Recitation 7

John Chilton

July 26, 2007

Problem 7.12

$$a^b \equiv c(modp)$$

$$\Downarrow\Downarrow\Downarrow$$

$$a^b \mod p == c \mod p$$

Problem 7.12: Using $s * t \mod q = (s \mod q)(t \mod q) \mod q$

$$a^b \mod p = (a \mod p) * (a^{b-1} \mod p) \mod p$$

Why isn't this polynomial time?

Problem 7.14

$$w_1 w_2 w_3 w_4 w_5 w_6 w_7 w_8 \dots w_{n-2} w_{n-1} w_n$$

$$(w_1 w_2 w_3)(w_4 w_5 w_6 w_7 w_8) \dots (w_{n-2} w_{n-1} w_n)$$

$$w_{1\dots3}, w_{4\dots8}, \dots, w_{n-2\dots n} \in A$$

2^n possible decompositions of w, exponential in the size of w. Use “dynamic programming“ to store the answer to recurring subproblems.

Problem 7.14: A_i indicates whether $w_{1 \dots i}$ be decomposed so that $w_{1 \dots i} \in A^*$.

i	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	$\dots$	w_{n-1}	w_n
A_i	0	1	1	0	1	0	?		$\dots$		

1. Figure out how to determine A_i from $A_1, A_2, \dots A_{i-1}$ and w in polynomial time.
2. Loop from $i = 1, 2, \dots, n$.
3. Use A_n to determine if $w \in A^*$

(Also take care of the special case)

Problem 7.16: You can use another somewhat similar DP algorithm to decide UNARYSSUM in polynomial time.

Problem 7.17: $A \neq \{\}$ and $A \neq \Sigma^*$. So for any such A there exists strings x, y such that $x \in A$ and $y \notin A$.

Problem 7.23

Show $CNF_2 \in P$. Don't need try possibilities, describe how to simplify such a formula.

Problem 7.24 ($\neq SAT$)

Part b gives you a polynomial time mapping, you MUST argue why it works. Should probably have two directions.

$MOSTLYSAT = \{< \phi > \ \mid \phi$ is a cnf-formula where at least all but one of the clauses are simultaneously satisifable $\}$

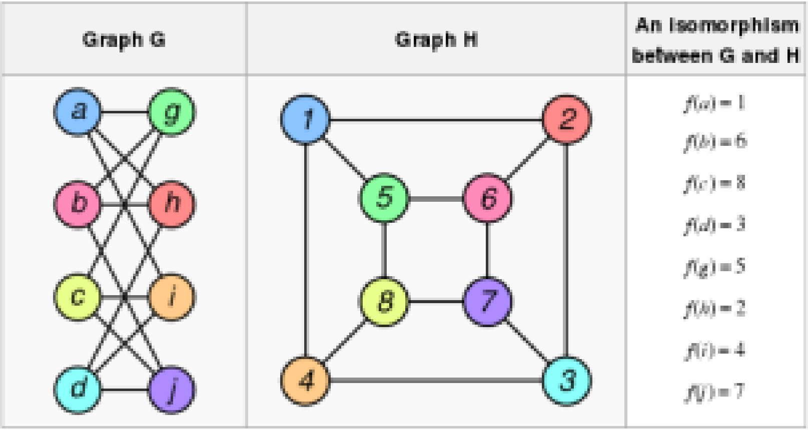
Show $MOSTLYSAT$ is NP-Complete.

$S =$ On input $< \phi >$

1. Let y be a variable not already in ϕ
2. Output $\phi \ \wedge \ y \ \wedge \ \overline{y}$

S is clearly runs in polynomial time, and ϕ is satisifable iff $\phi \ \wedge \ y \ \wedge \ \overline{y}$ is mostly satisifiable. Hence $3SAT \leq_P MOSTLYSAT$.

$MOSTLYSAT \in$ NP (show) and $3SAT \leq_P MOSTLYSAT$, hence $MOSTLYSAT$ is NP-Complete.



http://en.wikipedia.org/wiki/Graph_isomorphism

$SUBGRAPHISO = \{< G, H > \mid G$ is isomorphoric to a subgraph of $H\}$

Show that $SUBGRAPHISO$ is NP-Complete.

A complete graph is one where every node is connected by an edge to every other. A graph with a k-CLIQUE, has such a subgraph. So the following show $CLIQUE \leq_p SUBGRAPHISO$.

$$\begin{aligned} S = \quad & \text{On input } < G, k > \\ & \text{1. Create a complete graph with } k \text{ nodes, } H_k. \\ & \text{2. Output } < H_k, G > \end{aligned}$$

Problem 7.28.

$$SETSPLITTING = \{(S, C = \{C_i\}) \mid C_i \subseteq S \;\&\; (S, C) \text{ colorable}\}$$

S can be colored if each element can be chosen to be *red* or *blue* such that no C_i's elements are all the same color.

Coloring Examples

$$S = \{1, 2, 3, 4, 5\}$$

- C = $\{\{1,2\}, \{3,5\}\} \in$ *SETSPLITTING*
- C = $\{\{1,2,3\}, \{3,4,5\}\} \in$ *SETSPLITTING*
- C = $\{\{1,2\}, \{1,3\}, \{1,4\}, \{2,4\}\} \notin$ *SETSPLITTING*
- C = $\{\{1,2,3\}, \{2,3\}, \{1\}\} \notin$ *SETSPLITTING*

Show $SETSPLITTING$ is NP-complete, i.e. give a polynomial time reduction of some NP-complete problem to $SETSPLITTING$.

Some NP-complete problems SAT, $3SAT$, $HAMPATH$, $UHAMPATH$, $CLIQUE$, $VERTEXCOVER$, $SUBSETSUM$.

$$Variables = \{x_1, x_2, x_3\}$$

$$\phi = \{x_1 \vee x_2 \vee x_3\} \wedge \{\bar{x_2} \vee x_2 \vee x_3\} \wedge \{\bar{x_1} \vee x_3 \vee \bar{x_3}\}$$

$$\Downarrow\Downarrow\Downarrow$$

$$S = \{x_1, \bar{x_1}, x_2, \bar{x_2}, x_3, \bar{x_3}\}$$

$$C = \{\{x_1, x_2, x_3\}, \{\bar{x_2}, \bar{x_2}, \bar{x_3}\}, \{\bar{x_1}, x_3, \bar{x_3}\}\}$$

$$\Downarrow\Downarrow\Downarrow$$

$$S = \{\color{red}{x_1}, \color{blue}{\bar{x_1}}, \color{blue}{x_2}, \color{red}{\bar{x_2}}, \color{red}{x_3}, \color{blue}{\bar{x_3}}\}$$

$$C = \{\{\color{red}{x_1}, \color{blue}{x_2}, \color{red}{x_3}\}, \{\color{red}{\bar{x_2}}, \color{red}{\bar{x_2}}, \color{blue}{\bar{x_3}}\}, \{\color{blue}{\bar{x_1}}, \color{red}{x_3}, \color{blue}{\bar{x_3}}\}\}$$

A subset of nodes of a graph is a *dominating set* if every other node is adjacent to some node in the subset.

$DOMINATE = \{< G, k > \mid G \text{ has a dominating set with } k \text{ nodes } \}$

Show that *DOMINATE* is NP-complete by a giving a reduction from 3*SET* or *VERTEXCOVER*.