Carlos Mc Gregor c.mcgregormuro@mail.utoronto.ca

CSC373S: Algorithm Design, Analysis & Complexity

Lecture 27

Wednesday March 15, 2017
based on notes by Denis Pankratov

Complexity Theory *(cont.)*

$L \in NPC$ implies that:

1. $L \in NP$
2. $(\forall \tilde{L} \in NP)(\tilde{L} \leq_p L)$

Definition: A **Boolean** circuit is a *directed acyclic graph* so that:

1. Leaves are labelled with variables, e.g. $x_1, .., x_n$
2. Internal nodes are labelled with $\cap$, $\cup$, $\neg$ in-degree ($\cap$) = in-degree ($\cup$) = 2, in-degree ($\neg$) = 1
3. There is a single output node:

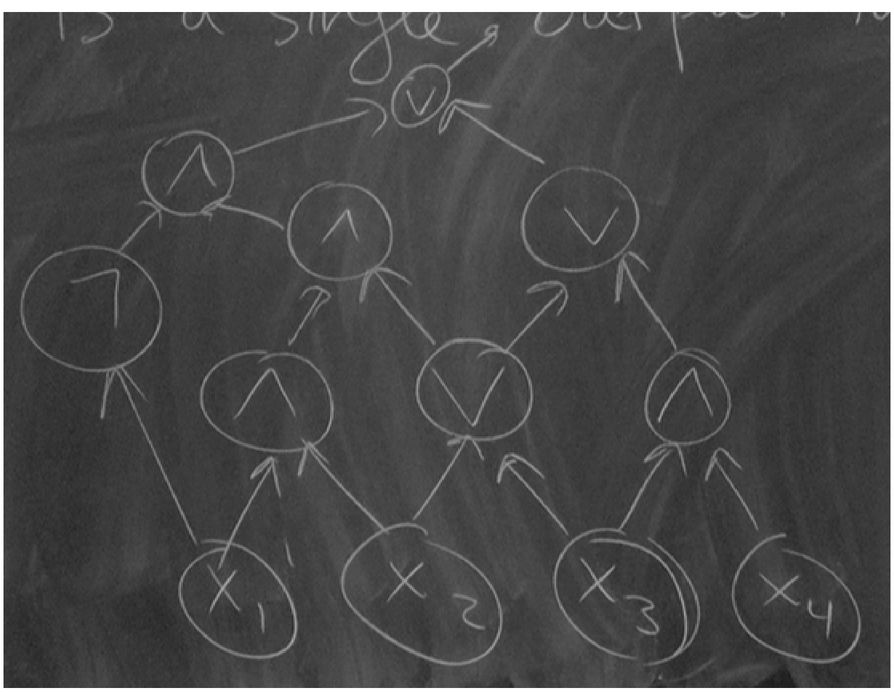

Figure 1: *Example*

Circuit Sat Problem Input: C - circuit
Output: 1 if $\exists$ assignment to vars of C so that C outputs 1 on that assignment
0 otherwise

$L_{C-SAT} = \{< C > | C \text{ is satisfiable}\}$

Theorem (version of Cook's theorem)

$$L_{C-SAT} \in NPC$$

Proof:

1. $L_{C-SAT} \in NP$
   Verifier on input
   $x =< C > -$ circuit
   $y =< \tau > -$ assignment to vars of C

   Evaluates C on τ, returns whatever C returns.
   Verifier runs in *polytime*:
   $< C > \in L_{C-SAT} \Leftrightarrow \in \tau$ verifier outputs 1 on $(< C >, < \tau >)$

2. Let $L \in NP$
   Given x need to construct circuit C efficiently so that $x \in L \Leftrightarrow < C > \in L_{C-SAT}$
   Let A be a polytime verifier for L, $\exists k_1 > 0$ so that $(\forall x)(x \in L \Leftrightarrow \exists y : |y| \leq |x|^{k_1} \cap A(x, y) = 1)$

