Homework 4

Ryan Johnston

No Institute Given

Theory 3500

Problem 1: We show that any language L that is Turing recognizable can be reduced to A_{TM}. Given that L is Turing recognizable, there exist a Turing machine M_L that recognizes L. We now define our reduction function (mapping) to be as follows:

Given w, $f(w) = < M, w >$

Notice $w \in L$ if and only if $f(w) \in A_{TM}$

Problem 2: Assume that $S = \{< M >: w \in L(M) \text{ iff } w^R \in L(M)\}$ is a Turing recognizable language. We will now show a reduction from A_{TM} to S. Let M_S be a Turing machine recognizing S, we define the following machine R

$R =$ on input $< M, w >$
1. Create the following machine:
    - $T =$ on input x
        1. Run w on M
        2. Accept x if $x = 01$
        3. If w is accepted by M, AND $x = 10$ then accept x.
        4. Otherwise Reject x
2. Run M_S on T
3. Accept if M_S accepts and reject if M_S rejects.

Now we show that the above machine decides A_{TM} given our assumption that S is Turing recognizable. Notice that the language produced by T which we denote by $L(T)$, is determine by whether or not M accepts w. If w is not accepted then $L(T) = \{01\}$, and if w is accepted then $L(T) = \{01, 10\}$. So it follows that If $< M, w > \in A_{TM}$ iff then T is accepted by M_S i.e. $T \in S$. However, we know that A_{TM} is undecibable, therefore our original assumption that S is decidable must be false.

Problem 3

1. Decidable. We know that A_{NFA} is decidable, i.e. Theorem 4.2 in Sipser. Further more given that our alphabet Σ is finite (the definition of an alphabet is any finite set), there are exactly $|\Sigma|^k$ different strings of length k in Σ^*. So here is a Turing machine deciding L_1:

M_{L_1} = on input $< D, k >$

For every string $w \in \Sigma^k$, run A_{NFA} on $< D, w >$ If there exist one such w which is an element of $L(D)$, i.e. the language produced by D, then accept otherwise reject.

2. undecidable. Assume that $L_2 = \{< M >: L(M)$ is a contex free language $\}$ is a Turing recognizable language. We will now show a reduction from A_{TM} to L_2. Let M_{L_2} be a Turing machine recognizing L_2, we define the following machine R

R = on input $< M, w >$

1. Create the following machine:
    - T = on input x
        1. Run w on M
        2. Accept x if x is of the form $0^\ell 1^m 0^n$
        3. If w is accepted by M, then accept x.
        4. Otherwise Reject x
2. Run M_S on T
3. Accept if M_{L_2} accepts and reject if M_{L_2} rejects.

Now we show that the above machine decides A_{TM} given our assumption that S is Turing recognizable. Notice that the language produced by T which we denote by L(T), is determine by whether or not M accepts w. If w is not accepted then $L(T) =$, and if w is accepted then $L(T) = \{01, 10\}$. The former language is not contex free an the latter obviously is. So it follows that If $< M, w > \in A_{TM}$ iff then T is accepted by M_S i.e. $T \in S$. However, we know that A_{TM} is undecibable, therefore our original assumption that S is decidable must be false.

3. Undecibable. Assume that $L_3 = \{< M > : M$ is a Turing machine and $L(M)$ is finite $\}$. We will now show a reduction from A_{TM} to L_3. Let M_{L_3} be a Turing machine recognizing L_3, we define the following machine R

R = on input $< M, w >$

1. Create the following machine:

   $T =$ on input x

   1. Run w on M
   2. Reject x if w is accepted, otherwise accept

2. Run M_{L_3} on T
3. Accept if M_{L_3} accepts and reject if M_{L_3} rejects.

Now we show that the above machine decides A_{TM} given our assumption that L_3 is Turing recognizable. Notice that the language produced by T which we denote by L(T), is determine by whether or not M accepts w. If w is not accepted then $L(T) =$, and if w is accepted then $L(T) = \{\Sigma^*\}$. The former language is finite and the latter obviously is infinite. So it follows that If $< M, w > \in A_{TM}$ iff then T is accepted by M_{L_3} i.e. $T \in S$. However, we know that A_{TM} is undecibable, therefore our original assumption that S is decidable must be false.

4. Undecibable. Assume that $L_4 = \{< M > : M$ is a Turing machine, T is a computable function, and M accepts w in $T(|w|)$ steps $\}$. We will now show a reduction from A_{TM} to L_4. Let M_{L_4} be a Turing machine recognizing L_4, we define the following machine R

We define the function $T_{M,w}(n)$ to be the halting time of M on w when M halts on input w otherwise define it to be -1, i.e. T is a constant function.

$R =$ on input $< M, w >$

1 Run M_{L_4} on $\langle M, w, T_{M,w} \rangle$

3. Accept if M_{L_4} accepts and reject if it rejects.

Now we show that the above machine decides A_{TM} given our assumption that L_4 is Turing recognizable. Notice that $T_{M,w}$ is computable if and only if M accepts or rejects on input w (and does not loop). If $< M, w > \in A_{TM}$ then M halts on w and $T_{M,w}$ is computable. Also $T_{M,w}(|w|)$ is equal to the number of steps M runs before accepting w. Thus $< M, w, T_{M,w} > \in L_4$. Conversely, If $< M, w > \notin A_{TM}$, then either $T_{M,w}$ is not computable or M rejects w, i.e. it does not accept in when it halts in $T_{M,w}(|w|)$ steps. In either case $< M, w, T_{M,w} > \notin L_4$