***Pledge:*** *I pledge my honor that I have abided by the Stevens Honor System.* - **Eric Altenburg**

---

**1: (a) Show that $\leq_m$ is a transitive relation. (b) Show that if A is TM-recognizable and $A \leq_m \overline{A}$, then A is decidable.**

---

**(a)**

Since $A \leq_m B$, have $B \leq_m C$. This means there are **computable** functions f and g that satisfy the following:

$w \in A \iff f(w) \in B$ and $y \in B \iff g(y) \in C$.

Now we can consider the composition function $h(w) = g(f(w))$.

Following this, we then construct a TM that computes $h$:

Simulate a TM for $f$ on input $w$ and call the output $y$.

Now simulate another TM for $g$ on $y$.

The output for this is $h(w) = g(f(w))$. Therefore, h is also a computable function. Also, since $w \in A \iff h(w) \in C$, this means that $A \leq_m C$ from the reduction of $h$.

**(b)**

If $A \leq_m \overline{A}$, then we can say that $\overline{A} \leq_m A$ from the same mapping reduction.

Since we know that A is TM-recognizable, this means that $\overline{A}$ is also TM-recognizable, and from this, it means that A is decidable.

---

**2: Is $DISJOINT_{TM}$ decidable or undecidable? Prove your answer.**

---

Assume $DISJOINT_{TM}$ is decidable. Because of this there is a decider D. Then we say that L(B) is all possible inputs.

Now use D to create another decider J that can decide on the language $E_{TM}$ such that {<M>: L(M) = $\phi$}.

From this, we know that $E_{TM}$ is not a decidable language, therefore, its decider J does the following:

For the decider J, J(<A>) =

  If D(<A>, <B>) accepts, then ACCEPT. Otherwise, REJECT.

given that L(B) = $\Sigma^*$.

So <A> $\in E_{TM}$ IFF (<A>, <B>) $\in DISJOINT_{TM}$. If L(A) $\neq \phi$, then (<A>, <B>) $\notin DISJOINT_{TM}$. Therefore, $E_{TM} \leq DISJOINT_{TM}$. Since $E_{TM}$ is undecidable, $DISJOINT_{TM}$ must be as well.

---

**3: A triangle is an undirected graph is a cycle of length 3. Show that the language $TRIANGLE = \{< G >: \textit{graph G contains a triangle}\}$ is in P.**

---

Let G = (V, E) where its set of vertices are V and its set of edges are E. We can enumerate all triples (u, v, w) with vertices $u, v, w \in V$ and $u < v < w$. Then check if the all possible pair of the three edges (u, v), (v, w), and (u, w) are in the enumerator E.

Enumerating all of these triples has time of $\mathcal{O}(|V|^3)$. Then checking to see if the pairs belong to E take time of $\mathcal{O}(|E|)$. Overall time is now $\mathcal{O}(|V|^3|E|)$, which is polynomial in the length of the input of <G>. From this it is now proven that $TRIANGLE$ is in P.

---

**4:**

---

### (i)

First ask whether or not the formula is satisfiable. To which they will reply yes because we have not made any guesses for the boolean variables yet.

Then give a guess for $x_1$.

If they say that the formula is still satisfiable, then the guess that was just made is correct. If it is not satisfiable, then you change it to the inverse boolean value.

Repeat for the rest of $x_2$ ... $x_n$.

### (ii)

Maximum number of queries made for the above algorithm is $n + 1$.

### (iii)

As one goes along after the first query, you continually guess the current boolean value and see whether or not the formula is still satisfiable. If it is true, then you can move on, otherwise, since you are only working with boolean values, swapping the value to its inverse is easy thus preventing the asker to start over from the very beginning. In the end after guessing and checking if still satisfiable, you will have satisfying assignments for a satisfiable formula.