Computational Complexity

Mohammad Mahmoody

4th Session

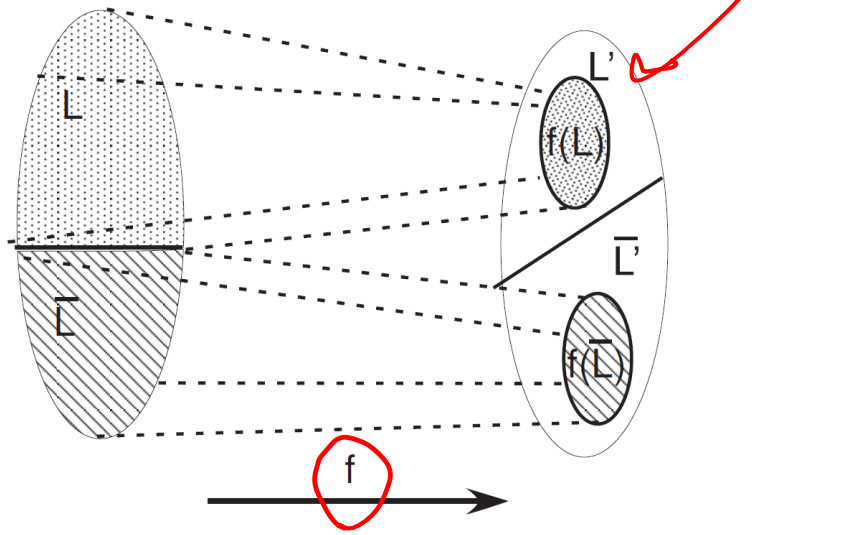
Thurs 23 Jan 2014

About Biweekly Assignments

- first set of assignments: this Saturday (25 Jan)
- Return them on Tuesday 28 Jan or Thursday 30 during office hours (1pm-2pm Rice Hall 511) or at the class (5pm 009 Olsson Hall).
- **After Thursday 5pm**, can only email (mahmoody@gmail.com) a **typed or scanned version** of answers by Monday 5pm (by Friday 5pm 80%, Sat 5pm 60%, Sun 5pm 40%, Mon 5pm 20%)
- Monday 5pm: Abbas (TA) will solve the problems OR sketch of answers will be uploaded + you will have graders
- P.S. 1: Come to the office hours if you have any questions before or after submitting the answers.
- P.S. 2: Any questions, always: email me or post to Piazza

Last Time

- The complexity class **NP**
  - A decision problem for which YES instances have a "witness"
- Notion of (Karp) reduction:
  - $L \leq_p L'$ if and only if there is a function f:
- A language Q is **NP** complete if:
  - $Q \in \mathbf{NP}$
  - $L \leq_p Q$ for all $L \in \mathbf{NP}$ (Q is **NP** hard)



Why do we care about **NP** completeness?

Theorem 2.8 1. *(Transitivity) If* $L \leq_p L'$ *and* $L' \leq_p L''$, *then* $L \leq_p L''$.

2. *If language* L *is* **NP**-*hard and* $L \in \mathbf{P}$ *then* $\mathbf{P} = \mathbf{NP}$.

3. *If language* L *is* **NP**-*complete then* $L \in \mathbf{P}$ *if and only if* $\mathbf{P} = \mathbf{NP}$.

Today

- **NP** complete problems exist
- Natural **NP** complete problems exist (Cook-Levin theorem)
- **NP** complete problems exist … almost everywhere (Karp's paper)
- Why **NP** is called **NP** ? (**N**on-deterministic **P**olynomial time)

Is there any **NP** complete problem?

