Complexity - Exercise 2

Oren Roth

December 10, 2017

Question 1

Let M be the TM which decide A and $p(n)$ a polynomial, which for any input with size n, M runs at most $p(n)$ steps with oracle access to inputs with length at most $n-1$. We will describe M' that will use only $n * p(n)$ space for any input of size n and decide A. This will show $A \in PSPACE$. Given $x \in \Sigma^*$, M' will save n slots with size $p(n)$ each. M' runs as follows:

- Copy x to the first slot
- run M on the input in slot 1. When oracle request is asked copy the input for the oracle to the next slot and runs M on it.
- Continue the same process when oracle request is asked copy the input for the oracle to the next slot and runs M on it and so on.
- When M answer on input of slot $i > 1$, clear slot i and return the answer to M on slot $i-1$ that asked the question to the oracle and continue the run of M on slot $i-1$.
- When answer of M appears on slot 1, M' return the same.

First notice that any calls for oracle is shrink the input by one, so we will use at most n slots, because each slot is with size $p(n)$, M' uses $poly(n)$ space. Moreover as M need $p(n)$ time for an input in size n, it for sure use at most $p(n)$ space for inputs in size $\leq n$, in each slot we run on inputs with size $\leq$ to n and hence each slot as sufficient space for running M.
If all the calls for the oracle are correct M' is just simulating M on slot 1 and hence $L(M') = L(M)$. Assume that all the calls for the oracle in input y with $|y| < k$ are simulating correctly by M' and let y be input for the oracle with $|y| = k$, for this we will open new slot and run M on y, with every oracle call on inputs in size at most $k-1$ which by the induction assumption we know M' simulate correctly and hence, M' on y will answer exactly as M on y.

Question 2

Assume $NP = CO - NP$, we will show $PH \subseteq NP$ and hence $PH = NP$.

$$PH = \bigcup_{i=0}^{\infty} \Sigma_i^p$$

We will show by induction on i that $\Sigma_i^p \subseteq NP$

Question 3

Question 4

a.

b.

SAT is a special case of boolean formula, so as we showed in 1, iff $x \in SAT$ there is a witness for it, a interpretation for it a with:

$$(x, a) \in FVAL$$

Because the witness tape is just like a normal tape we can run the algorithm from section a, and determine $x \in SAT$ iff $\exists a$ s.t. $(x, a) \in FVAL$.

c.

Claim 0.1 *G is not bipartite graph $\iff$ there is a cycle of odd length in G*

We will show that $BIPARTITE \in CO - NL$ and by theorem we showed in class ($NL = CO - NL$) we will conclude $BIPARTITE \in NL$. Let us show $\overline{BIPARTITE} \in NL$ by describe M non deterministic TM which decide $\overline{BIPARTITE}$.
M will receive as an witness the odd cycle in G (by our claim we know there exist iff $G \in \overline{BIPARTITE}$) and just check that this cycle exists in G.
M will accept $\iff$ there is such witness $\iff$ $G \in \overline{BIPARTITE}$.

Proof of claim *($\Leftarrow$) Let $C = \{v_1, v_2, \ldots, v_{2n+1}\}$ be a cycle of odd length, by contradiction assume $G = (V \cup E)$ is bipartite with $V = L \cup R$ division of V to two disjoint sets as promised. WLOG assume $v_1 \in L$ so $v_2 \in R$, $v_3 \in L$ and so on (there is edge between those vertices so they have to be in different sets), we conclude all the odd vertices are in L but there is edge between v_{2n+1} to v_1 and we arrive to contradiction.*
($\Rightarrow$) Let be ■

Question 5