School of Mathematical and Computational Sciences
Indian Association for the Cultivation of Science

Master's/Integrated Master's-PhD Program/ Integrated Bachelor's-Master's Program/PhD Course

Theory of Computation II: COM 5108

Tutorial V (14 September 2023)

Instructor: Goutam Biswas *Autumn Semester 2023*

1. Consider the following languages and answer whether they belong to **P** or they are **NP**-complete?

   (i)
   $$SAT_e = \{\phi\ 0\ 1^{2^n} : \phi \in SAT \text{ and } |\phi| = n\}.$$

   (ii)
   $$SAT_p = \{\phi\ 0\ 1^{n^c} : \phi \in SAT,\ n = |\phi|, \text{ and } c \text{ is a constant}\}.$$

   Ans.

   (i) The membership of $x \in SAT_e$ can be tested using the truth table of the boolean formula ϕ. The size of the truth table is $O(2^n)$. So time taken is linear in the length of the input and $SAT_e \in \mathbf{P}$.

   (ii) SAT_p certainly belongs to **NP**. A satisfying assignment of the variables of ϕ is a certificate.
   It is **NP** complete as SAT can be reduced to SAT_p in polynomial time.
   From the length n of a boolean formula ϕ, n^c can be computed in polynomial time and those many 1's can be augmented.

2. If $\mathbf{P} = \mathbf{NP}$ and $L \in \mathbf{P}$, but $L \neq \phi$ or Σ^*, then prove that L is **NP**-complete.

   Ans. Let $L' \in \mathbf{NP} = \mathbf{P}$ is decided by a polynomial time bounded DTM M, and $\exists x_0, x_1 \in \Sigma^*$ such that $x_0 \in L$ and $x_1 \notin L$ ($L \neq \phi$ or Σ^*). The polynomial time reduction machine is as follows.

   N: "Input x:

   (i) Run M on x.
   (ii) If M comes to 'Y' halt, return x_0.
   (iii) If M comes to 'N' halt, return x_1.

3. UNARY-SSUM = $\{< S, t >$: $S = \{x_1, \cdots, x_k : x_i \in \mathbb{N}$, where x_i's are represented as unary numerals $\}$ is a multiset and for some $\{y_1, \cdots, y_l\} \subseteq S$, $\sum_{i=1}^{l} = t\}$.

   (i) Is UNARY-SSUM in **NP**?
   (ii) Is UNARY-SSUM **NP**-hard?

Ans.

(i) The certificate $C \subseteq S$ where $< S, t >$ is the input, will also be in unary. So it is polynomial over the length of the input. Verifying $C \subseteq S$ and $\sum C = t$ can be done in polynomial time. So UNARY-SSUM is in **NP**.

(ii) The reduction from SUBSET-SUM to UNARY-SSUM cannot be done in polynomial time as there will be exponential increase in the length to convert a radix $b \geq 2$ numeral to an unary numeral. So UNARY-SSUM is not **NP**-hard?

4. In the proof of Cook-Levin theorem a window of size 2×3 was used to establish the correctness of transition from configuration C_i to C_{i+1}. Justify that it cannot be done using window of size 2×2.

   Ans. Consider two transitions $\delta(p, 0) = \{(q, 1, \leftarrow), \cdots\}$ and $\delta(p, 1) = \{(q, 0, \rightarrow), \cdots\}$. We have a portion of $C_i = \cdots 0\ 1\ 0 \cdots$ and the corresponding cells in $C_{i+1} = \cdots 0\ q\ 0 \cdots$. A 2×3 cell can immediately detect its invalidity due to the appearance of a state symbol at the middle $\frac{0\,1\,0}{0\,q\,0}$.
   But a 2×2 window will look only at (i) $\frac{0\,1}{0\,q}$ and (ii) $\frac{1\,0}{q\,0}$.
   Window (i) can be extended to $\frac{0\,1\,p\,0}{0\,q\,1\,*}$ making it valid.
   Window (ii) can be extended to $\frac{p\,1\,0}{0\,q\,0}$ making it valid.

5. Give a polynomial time reduction of 3COL to SAT. What is the time complexity of the reduction.

   Ans. Let the graph $G = (V, E)$ has n vertices $V = \{v_1, \cdots, v_n\}$. We take $3n$ variables, $\{x_{11}, x_{12}, x_{13}, \cdots, x_{n1}, x_{n2}, x_{n3}\}$. The variable $x_{ij},\ 1 \leq i \leq n,\ 1 \leq j \leq 3$, is *true* if the vertex v_i is coloured with the colour j. We have the following set of clauses:

   (a) Each vertex must have at least one colour:

   $$\bigwedge_{i=1}^{n} (x_{i1} \vee x_{i2} \vee x_{i3}).$$

   (b) No vertex can have two colours:

   $$\bigwedge_{i=1}^{n} ((\neg x_{i1} \vee \neg x_{i2}) \wedge (\neg x_{i2} \vee \neg x_{i3}) \wedge (\neg x_{i3} \vee \neg x_{i1})).$$

   (c) No pair of adjacent vertices can have same colour.

   $$\bigwedge_{\{v_a, v_b\} \in E} \bigwedge_{j=1}^{3} (\neg x_{aj} \vee \neg x_{bj}).$$

   The length is of $O(n) + O(n) + O(n^2) = O(n^2)$.