CS 334 - PS 9

(12/1/21)

"I pledge my honor I have abided by the Stevens Honor Code." - Haris Spahic
(Partner ~ Hiya Bhavsar)

1. According to Corollary 5.29 in the textbook, if $A \leq_m B$ and A is not turing recognizable, then B is not turing recognizable.

Thus, to show S is not turing recognizable we can show $A \leq_m \bar{S}$, for some non-turing recognizable language A. Conveniently, $\bar{A}_{TM}$ is such a language & the property $A \leq_m B \Leftrightarrow \bar{A} \leq_m \bar{B}$ holds. Thus we show $A_{TM} \leq_m \bar{S}$, implying $\bar{A}_{TM} \leq_m S$.

Consider the reducing function f.

F = "On input $\langle M, w \rangle$ where M is a TM & w is a string"

1. Construct the following TM, M_1, M_2, M_3.

   M_1 = "on any input"
   1. reject

   M_2 = "on any input x"
   1. If $x == \langle M_2 \rangle$ accept.
   2. Else run x on M, and accept if accepts.

   M_3 = "on any input"
   1. Run $\langle M \rangle$ on M.
   2. If accept, output $\langle M_1 \rangle$
   3. Else output $\langle M_2 \rangle$

2. If $w == \langle M \rangle$, run w on $\langle M \rangle$.
3. If accepts, return $\langle M_1 \rangle$.
4. Else output $\langle M_2 \rangle$.
5. Else run M_3.

Thus if $w == \langle M \rangle$, is accepted we return $\langle M_1 \rangle$ which rejects itself & is accepted by $\bar{S}$. If $w == \langle M \rangle$ is rejected, then return M_2 which accepts itself, and is thus rejected by $\bar{S}$. Otherwise w is not $\langle M \rangle$ so we run M_3, which checks if M accepts itself, and does the same action. Thus $A_{TM} \leq_m \bar{S}$, meaning $\bar{S}$ is not turing recognizable. I don't have enough time to show S is not turing recognizable but the proof should be almost Identical. Just with M_1 &

2.

1. Let P be any property of the language of a TM that satisfies conditions a and b,
2. Assume that $\mathcal{L}$ is decidable and let TM D decide $\mathcal{L}$.
3. Since condition b is satisfied by P, let $\langle M \rangle \in \mathcal{L}$ and $\langle N \rangle \notin \mathcal{L}$.
4. Now consider the following TM X:

   On input w:

   1. Compute own description $\langle X \rangle$.
   2. If D accepts $\langle X \rangle$ then run w on $\langle N \rangle$.
   3. If D rejects $\langle X \rangle$ then run w on $\langle M \rangle$.

Since in both cases X contradicts D, we conclude that $\mathcal{L}$ is undecidable.

3. Recall $A <_p B$ if there is a polynomial-time computable function f that $w \in A \Leftrightarrow f(w) \in B$.

3a. Show transitivity. Ie show $A <_p B$, $B <_p C \Rightarrow A <_p C$.

$\rightarrow$ $w_1 \in A \Leftrightarrow f(w_1) \in B \quad \wedge \quad w_2 \in B \Leftrightarrow g(w_2) \in C$

Notice if we compose $g(f(w))$ then $w_2 = f(w_1) \in B$. Thus given any input $w_1 \in A \rightarrow g(f(w_1)) \in C$. Since $g(f(w_1))$ is a computable function f that satisfies

$w \in A \Leftrightarrow g(f(w)) \in C \Rightarrow A <_p C$. $\blacksquare$

3b. Construct a TM X to decide B using A.

X: "on input w"

1. Construct a description of $\langle A \rangle$.
2. Run w on $\langle A \rangle$.
3. If $\langle A \rangle$ accepts, we know since $B \neq \emptyset$ & $B \neq \Sigma^*$, $\overline{B} \wedge B$ are not empty. Thus there are some strings, a & b in B & $\overline{B}$ respectively

Erase the tape and output a.

4. Else A rejects, output b.

We don't need to know what a & b are, just that they exist and we can find one in polynomial time.

3c. Notice, by 3b $\forall A, B \in P$ if $B \neq \emptyset$ and $B \neq \Sigma^*$ then $A <_p B$. Since $P = NP$, then

$\rightarrow \forall A, B \in P = NP \rightarrow A <_p B$ where we exclude $\emptyset$ & Σ^*.

For a language (Say B) to be NP-Complete:

1. $B \in NP$
2. $\forall A \in NP \rightarrow A <_p B$

Since $P = NP$, this is true for all B in P, and we showed property 2 above.

4. We construct a turing-machine to decide TRIANGLE In polynomial time. Call it Square.

Square: "on input $\langle G \rangle$"

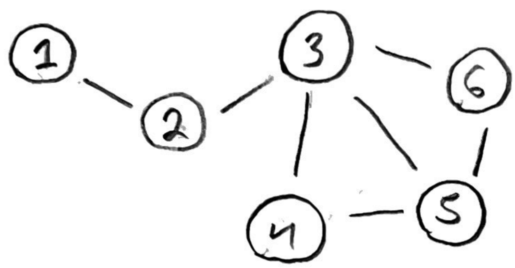
1. Output $\langle G \rangle$ on the tape.
2. Run BFS on $\langle G \rangle$, mark each node lexographically and write the transition between each node onto the tape. (Takes $E + V$ where E is # Edges, V is # nodes)
3. For every #'d node, run through each transition pair. If there is only one transition containing that #, remove it. Repeat 3 until no nodes are removed. (Time E^2)
4. Run BFS on the graph. Start at the lowest remaing node, then go to the edge with the smallest number

that isn't our current node. Then traverse the entire graph marking as we go. Simultaneously, keep a count for every node that's incremented whenever we visit a node. Whenever we node with multiple edges to unvisited nodes, keep a count of number of increments. If we reach a node with a count, check if that count is 3. If it is accept. Else, continue. When we return recursively to a node with many paths to new nodes, subtract number of increments from each node that was incremented during that path.

6. Else, reject.

Since this turing machine square decides TRIANGLE in $O(E^2)$ time, which is polynomial, Triangle is polynomial.

Ex of running algorithm

G = 

2. Label nodes, & edges.

12, 23, 34, 35, 36, 45, 56

3. Remove "loners / leafs"

23, 34, 35, 36, 45, 56

34, 35, 36, 45, 56

4. Start at min pair (3,4). 3 is new node, increment.

3	4	5	6
IIII	II	I	

34, 35, 36, 45, 56

Then since 5 is the smallest pair with 3, go to 5. New node.

34, 35, 36, 45, 56

Then 53. Since 3 has a count, check if its 3. It is, accept.

34, 35, 36, 45, 56