COMS 4236 Homework 3

Mengqi Zong $< mz2326@columbia.edu >$

March 20, 2012

Problem 1

a)
Since C has 20 input variables, there are 2^{20} possible input assignments, which is a constant. For any given CIRCUIT-SAT-20 problem, we can solve it using brutal force algorithm in $O(2^{20})$. So, CIRCUIT-SAT-20 is in P.

b)
$B \in NP$.

Since C is NP-complete, given an input assignment α_C, we can verify the output in P. Because $B \leq_p C$, for any input assignment α_b, there is a poly-time reduction $R_{B \to C}(\alpha_B)$ from B to C. Therefore, we can also verify the output of α_B in polynomial time. So $B \in NP$.

B is NP-hard.

Since $A \leq_p B$, there is a poly-time reduction $R_{A \to B}(\alpha_A)$ from A to B. Because A is NP-complete, A is NP-hard. If $B \in P$, then we can also solve A in poly-time, which contradicts the fact that A is NP-hard. So, B must be NP-hard.

To sum up, B is NP-complete.

Problem 2

a)

$\bar{L} \in coC$.

Since L is complete for class C, $L \in C$. By the definition of $coC = \{\bar{L}|L \in C\}$, we know that $\bar{L} \in coC$.

$\bar{L}$ is coC-hard.

Since L is complete for class C, then L is C-hard. That is, any language $L' \in C$ can be reduced to L. For any input assignment $\alpha \in L'$, we have $R(\alpha) \in L$. Then for any input assignment $\alpha' \in \bar{L}'$, we have $R(\alpha') \in \bar{L}$. That is, any language $\bar{L}' \in coC$ can be reduced to $\bar{L}'$. So $\bar{L}$ is coC-hard.

To sum up, $\bar{L}$ is coC-complete.

b)
If C = coC, M is in coC.

Since M is complete for C, $M \in C$. Due to C = coC, $M \in coC$.

If M is in coC, C = coC.

- $C \subset coC$

  First, we will prove that coC is closed. Since C is closed, then whenever L reduces to L' and L' is in C, then also L is in C. We can also get whenever $\bar{L}$ reduces to $\bar{L}'$ and $\bar{L}'$ is in coC, then also $\bar{L}$ is in coC. So coC is also closed.

  Since M is C-complete, any language L in C can be reduced to M. Since M is in coC and coC is closed, any language $L \in C$ is in coC. We get $C \subseteq coC$.

- $coC \subset C$

  Since M is in coC, then $\bar{M} \in C$. Form part a, we know that $\bar{M}$ is coC-complete. That means any language L in coC can be reduced to $\bar{M}$. Because $\bar{M}$ is in C and C is closed, then any language $L \in coC$ is in C. We get $coC \subseteq C$.

To sum up, C = coC.

Problem 3

a)
NP

For any language L that $L \leq_p L'$ and $L' \in NP$, we will prove that L is in NP.

L can be verified in polynomial time. Since L' is in NP, we know that L' can be verified in polynomial time. We also know the reduction takes poly-time, then we can verify L in polynomial time by first reducing the input α into $R_{L \to L'}(\alpha)$ in poly-time, then verify $R_{L \to L'}(\alpha)$ also in poly-time. As a result, we can verify L in poly-time. So L is in NP.

coNP

For any language L that $L \leq_p L'$ and $L' \in coNP$, we will prove that L is in coNP.

Since L' is in coNP, $\bar{L'} \in NP$. And $\bar{L'}$ can be verified in polynomial time. Since the reduction is polynomial time, then $\bar{L}$ can also be verified in polynomial time. So $\bar{L} \in NP$, and L is in coNP.

PSPACE

For any language L that $L \leq_p L'$ and $L' \in PSPACE$, we will prove that L is in PSPACE.

We know that $TIME(f(n)) \subset NTIME(f(n)) \subset SPACE(f(n))$, so $PTIME \subset PSPACE$. Since L' is in PSPACE and the reduction takes at most PSPACE, then L is in PSPACE.

b)
There exists language L and L' that $L \leq_p L'$ and $L' \in SPACE(n)$, but L is not in SPACE(n).

Let language L be a language in $TIME(n^2)$ and $L' = pad(L, n^2)$. From homework 2, we know that L' is in TIME(n). Since $TIME(f(n)) \subset NTIME(f(n)) \subset SPACE(f(n))$, we know that L' is in SPACE(n). And obviously, the reduction from L to L' takes $O(n^2)$, which is polynomial. However, L is not necessarily in SPACE(n). Because $TIME(n^2) \subset SPACE(n^2)$, L could be in $SPACE(n^2)$. In this case, L, which is in $SPACE(n^2)$, can be reduced to L', which is in SPACE(n) by polynomial time reductions. So SPACE(n) is not closed under polynomial time reductions.

