cases} 01 & \text{if } M_B \text{ accepts } w \\ 0 & \text{otherwise} \end{cases}$$

Since M_B always halts, f is computable. Furthermore, $f(w) \in C$ iff $f(w) = 01$, iff M_B accepts w, iff $w \in B$.

The proof of the second claim that *if a language B is mapping reducible to C then it is decidable* is almost identical to the one in the previous question.