   Idea: C will have y as inputs & will simulate $A(x, y)$
   Let $T(|x|)$ denote the runtime of A on input $|x|$
   $\exists k_2 > 0$ so that $T(|x|) \leq |x|^{k_2}$ (A is polytime)
   Let c_i be the state of the computer running A on input (x, y)

c_i :	Code of A	Program counter	x	y	memory
	constant	constant	$\|x\|$	$\|x\|^{k_1}$	$\|x\|^{k_2}$

   Let M be the circuit implementing a single step (processor): c_i to c_{i+1}

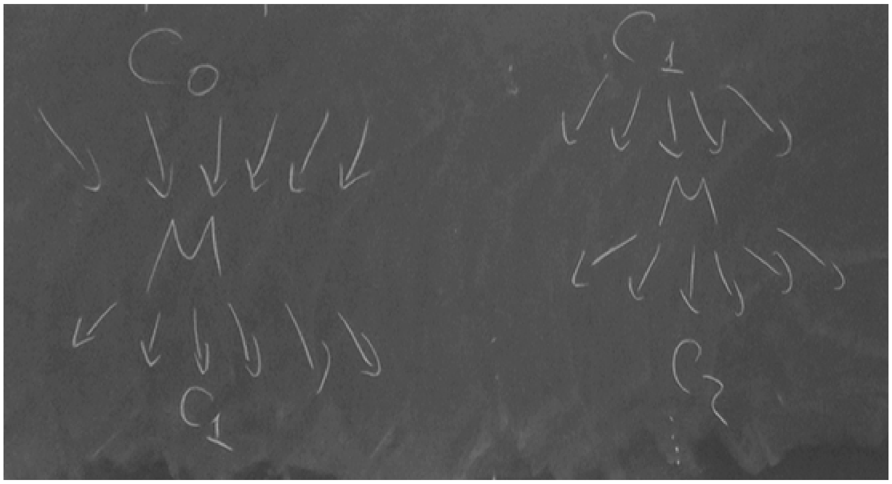

Figure 2: We stack $T(|x|)$ copies of M

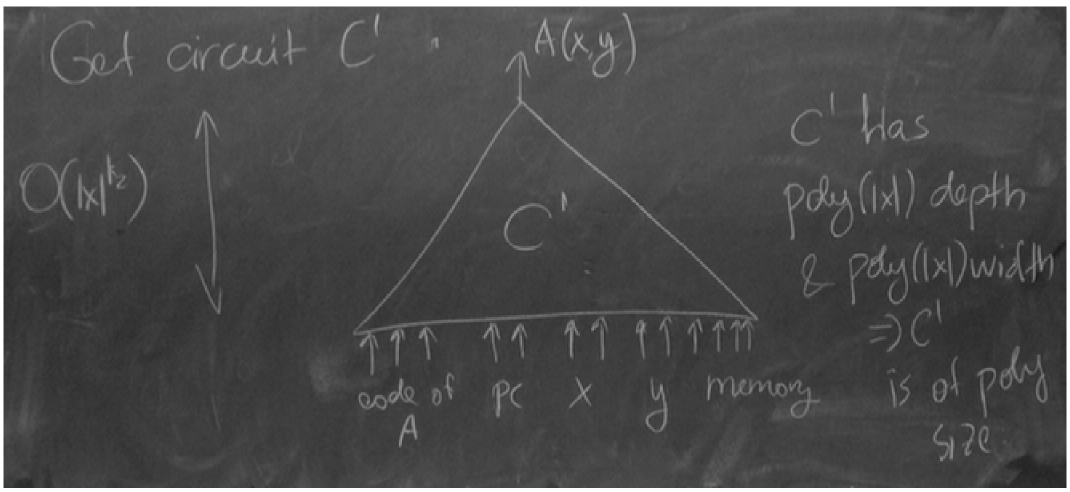

Figure 3: Get circuit C': what happens when we unroll the process with time

C' has poly($|x|$) depth & poly($|x|$) width $\Rightarrow$ C' is of poly size

To get C:

$$\text{applying a restriction} \begin{cases} \text{hard-code } x \\ 0's \text{ for memory} \\ 0's \text{ for PC} \\ \text{code of } A \text{ to code of } A \end{cases}$$

