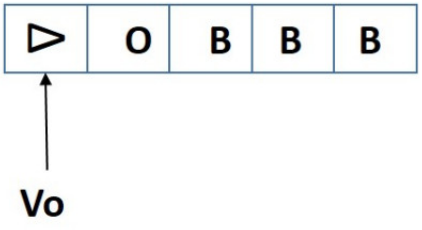
Advanced Algorithms and Complexity : Lecture 4 Cook Levin Theorem (Satisfiability is NP-Complete)

August 10, 2018

A simplified reduction for $L \in P$: $L \in P \implies |y| = 0$.
M will take as input x and will simulate $D(x)$ in polynomial time and will generate an encoding of computation done by D on input x as follows:
The encoding is explained with the help of the example DTM for L_e that we have considered earlier.
Alphabet and states are encoded in binary:
We have $\Gamma = \{\triangleright, B, 0, 1\}$. We assign binary codes to alphabet symbols as follows:
$\triangleright \to 00, B \to 11, 0 \to 01, 1 \to 10$.
We have $Q = \{q_0, q_h, q_1, q_2\}$. We assign binary codes to states as follows: $q_0 \to 00, q_h \to 11, q_1 \to 01, q_2 \to 10$
Suppose for the example input we take $x = 0$:

We define snapshot as the pair of the state q and symbol a such that the TM is in state q scanning the symbol a.
For this example, the sequence of snapshots are:
$z_1 = (q_0, \triangleright) = 0000$
$z_2 = (q_1, 0) = 0101$
$z_3 = (q_1, B) = 0111$



$z_4 = (q_2, 0) = 1001$
$z_5 = (q_h, 0) = 1101$
With the input as $\rhd 0 = 0001$. For encoding the computation on input x, we list the encoding of x followed by encoding of $Z_i^{'s}$:
$x, z_1, z_2 \ldots.., z_{T(n)}$
For the given example, the encoding will be:

$\underline{0001}(x)\underline{0000}(z_1)\underline{0101}(z_2)\underline{0111}(z_3)\underline{1001}(z_4)\underline{1101}(z_5)$

The encoding is of polynomial size because the DTM runs in polynomial-time $T(n)$. M will give as output a Boolean formula in which there will be variables corresponding to each bit position. Formula will be AND of four CNF formulas in which the first formula will represent the fact that the input is correct, the second formula will represent the fact that the DTM starts correctly, the third formula will represent the fact that the computation is correct (according to the transition function), and the fourth formula will represent the fact that the DTM halts in a halting (accepting state). For the given example, the formula will look something like:

$$\underbrace{(x = \rhd 0)}_{1} \wedge (\underbrace{z_1 = (v_{0,} \rhd)}_{2}) \wedge \underbrace{(z_1 \text{->} z_2) \wedge (z_2 \text{->} z_3)} \wedge$$
$$\underbrace{(z_3 \text{->} z_4) \wedge (z_4 \text{–>} z_5)}_{3} \wedge (\underbrace{z_5 = (v_{h,} \underline{\quad})}_{4}$$

CNF for formulas $1, 2$ and 4 are easy. For example, CNF for formula 1 will be :
consider the identity:

$$0 = (\underset{0}{x_{11}} \vee \underset{0}{x_{12}} \vee \underset{0}{x_{21}} \vee \underset{0}{x_{22}}) \wedge (\underset{0}{x_{11}} \vee \underset{0}{x_{12}} \vee \underset{0}{x_{21}} \vee \underset{1}{\overline{x_{22}}}) \wedge (\,) \ldots.$$

where we have considered all possible combinations of four variables. Each clause is 0 for exactly one combination (shown below). If we remove the clause corresponding to $0001(x_{11} \vee x_{12} \vee x_{21} \vee \bar{x}_{22})$,
We will get the CNF for formula 1: $(x = \triangleright 0)$
CNF formula for 3 is also similar. For example, for $z_1 \to z_2$, we will get the CNF by removing the clause corresponding to $z_1 z_2 = 00000101$:
$() \wedge () \wedge .... \wedge (z_{11} \vee z_{12} \vee z_{13} \vee z_{14} \vee z_{21} \vee \bar{z}_{22} \vee z_{23} \vee \bar{z}_{24})...$

