Computational Complexity **(Due: Flexible)**

Assignment #4 (Arora Barak Ch - 4)

Instructor: Shreesh Maharaj *Name:* Jimut Bahan Pal

Q. 1. Show that 2SAT is in **NL**

Soln. Since **NL** = **co-NL**, it is sufficient to show that [1, 2] $\overline{2-SAT} \in$ **NL**. We have to demonstrate a logspace reduction from $\overline{2-SAT}$ to **PATH** $\in$ **NL**. So, let **F** $\in$ **2-CNF** have variable set X and literal set L. Let **G** be a graph, for each $l \in L$, create a new copy G_l of G, for each l $\in$ L, create a new copy G_l of G.
V(G_l) = V(G) x $\{l\}$
E(G_l) = {((a,l),(b,l))| (a,b) $\in$ E(G)}
Let s, t be 2 nodes not in $V' = \bigcup_{t \in L} V(G_t)$. Define a graph H by $V(H) = V' \cup s, t$ and
$E(H) = \bigcup_{l \in L} E(G_l) \cup \{(s,(x,x))|x \in X\} \cup \{((\bar{x},x),(x,\bar{x}))|x \in X\} \cup \{((x,\bar{x}),t)|x \in X\}$.
We can arrange G_l into two rows: the first row for those G_x where $x \in X$ and the second row for those $G_{\bar{x}}$ where $x \in X$. s is above the first row and t is below the second row. For each $x \in X$, there is an edge from s to the copy of x in G_x, an edge from the copy of $\bar{x}$ in G_x to the copy of x in G_x, an edge from the copy of $\bar{x}$ in G_x to the copy of $\bar{x}$ in $G_{\bar{x}}$, and an edge from the copy of x in $G_{\bar{x}}$ to t.

We have to show that there is an $x \in X$ and a path from x to $\bar{x}$ in G iff there is a path from s to t in H.

($\Longrightarrow$) Let p be a path in G from x to $\bar{x}$ to x. Then the corresponding path in H goes from s to the copy of x in G_x, follows the isomorphic copy of p until reaching the copy of $\bar{x}$, then takes the edge from the copy of $\bar{x}$ in G_x to the copy of $\bar{x}$ in $G_{\bar{x}}$, then continues to follow the isomorphic copy of p until reaching the copy of x, and finally takes the edge from the copy of x in $G_{\bar{x}}$ to t.

($\Longleftarrow$) If p is a path from s to t in H, then it must first enter G_x for some $x \in X$. By stripping off s, t from p and the edge from G_x to $G_{\bar{x}}$ and removing the second component information in the nodes of p, we obtain a path in G from x to $\bar{x}$ to x.

So, by this there is a reduction from $\overline{2-SAT}$ to $PATH$. The following algorithm shows that H can be computed in log space.

$for\ each\ l \in L\ and\ \{l_1, l_2\} \in F$
  $output\ ((\bar{l_1}, l),(l_2, l)),((\bar{l_2}, l),(l_1, l))$
$for\ each\ x \in X$
  $output\ (s,(x,x)),((\bar{x},x),(x,\bar{x})),((x,\bar{x}),t)$

Q. 2.Prove that the language $SPACETM$ of (4.3) is **PSPACE**-complete.

Soln. Let us check [3, 4] the definition once more, $SPACE\ TMSAT$ = { $< M, w, 1^n >$: DTM M accepts w in space n}. A language L is **PSPACE**-complete if for all L' in PSPACE, $L' \leq_P L$ in **PSPACE**. We can find an essence of **PSPACE** by understanding 2 player games, i.e., can the first/ second player always win?.

Let us understand this via a Quantified Boolean Formula (QBF) game. Let us consider two players, Alice and Adversary, each given n (mutually disjoint) sets of variables numbered from [1,n]. Given a boolean formula over these variables, in the i^{th} round, players set the values of the variables in their i^{th} sets. Say, Alice moves first. When all variables are set, formula is evaluated. Either true Alice wins or Adversary wins. We need to find out whether Alice has a sure-to-win strategy given a QBF game.

Let us see a QBF example consisting of variables $x_1, y_1, x_2, y_2, x_3\ and\ y_3$. The formula will be $\phi(x_1, y_1, x_2, y_2, x_3, y_3)$. Say, there are no variables for adversary, only x_1. The strategy for Alice will be $\exists\ x_1\ \phi(x_1)$ true? Say there are no variables for Alice only y_1, then the strategy for Alice will be $\forall\ y_1\ \phi(y_1)$ true? Say we are having only x_1, y_1 now, and that looks more like a game. Strategy for Alice is $\exists\ x_1\ \forall\ y_1\ \phi(x_1, y_1)$ is true? In general, winning strategy for Alice exists iff, $\exists\ x_1\ ,\forall\ y_1, ...\exists\ x_n\ \forall\ y_n.\phi(x_1, y_1, ..., x_n, y_n)$ is true, else adversary has a winning strategy.