$x \in L \Leftrightarrow \exists y : |y| \leq |x|^{k_1}$ and $A(x,y) = 1$
$\Leftrightarrow \exists y C(y) = 1$
$\Leftrightarrow < C > \in L_{C-SAT}$

Given x, you can construct C in *polynomial time*

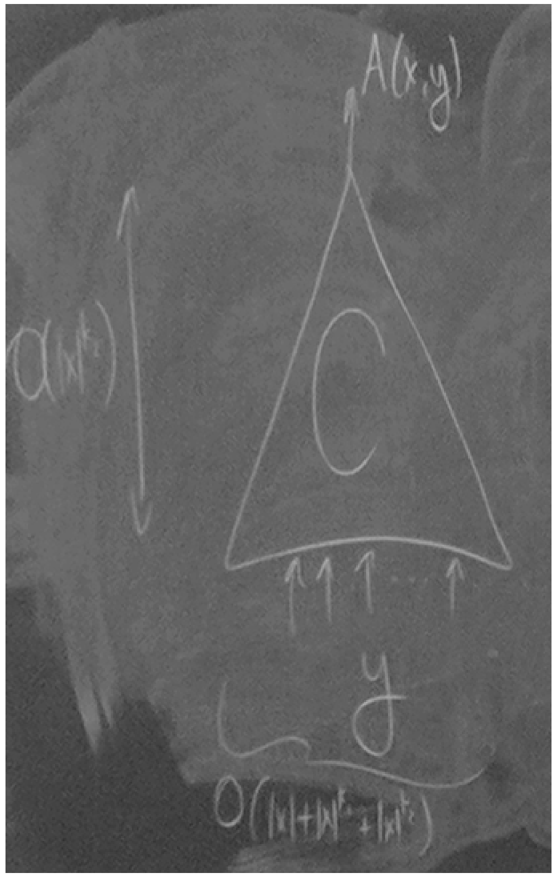

Figure 4: *Visualization*

Note:
$\mathcal{O}(|x|^{k_2})$ with vertical arrow
$\mathcal{O}(|x| + |x|^{k_1} + |x|^{k_2})$ with horizontal arrow

Exercise: Show $\leq_p$ is transitive if $L_1 \leq_p L_2$ & $L_2 \leq L_3$ then $L_1 \leq_p L_3$
Claim: If $L_1 \in NPC$ & $L_1 \leq_p L_2$, then L_2 is NP-hard. Moreover, if $L_2 \in NP$, then $L_2 \in NPC$

Proof: Let $\tilde{L} \in NP$, then $\tilde{L} \leq_p L_1 (l_1 \in NPC) \Rightarrow$ by transitivity $\tilde{L} \leq_p L_2$

Claim: $L_{3-SAT} \in NPC$
Proof:

1. $L_{3-SAT} \in NP$ (last lecture)
2. $L_{C-SAT} \leq_p L_{3-SAT}$
   Need to show (NTS): given circuit C, we can construct in *polytime* formula φ so that:
   - φ is $3-CNF$
   - C is satisfiable $\Leftrightarrow \varphi$ is satisfiable

