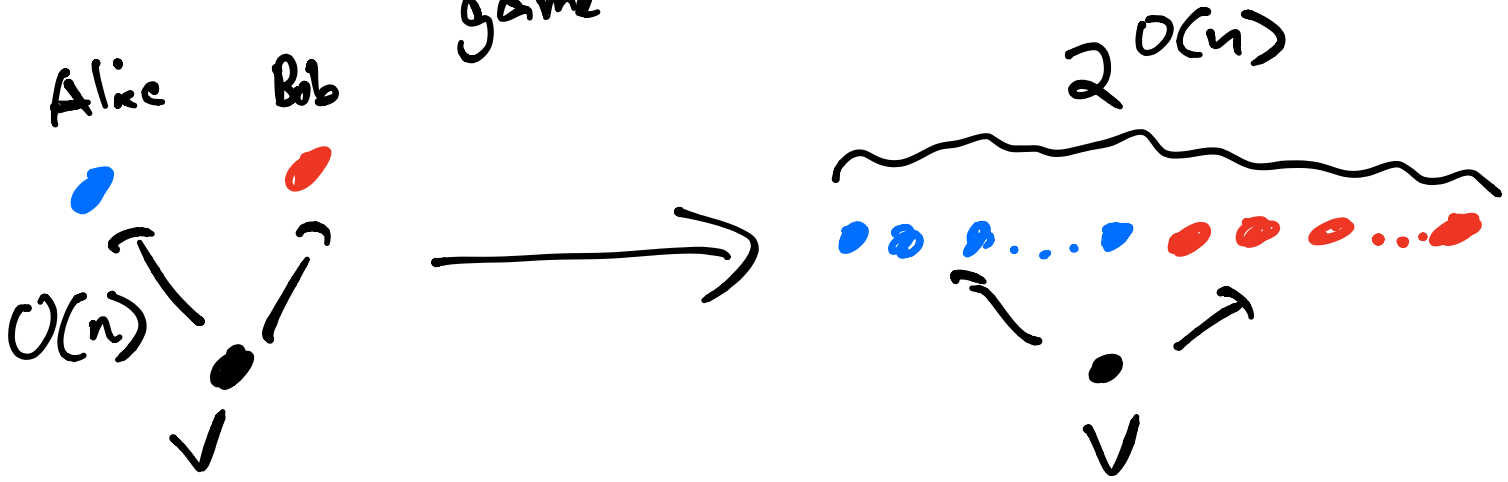
6.S979 Lecture 15

Last time:
- Understand classical MIP

MIP protocol / game $\Longleftrightarrow$ CSP

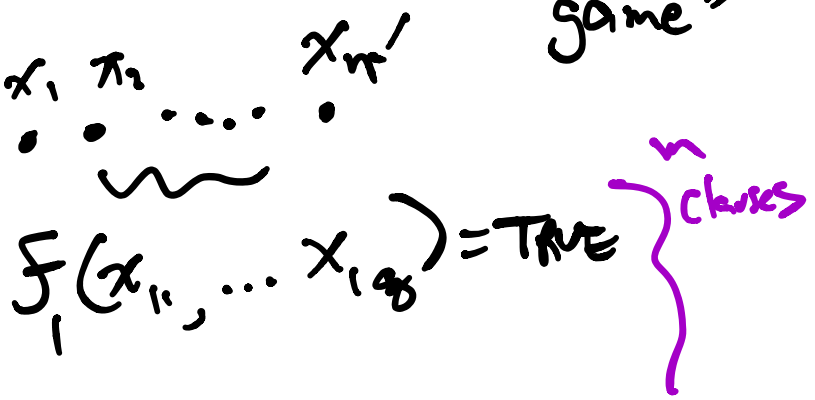


(x,y) question pair $\longrightarrow$ constraint on x,y entries

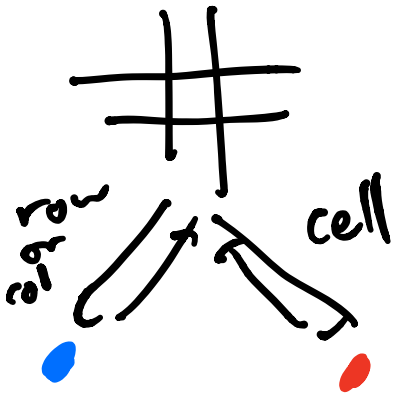
CSP $\longrightarrow$ MIP protocol

$O(\max(\log n, \log m))$ bits

"clause variable game"



