Theory of Computation HW9 P vs NP Ryzeson Maravich

1

The following questions are based upon the Cook-Levin Theorem as described in the Sipser textbook.

a. Which of the four parts of the formula are dependent on the TM N?
$\phi_{cell}, \phi_{start}, \phi_{move}$, and ϕ_{accept}
b. Which of the four parts of the formula are dependent on the the string w?
$\phi start$
c. Give a different legal window for the Turing Machine described as an example.

a	q_1	a
a	b	q_1

a	a	#
a	a	#

a	a	a
q_2	a	a

d. Give a different illegal window for the Turing Machine described as an example.

a	a	a
a	b	a

a	#	b
a	#	b

a	q_1	a
q_2	a	a

e. Why are there n^k columns in the tableau?
Each row of the tableau corresponds to a configuration of the Turing machine N on w. We assume that N runs in n^k time (where n is the length of w) for some constant k. This means that there are at most n^k possible configurations, which accounts for the n^k rows. It takes n steps for the head to reach the final character on the tape. From here the tape can add one character for the remaining n^{k-1} steps. Therefore the longest any configuration could be is $n + n^{k-1} = n^k$ characters long, which is why there are also n^k columns.

2

Claim: The formula $(x \vee y) \wedge (x \vee \overline{y}) \wedge (\overline{x} \vee y) \wedge (\overline{x} \vee \overline{y})$ is unsatisfiable.
Proof: Let us denote the formula by ϕ, and create the following truth table.

x	y	$x \vee y$	$x \vee \overline{y}$	$\overline{x} \vee y$	$\overline{x} \vee \overline{y}$	ϕ
T	T	T	T	T	F	F
T	F	T	T	F	T	F
F	T	T	F	T	T	F
F	F	F	T	T	T	F

Because there is no assignment of truth values to the variables that makes ϕ true, ϕ is unsatisfiable.

3

A triangle in an undirected graph is a 3-clique. Let $TRIANGLE = \{\langle G \rangle | G \text{ contains a triangle}\}$.
Claim: $TRIANGLE \in$ P.
Proof: Create a Turing machine M that decides $TRIANGLE$.
$M =$ "On input $\langle G \rangle$ where G is a graph with at least three nodes:

1. List all possible triples of nodes in G.

2. For each triple, see if their edges are contained in the encoding of G. If they are contained for some triple, *accept*, otherwise *reject*."

Step 1 can be computed in m^3 stages where m is the number of nodes in G. (m^3 is obtained by testing all m nodes as the first node in the triple, then all m as the second, and then the third.) Step 2 takes n stages where n is the number of edges in G, so collectively M takes $m^3 n$ stages to run. Because M runs in polynomial time, $TRIANGLE \in$ P.

4

G and H are isomorphic if the nodes of G may be reordered so that it is identical to H. Let $ISO = \{\langle G, H\rangle | G$ and H are isomorphic graphs$\}$.
Claim: $ISO \in$ NP.
Proof: Create a non-deterministic Turing machine M_2 that decides ISO.
$M_2 =$ "On input $\langle G, H\rangle$ where G and H are graphs:

1. Test to see if G and H have the same nodes.
2. Non-deterministically guess all possible reorderings of the nodes in G, and check to see if a reordering is identical to H.
3. If both accept, *accept*, otherwise *reject*."

Step 1 could be computed in $m \log m$ stages where m is the number of nodes in G. Because step 2 is done non-deterministically, it runs in $m + n$ time, where n is the number of edges. Because these are both polynomial, ISO can be decided in polynomial time, and therefore is in NP.

Alternate Proof: Create a verifier V for ISO, and let c be the certificate.
$V =$ "On input $\langle\langle G, H\rangle, c\rangle$:

1. Test whether c is a graph that contains the same nodes as G.
2. Test whether c's encoding of nodes and edges is identical to H's.
3. If both accept, *accept*, otherwise *reject*."

Step 1 can be computed in $m \log m$ time where m is the number of nodes in G. Step 2 can be computed in $m + n$ time where n is the number of edges. Because these are both polynomial, ISO can be verified in polynomial time, and therefore is in NP.

I affirm that I have upheld the highest principles of honesty and integrity in my academic work and have not witnessed a violation of the honor code.