2.5/2

Lamess Kharfan
Student ID: 10150607

CPSC 313 – Assignment 4

1) Give a Turing machine with input alphabet {1, #} that on input $1^m\# 1^n$ halts with $1^{\gcd(m,n)}$ written on its tape.

Let TM(GCD) = "On input x:

1. If the input is not of the form $1^m\# 1^n$ then reject. This can be checked by reading the input starting at the beginning of the string, expecting consecutive 1's, a #, and consecutive 1's again. If the string deviates from what we expect, we reject it.
2. Split the tape into 2 tapes by the #, deleting the #, and placing one above the other.
3. We check which tape has the larger input by starting at the beginning of the top and bottom tape, when we cross a 1 on one tape, we should expect to cross a 1 on the other tape. If we cross a 1 on one tape, and encounter a blank on the other, then the one where we encountered a blank is the smaller input, put that tape on the bottom, and the larger one on the top.
4. Starting at the beginning of the bottom tape, repeat:
   a. Check if there is an uncrossed 1 on the bottom tape and the top tape.
   b. If there is, cross them both, move one to the right, and goto a.
   c. If we encounter a blank on the bottom tape, but still are 1's on the top tape, replace all crossed 1's on the top tape with blanks, and uncross the 1's on the bottom tape. Goto a.
   d. If there is a 1 on the bottom tape that matches with a blank on the top tape, uncross the 1's on the top tape, uncross the 1's on the bottom tape, and swap the tapes positions (bottom is now top, top is now bottom. Go to a.
   e. If the bottom tape still contains 1's and the top tape is all blanks, meaning it is equal to 0, then reject.
5. Read the output on the bottom tape. This is the gcd(m,n)."

Great!

2) Let Σ be an alphabet, and suppose that A, B ⊆ Σ ∗ are Turing recognizable languages for which both A ∩ B and A ∪ B are decidable. Prove that A is decidable.

We know that, for some input x:

- TM(AUB) decides AUB, meaning:
  - If x ∈ AUB, TM(AUB) accepts.
  - If x ∉ AUB, TM(AUB) rejects.

Lamess Kharfan
Student ID: 10150607

- TM(A ∩ B) decides A ∩ B, meaning:
    - If x ∈ A ∩ B, TM(A ∩ B) accepts.
    - If x ∉ A ∩ B, TM(A ∩ B) rejects.
- TM(A) accepts A, meaning:
    - If x ∈ A, TM(A) accepts.
- TM(B) accepts B
    - If x ∈ B, TM(B) accepts.

To prove that A is decidable, we need to create a TM, TM(DEC_A), that decides A, such that for some input x, if x ∈ A, TM(A) accepts, If x ∉ A, TM(A) rejects.

Let TM(DEC_A) = "For some input x in A:

1) Run TM(AUB) on x.
2) If TM(AUB) rejects, then reject. This filters out all strings not in A and not in B.
3) If TM(AUB) accepts, run T(M)A on x. If TM(A) accepts, then TM(DEC_A) accepts.
4) Otherwise, run TM(A∩B).
5) If TM(A∩B) rejects, then x is either in B-(A∩B) or A-(A∩B). Run TM(B).
6) If TM(B) accepts, and we know we have rejected all possibilities of A∩B already from the reject in (4), then this implies that A has been rejected, therefore TM(DEC_A) rejects.

Perfect!

Thus, since A is able to accept x if x is in A, or reject x if x is not in A, A is decidable.

3) Prove that the following language is undecidable.

$$\text{REVACCEPT} = \{< M >|\ \text{M accepts} < M >^R \}$$

Assume that there is a TM RA that decides REVACCEPT. For any TM M:

(1) RA accepts <M> ⇔ M accepts $< M >^R$
RA rejects <M> ⇔ M does not accept $< M >^R$

Let RAx be a TM obtained from RA by:

- Redirecting any transitions to go into the opposite direction.
- Making the accept state the reject state, and the reject state the accept state.

Perfect!

Put RAx instead of M into (1):

- RA accepts $< RAx >$ ⇔ RAx accepts $< RAx >^R$
- RA rejects $< RAx >$ ⇔ RAx does not accept $< RAx >^R$