Theorem 2.9 *The following language is* **NP***-complete:*

$$\text{TMSAT} = \{\langle \alpha, x, 1^n, 1^t \rangle : \exists u \in \{0,1\}^n \text{ s.t. } M_\alpha \text{ outputs 1 on input } \langle x, u \rangle \text{ within } t \text{ steps}\}$$

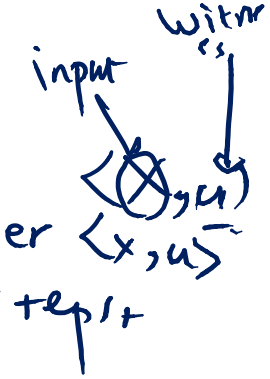
where M_α denotes the (deterministic) TM represented by the string α.[2] ♢

Why TMSAT $\in$ NP?

Claim: $\langle \alpha, x, 1^n, 1^t \rangle \in$ TMSAT

Proof: we use $\mathcal{U}$ (universal) as the "verifier" of TMSAT and u will be the witness.

$\langle \alpha, x, 1^n, 1^t \rangle \in$ TMSAT $\iff$ see def 2.9 $\iff$ when $\mathcal{U}$ runs M_α accepts in # steps over $\langle x, u \rangle$



TMSAT $\in NP$

Claim $\forall L \in NP$ $\quad L \leq_p$ TMSAT.

$\langle \alpha, x, 1^n, 1^t \rangle \in$ TMSAT iff $\exists u$ s.t. M_α accepts (x,u) in $\leq t$ steps.

$L \in NP$

$\Rightarrow$ V_L is a verifier of witnesses for L: $y \in L$ iff $\exists\, u \in \{0,1\}^{m \leq poly(|y|)}$ s.t. $V_L(x,u)$ accepts in $poly(m) \geq t$ steps.

poly-time Turing Machine.

f: reduction function

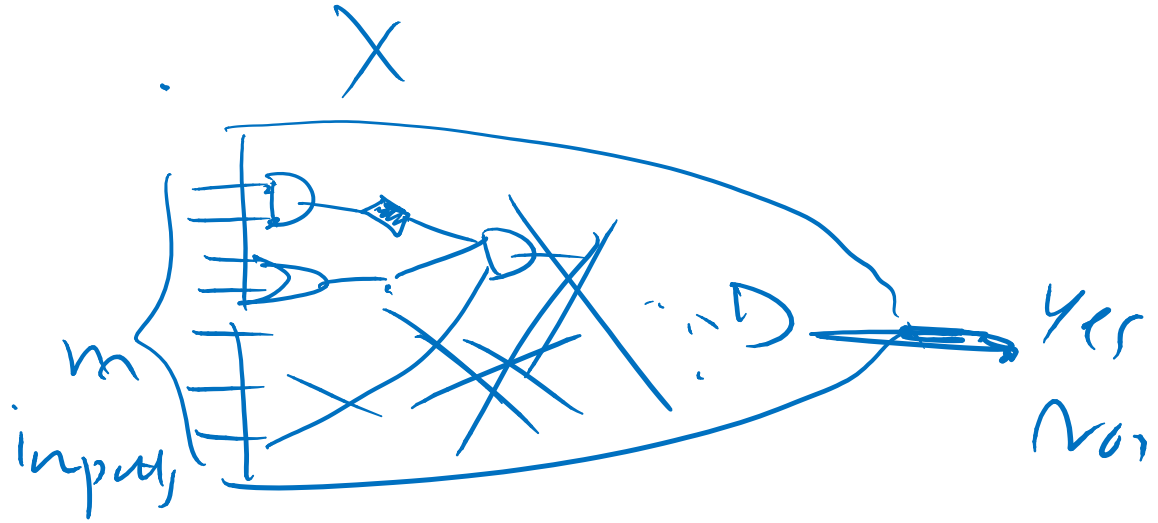
given $y \in L$? $\xrightarrow{f}$ $x \in$? TMSAT.

$x = (V_L, y, 1^n, 1^t)$

Do "natural" **NP** complete problems exist?