Linear systems games



Check that $F_j(a_1 \dots a_q) = \text{TRUE}$ and $a_i = b$

CSP

$\omega(\Phi) :=$ max fraction of constraints that can be satisfied

$\omega(G) :=$ max prob. of success for provers

$$\omega(\Phi) = 1 \implies \omega(G_{CV}) = 1$$

$$\omega(\Phi) \leq \frac{1}{2} \implies \omega(G_{CV}) \leq 1 - \frac{1}{10q^2}$$

CSP on n vars, m clauses

$\longrightarrow$ MIP protocol with messages of size $O(\log(n) + \log(m))$

$MIP = NEXP$

↑ exp. bigger NP

3-coloring on an exp. big graph

For now, think about NP instead of NEXP: want to show

$NP \subseteq MIP$ [log(n)-size messages]

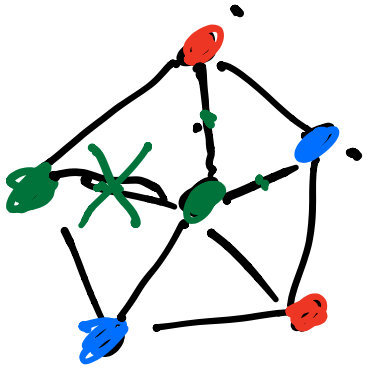
By the clause-var game, it would suffice to show that following prob. is NP-hard:

Given CSP instance $\mathcal{D}$, decide if

- YES: $\omega(\mathcal{D}) = 1$
- NO: $\omega(\mathcal{D}) \leq 1/2$

($\mathcal{D}$: $O(n)$ bits long)

Recall: 3-coloring is NP-complete



Is there a way to color vertices w/ 3 colors s.t. no monochromatic edges

YES: $\exists$ coloring s.t. all edges are good $\omega(\Phi) = 1$

NO: $\forall$ colorings, $\exists$ an edge which is bad

$\omega(\Phi) \leq 1 - \frac{1}{\#\text{edges}}$

Idea: find a way to "encode" a coloring s.t. checking a few constraints on encoding checks many edges of coloring

First: switch from 3-coloring to quadratic equations

$x_1, \ldots, x_n \in \{0,1\}$

$f_j: \underbrace{\sum a_i^{(j)} x_i}_{} + \underbrace{\sum b_{ke}^{(j)} x_k x_e}_{} = \underbrace{c^{(j)}}_{} \pmod 2$

$\vdots$

This is also NP-complete

We want encoding $\mathcal{E}$

$x_1, \ldots, x_n \xrightarrow{\mathcal{E}} y_1, \ldots, y_N$

s.t. I can check whether $x_1 \ldots x_n$ satisfy eqns. by reading a few y_i's

Def: The Hadamard code

$\{0,1\}^n \xrightarrow{\mathcal{E}_H} \{0,1\}^{2^n}$

$\underbrace{x_1 \ldots x_n}_{x}$ $\quad y_1 \ldots y_{2^n}$

$$y_a := \langle x, a \rangle = \sum_{i=1}^{n} x_i a_i \bmod 2$$

$a \in \{0,1\}^n$

x_n $\quad$ x_{n-1} $\quad$ $x_n + x_{n-1}$

$$y = \big(\langle x, 0 \rangle, \langle x, 0 \ldots 01 \rangle, \langle x, 0 \ldots 10 \rangle, \langle x, 0 \ldots 11 \rangle \ldots \big)$$

for grad. eq.

$$n \begin{cases} x_1 \\ \vdots \\ x_n \end{cases} \xrightarrow{\Sigma} \Big(\underbrace{\Sigma_H(x)}_{2^n}, \underbrace{\Sigma_H(x \otimes x)}_{2^{n^2}} \Big)$$

$$x \otimes x = (x_1 x_1, x_1 x_2, x_1 x_3, \ldots, x_2 x_1, \ldots) \in \{0,1\}^{n^2}$$

$$(*)\ \sum_i a_i^{(j)} x_i + \sum_{k\ell} b_{k\ell}^{(j)} x_k x_\ell = c^{(j)}$$

Given $\Sigma_H(x)$, $\Sigma_H(x \otimes x)$

Can I check $(*)$?