Reduction for the case of $L \in NP$: The difficulty for this case is that M knows only x. It has no idea of y. We can remove this difficulty by making use of *Oblivious* TM'^s. An oblivious TM is a TM for which the head position is independent of the input x, it only depends on the input length $|x|$. For example, the DTM for L_e is oblivious TM. Given any DTM running in time $T(n)$, we can create an equivalent oblivious DTM running in time $T(n)^2$. With this simplification, M can simulate $D(x, 0^{p(|x|)})$ and determine the head position at step i, and also the last step at which the current cell content is modified. In polynomial-time M can determine functions $f(i)$ and $g(i)$ such that $g(i)$ is the head position at step i, and $f(i)$ is the number of step j such that in j^{th} step, the head was at the same position as the i^{th} step $(j < i)$. If there is no such j then $f(i) = 0$. $f(1) = 0, f(2) = 0, f(3) = 0, f(4) = 2, f(5) = 4$.
For the example TM, since it does not modify any cell location, we have: $g(1) = 1, g(2) = 2, g(3) = 3, g(4) = 2, g(5) = 2$. For this case, the encoding will be:

$$\underbrace{x_{11} \;....\; x_{1d}}_{x_1} \quad \underbrace{x_{21} \;....\; x_{2d}}_{x_2} \quad \underbrace{x_{n1} \;....\; x_{nd}}_{x_n}$$

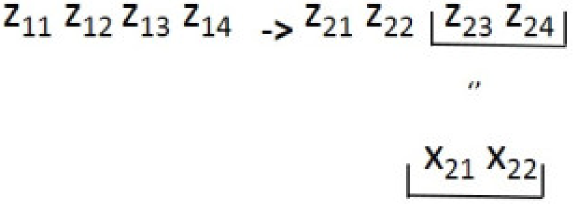
$$\underbrace{y_{11} \;....\; y_{1d}}_{y_1} \quad \underbrace{y_{21} \;....\; y_{2d}}_{y_2} \quad \underbrace{y_{p(n)1} \;....\; y_{p(n)d}}_{y_{p(n)}}$$

$$\underbrace{z_{11} \;....\; z_{1c}}_{z_1} \quad \underbrace{z_{21} \;....\; z_{2c}}_{z_2} \quad \underbrace{z_{T(n)1} \;....\; z_{T(n)c}}_{z_{T(n)}}$$

where $d =$ number of bits used to encode Γ
$c =$ number of bits used to encode Γ and Q.
For the given example, we have $d = 2$ and $c = 4$.
As before, the output formula Φ will be AND of four CNF formulas:
$\Phi = \Phi_1 \wedge \Phi_2 \wedge \Phi_3 \wedge \Phi_4$
$\Phi_1 \equiv$ CNF for the the fact that input is x.
$\Phi_2 \equiv$ CNF for the the fact that TM starts correctly $(q_0, \triangleright x)$.
$\Phi_3 \equiv$ CNF for the the fact that computation is correct (according to transition function).
$\Phi_4 \equiv$ CNF for the the fact that TM halts and accepts in state q_h
As before, Φ_1, Φ_2 and Φ_4 are easy to determine.
For Φ_3, we make use of functions f and g as follows:
Consider the encoding of transition function:

	z_i	z_{i+1}
$\delta(q_0, \triangleright) = (q_1, \triangleright, R)$	0000	01 _ _
$\delta(q_1, 0) = (q_1, 0, R)$	0101	01 _ _
$\delta(q_1, 1) = (q_1, 1, R)$	0110	01 _ _
$\delta(q_1, B) = (q_2, B, L)$	0111	10 _ _
$\delta(q_2, 0) = (q_h, 0, S)$	1001	11 _ _

If $f(i+1) \neq 0$, then we can find the unknown values from $Z_{f(i+1)}$, otherwise we have to use $(x, y)_{g(i+1)}$. Example CNF for $(z_1 \rightarrow z_2)$: We have $f(2) = 0 \implies$
We have to use $g(2) = 2$. For the unknown we have to substitute the variables $(x, y)_{g(2)}$ [Since part of the input y is unknown, it is taken as variable]. The CNF for $(z_1 \rightarrow z_2)$ we look like:



$\Rightarrow \underline{CNF(z_{11}, z_{12}, z_{13}, z_{14}, z_{21}, z_{22})} \wedge (z_{23} = x_{21}) \wedge (z_{24} = x_{22})$
where the underlined CNF formula is obtained by removing from the CNF formula with all possible combinations (= 0) the corresponding classes from the transition function (000001,010101,011110,100111). We can convert the remaining part in CNF as follows:
$(z_{23} \vee \bar{x}_{21}) \wedge (\bar{z}_{23} \vee x_{21}) \wedge (z_{24} \vee \bar{x}_{22}) \wedge (\bar{z}_{24} \vee x_{22})$.