- Cook-Levin theorem: CIRC-SAT is **NP** complete.
- Definition of CIRC-SAT problem:

$$\{ C \mid \exists z \; C(z) = 1 \}$$

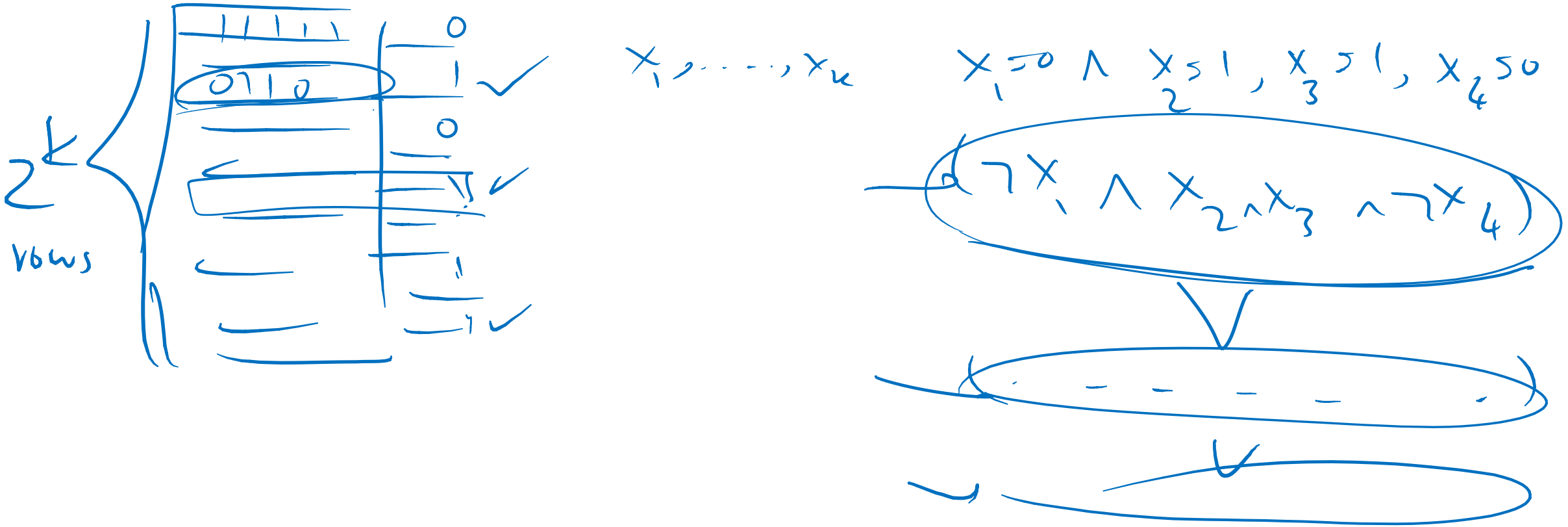


CIRC-SAT $\in$ NP ?

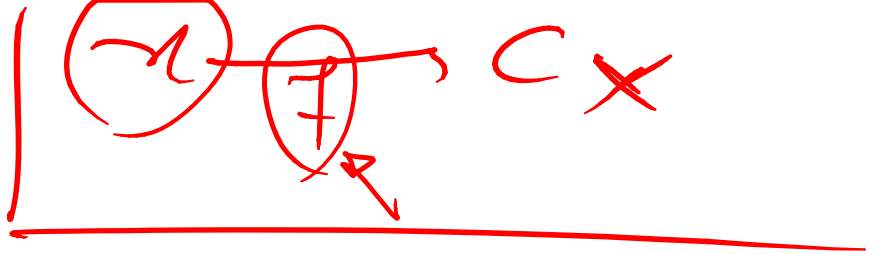
witness (x) verification of (x, C): run C over x.

How Powerful are Boolean Circuits?

- Any function $f : \{0,1\}^k \to \{0,1\}$ can be computed by a circuit C_f of size $O(k \cdot 2^k)$.



Proof of Cook-Levin Theorem: CIRC-SAT is **NP** complete.



CIRC-SAT $\in$ NP ✓

CIRC-SAT is NP-hard.

$L \in$ NP: $\exists$ V_L & $x \in L$ iff $\exists u$ (x,u) make V_L accept.