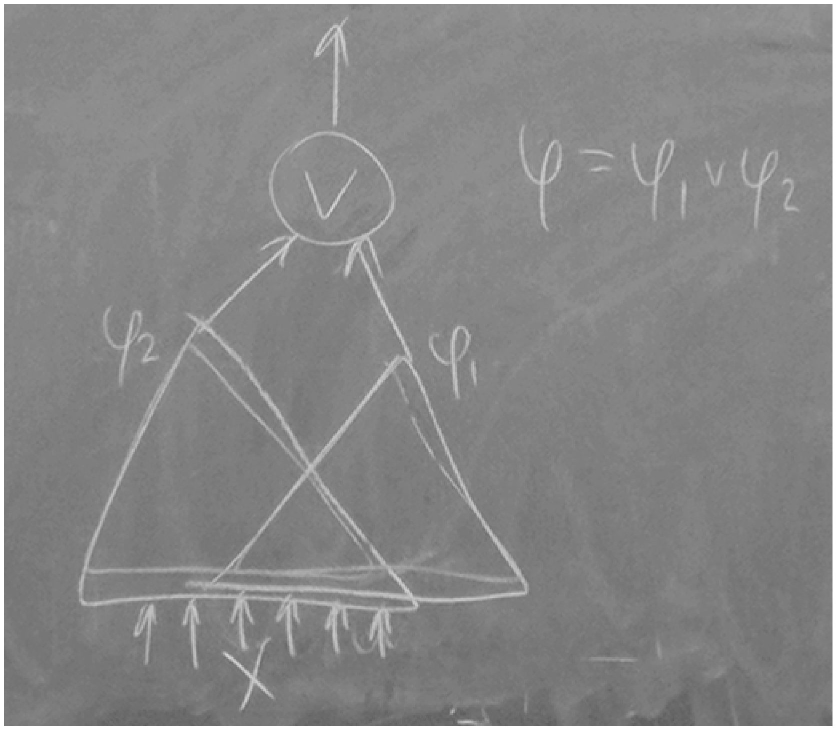

Figure 5: *Visualization*

(1) Keep vars x_i
For each internal node u, introduce a new variable y_u intended meaning y_u has the value of the circuit evaluated on x at gate u

(2) Add clauses:

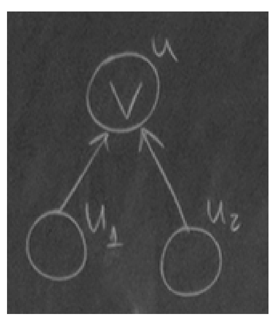

Figure 6: *Clause*

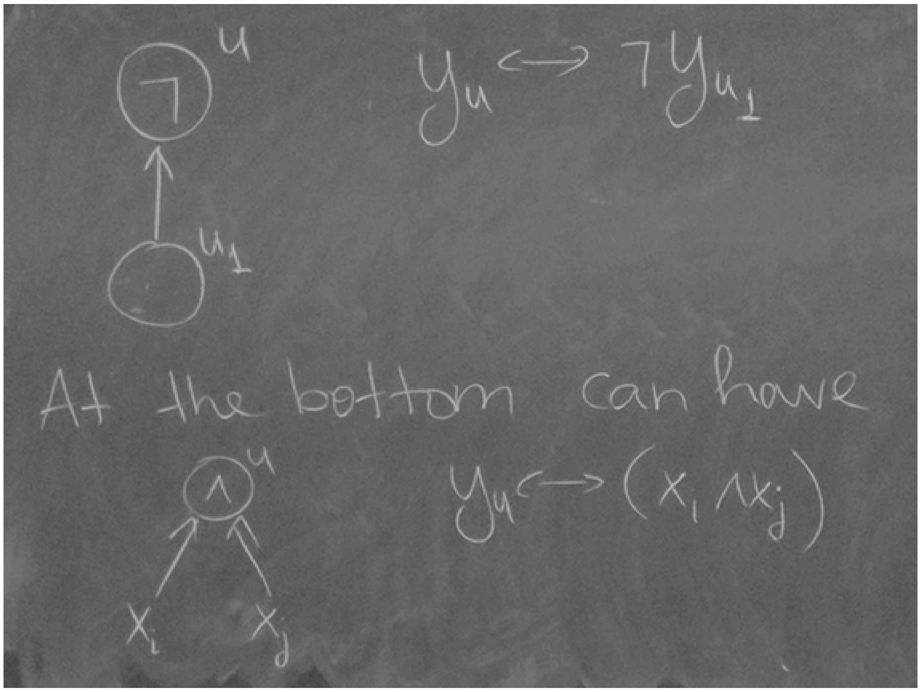

Figure 7: *Clauses*

(3) Add clause y_u where u is the output gate

(4) Convert all clauses into $CNFs$
Example:

y_{u_1}	y_u	$y_u \leftrightarrow y_{u_1}$
0	0	0
0	1	1
1	0	1
1	1	0

$\neg(y_u \leftrightarrow \neg y_{u_1}) \equiv (\neg y_{u_1} \wedge \neg y_u) \vee (y_u \wedge y_u)$