Obs: $\left(\Sigma_H(x)\right)_{a^{(j)}} = \sum_i a_i^{(j)} x_i$

$$\left(\Sigma_H(x \otimes x)\right)_{b^{(j)}} = \sum_{k\ell} b_{k\ell}^{(j)} x_k x_\ell$$

So you can check $(*)$ w/ just 2 queries to $\Sigma_H(x)$, $\Sigma_{H'}(x \otimes x)$

Recall:

Goal: $\omega(\text{QUADEQ}) = 1 \Rightarrow$ every check on $\Sigma(x)$ should pass

$\omega(\text{QUADEQ}) \leq 1 - 1/\#\text{eqs}. \Rightarrow$ at least $\Omega(1)$-fraction of

checks on $\xi(x)$ should fail

Fact: If $x \in \{0,1\}^n \neq 0$

then $\Pr_{u \in \{0,1\}^n}[\langle x, u \rangle = 0] = \frac{1}{2}$

$= 1 - \frac{1}{2}$

Suppose I fix an assignment $x_1, \ldots, x_n$

$$f^{(1)} = \sum a^{(1)} x + \sum b^{(1)}_{xx} = \underline{\underline{d}}^{(1)} \stackrel{?}{=} c^{(1)}$$

$$\vdots$$

$$f^{(m)} = \sum a^{(m)} x + \sum b^{(m)}_{xx} = \underline{\underline{d}}^{(m)} \stackrel{?}{=} c^{(m)}$$

Check that $\vec{d} - \vec{c} = 0$

Fact $\Rightarrow$ - If $\vec{d} - \vec{c} = 0$, then

$$\Pr_{\vec{u}}[\langle \vec{u}, \vec{d} - \vec{c} \rangle = 0] = 1$$

- If $\vec{d}-\vec{c} \neq 0$, then

$$P_{\vec{u}}\left[\langle \vec{u}, \vec{d}-\vec{c}\rangle = 0\right] = 1/2$$

Translating this idea back to eqns, Consider

(†) $\langle \vec{u}, \vec{F}\rangle = \sum_{j=1}^{m} u_j \left(\sum a^{(j)} x + \sum b^{(j)} x x\right)$

$$\stackrel{?}{=} \sum_{j=1}^{m} u_j c^{(j)}$$

By above, if $f_1 \ldots f_m$ is satisfied, th (†) is satisfied w/ prob 1

if $f_1 \ldots f_m$ not satisfied, then (†) is satisfied w/ prob $1/2$

Given (†), $\Sigma_H(x)$, $\Sigma_H(x \otimes x)$, can check (†) in 2 queries

$$(\dagger): \sum_i \left(\sum_{j=1}^m u_j a_i^{(j)}\right) x_i + \sum_{k\ell} \left(\sum_{j=1}^m u_j b_{k\ell}^{(j)}\right) x_k x_\ell \doteq \sum_j u_j c^{(j)}$$

$(\Sigma_H(x)) \sum_{j=1}^m u_j a^{(j)}$

$(\Sigma_H(x \otimes x)) \sum_{j=1}^m u_j b^{(j)}$

Putting it all together, we showed that following prob. is NP-hard

Given an instance of QUADEQ

- YES: $\exists x, \Sigma(x)$ that satisfies all constraints

- NO: $\forall x$,

$$\Pr_{\vec{u}}\left[\tilde{\Sigma}_H(x) \Sigma_j u_j a^{(j)} + \tilde{\Sigma}_H(x \otimes x) \Sigma_j u_j b^{(j)} = \Sigma_j u_j c^{(j)}\right] \leq 1/2$$

Soundness only holds against valid encodings

To fix this, need a way to test that a given Π is an encoding of some x.