Problem 4

a)
DAG Reachability problem is in NL.

Given $H = (N, A)$, node s, t, we can check if there is a path in H from s to t in NL. All we need to do is starting at node s, randomly guess the next node to visit, until we reach node t or there's no node to visit (the graph is acyclic). Since we just need to store the current node, which takes log-space, then DAG Reachability problem is in NL.

DAG Reachability problem is NL-hard.
We know that general Graph Reachability problem is NL-complete. And we can reduce Graph Reachability problem to a DAG Reachability problem. The reduction is the transformation given in the content. Since we just need to store the index of node u, v, and the respective number i,j, which takes log space, this transformation takes log-space. As a result, DAG Reachability problem is NL-hard.

To sum up, DAG Reachability is NL-complete.

b)
Graph Cyclicity is in NL.

We can solve the Graph Cyclicity problem in NL. All we need to do guess a node which is in the cycle. Then guess the length of the cycle. Then start traversing from this node, guess the next node to visit. After the length of the cycle step, if the current node is the starting node, then return true. If not, return false. Since we just need to store the starting node, the current node and the length of the circle, which takes log-space, Graph Cyclicity is in NL.

Graph Cyclicity is NL-hard.

We can reduce the DAG Graph Reachability problem to Graph Cyclicity. Given $H = (N, A)$, two node s, t, we can translate graph H into another DAG $H' = (N, A'), A' = A \cup (t, s)$. Obviously, the reduction takes log-space. Since H is a DAG, then the only possible circle in H' occurs when there's a path in H from s to t. As a result, finding the circle in H' is equivalent to finding the path from s to t in H. So, Graph Cyclicity is NL-hard.

To sum up, Graph Cyclicity is NL-complete.

c)
As we can see, Graph Acyclicity is the complement problem of Graph Cyclicity. Since Graph Cyclicity is NL-complete, from problem 2-a, we can get that Graph Acyclicity is coNL-complete.

From class we know that NL=coNL. Since Graph Acyclicity is coNL-complete, Graph Acyclicity is NL-complete.

Problem 5

a)
First, we can easily verify that $a = b \wedge c$ satisfies the inequalities.

Second, we will prove that the value of a that satisfies the inequalities is unique.

From $a \leq b$ and $a \leq c$, we get $2a \leq b+c$. Since $b+c \leq a+1$, we get $2a \leq b+c \leq a+1$. That is, $a \leq 1$. Since $0 \leq a$, we get $0 \leq a \leq 1$.

Here are all the possible cases:

- Both b and c are 1.

  In this case, from $b + c \leq a + 1$, we get $1 \leq a$. Since $0 \leq a \leq 1$, we know that $a = 1$ is the unique value that satisfies the inequalities.

- At least one of b and c is 0.

  In this case, from $a \leq b$ and $a \leq c$, we know that $a \leq 0$. Since $0 \leq a \leq 1$, we know that $a = 0$ is the unique value that satisfies the inequalities.

To sum up, the value $a = b \wedge c$ is the unique value over real numbers that satisfies the inequalities.

b)
Here is the set of inequalities: $b \leq a, c \leq a, a \leq b + c, a \leq 1$.

First, we can easily verify that $a = b \vee c$ that satisfies the inequalities.

Second, we will prove that the value of a that satisfies the inequalities is unique.

From $b \leq a$ and $c \leq a$, we get $b + c \leq 2a$. Since $a \leq b + c$, we get $a \leq b + c \leq 2a$. That is, $0 \leq a$. Since $a \leq 1$, we get $0 \leq a \leq 1$.

Here are all the possible cases:

- Both b and c are 0.

  In this case, from $a \leq b + c$, we get $a \leq 0$. Since $0 \leq a \leq 1$, we know that $a = 0$ is the unique value that satisfies the inequalities.

- At least one of b and c is 1.

  In this case, from $b \leq a$ and $c \leq a$, we know that $1 \leq a$. Since $0 \leq a \leq 1$, we know that $a = 1$ is the unique value that satisfies the inequalities.

To sum up, the value $a = b \vee c$ is the unique value over real numbers that satisfies the inequalities.

c)
We can reduce the Circuit Value problem with fan-in 2 to a Linear Inequalities problem.

Here is the reduction:

- For any AND gate $a = b \wedge c$ in circuit C, we will construct the following inequalities: $a \leq b, a \leq c, b + c \leq a + 1, 0 \leq a$.

- For any OR gate $a = b \vee c$ in circuit C, we construct will the following inequalities: $b \leq a, c \leq a, a \leq b + c, a \leq 1$.

Obviously, this reduction takes log-space. Then solving the Circuit Value problem is equivalent to solving the respective Linear Inequalities problem.

And since Circuit Value problem with fan-in 2 is a P-complete problem, Linear Inequalities problem is P-hard under log-space reductions.