True quantified boolean formula can be written as: $\psi = \exists x_1\ \forall y_1 ... \exists x_n \forall y_n . \phi(x_1, y_1, ..., x_n, y_n)$.
TQBF = $\{\psi | \psi\ is\ true\}$, e.g., $\psi_1 : \exists x \forall y(x = y).\psi_2 : \forall y \exists x(x = y)$. We will show that TQBF is in **PSPACE**. For this we need to consider when is QBF true? e.g., $\exists a, b,\ \forall c \phi(a, b, c)$. We need to ask the winning strategy from each node. We need to put yes from $\exists$ node if yes from either of the child, and yes from $\forall$ node if we

have yes for both of the child. Naive evaluation takes exponential space and time, but we canuse the left child's computation space for the right child. Space needed $= O(depth) + forevaluation =$ poly($|QBF|$).

Now we will show that TQBF is **PSPACE**-hard. For L in **PSPACE** (i.e., $TM\ M_L$ decides L in space $poly(n)$, or with configs of size $s(n) = poly(n)$), we need to show that $L \leq_p TQBF$. Given x, we need to output $f(x) = \psi$, such that ϕ is true iff M_L accepts x. We can $x \longrightarrow \psi$ in poly time. In particular size of ψ is $poly(n)$. As in Cook's theorem, we can build an unquantified formula ϕ (even 3CNF) such that ϕ is true iff M_L accepts x. But the size is poly (time bound on M_L) $= exp(n)$.

An exponential QBF can be written as $\exists\ c_1, c_2, ...c_T\ \psi_o(c_{start}, c_1) \wedge\ \psi_o(c_1, c_2) \wedge ...\psi_o(c_T, c_{accept})$. Here c_i are variables whose value assignment corresponds to configurations $|c_i| = O(S(n))$, $|\psi_o(c, c')| = o(s(n))$, $T = 2^{(O(S(n)))}$. $\psi_o(c, c')$ is an unquantified formula (only variables being c, c'), such that, it is true iff c evolves c' in one step. F be the constant sized formula to derive each bit of new configuration from a few bits in the previous configuration. $\psi_o(c, c')$ is conjunction of equality conditions enforcing consistency with F. $|\psi_o(c, c')| = O(|c|)$. $\psi_o(c, c') = \bigwedge_j (c'^{(j)}\ =\ F(c^{(j-c)}, ..., c^{(j+c)}))$.

ψ: A partially quantified boolean formula ψ_i such that $\psi_i(c, c')$ is fully quantified and is true iff c' is reachable from c in the configuration graph $G(M_L, x)$ within 2^i step, output $\psi\ =\ \psi_s(n)(start, accept)$. Base case (i=0): an unquantified formula, ψ_o. $\psi_{i+1}(c, c')\ =\ \exists c"\ \psi_i(c, c") \wedge \psi_i(c", c')$. This needs to be written in Prenex Normal form. $\psi_{i+1}(c, c')\ =\ \exists c" \psi_i(c, c") \wedge \psi_i(c", c')$. There is a problem with this, $|\psi_{s(n)}|$ is exponential in $s(n)$ and we need more variables/quantification to "reverse" formula.

From Savitch's theorem, we get,

$\psi_{i+1}(C, C')\ =\ \exists C" \forall (D, D").((D, D')\ =\ (C, C") \vee (D, D') = (C", C')) \Rightarrow \psi_i(D, D')$.
$|\psi_{s(n)}| = O(s(n)) + |\psi_{s(n)-1}| = O(s(n)^2) + |\psi_o| = O(S(n)^2)$.

So, we can say that $TQBF$ is **PSPACE**-complete.

Disclaimer: All the answers are collected from the internet, and are not the author's creation.

References

[1] Abhijit Das. Chapter3 : Space complexity: Solutions of the exercises, 2004. https://cse.iitkgp.ac.in/~abhij/course/theory/CC/Spring04/soln3.pdf last accessed June 21, 2020.

[2] Chris Calabro. Cse 200 homework 5 solutions, 2006. http://cseweb.ucsd.edu/classes/wi06/cse200/hw5.pdf last accessed June 21, 2020.

[3] Yuval Filmus (https://cs.stackexchange.com/users/683/yuval filmus). Proving that the language space tm-sat is pspace-complete? Computer Science Stack Exchange. URL:https://cs.stackexchange.com/q/16768 (version: 2013-11-06).

[4] Manoj M. Prabhakaran. Computational complexity lecture 5: in which we relate space and time,and see the essence of pspace (tqbf), 2009. https://courses.engr.illinois.edu/cs579/sp2009/slides/CC-S09-Lect05.pdf last accessed June 21, 2020.