Last time:
- Interactive Proof Systems
- The class IP
- Graph isomorphism problem, $ISO \in IP$
- #$SAT \in IP$ (part 1)

Today:
- Arithmetization of Boolean formulas
- Finish #$SAT \in IP$ and conclude that coNP $\subseteq IP$

Posted: Solutions to Problem Set 6
Review: Interactive Proofs

Two interacting parties
Verifier (V): Probabilistic polynomial time TM
Prover (P): Unlimited computational power

Both P and V see input \( w \).
They exchange a polynomial number of polynomial-size messages.
Then V accepts or rejects.

Defn: \( \Pr[(V \leftrightarrow P) \text{ accepts } w] \) = probability that V accepts when V interacts with P, given input \( w \).

Defn: \( \text{IP} = \{A\} \) for some V and P (This P is an “honest” prover)
\[
\begin{align*}
    w \in A & \rightarrow \Pr[(V \leftrightarrow P) \text{ accepts } w] \geq \frac{2}{3} \\
    w \notin A & \rightarrow \forall \tilde{P} \Pr[(V \leftrightarrow \tilde{P}) \text{ accepts } w] \leq \frac{1}{3}
\end{align*}
\]
Think of \( \tilde{P} \) as a “crooked” prover trying to make V accept when it shouldn’t.

Equivalently: \( \text{IP} = \{A\} \) for some V
\[
\begin{align*}
    w \in A & \rightarrow \exists P \Pr[(V \leftrightarrow P) \text{ accepts } w] \geq \frac{2}{3} \\
    w \notin A & \rightarrow \forall P \Pr[(V \leftrightarrow P) \text{ accepts } w] \geq \frac{1}{3}
\end{align*}
\]
Here, we emphasize how P is similar to the certificate for NP-languages.

An amplification lemma can improve the error probability from \( \frac{1}{3} \) to \( \frac{1}{2\text{poly}(n)} \).
Surprising Theorem: $\text{IP} = \text{PSPACE}$

- $\text{IP} \subseteq \text{PSPACE}$: standard simulation, similar to $\text{NP} \subseteq \text{PSPACE}$
- $\text{PSPACE} \subseteq \text{IP}$: show $TQBF \in \text{IP}$, we won’t prove
- $\text{coNP} \subseteq \text{IP}$: weaker but similar, show $\#\text{SAT} \in \text{IP}$ ($\#\text{SAT}$ is coNP-hard)

$\#\text{SAT} = \{ (\phi, k) \mid \text{Boolean formula } \phi \text{ has exactly } k \text{ satisfying assignments} \}$

**Theorem:** $\#\text{SAT} \in \text{IP}$

**Proof:** First some notation. Assume $\phi$ has $m$ variables $x_1, \ldots, x_m$.

Let $\phi(0)$ be $\phi$ with $x_1 = 0$ (0 substituted for $x_1$) $0 = \text{FALSE}$ and $1 = \text{TRUE}$. Let $\phi(a_1 \ldots a_i)$ be $\phi$ with $x_1 = a_1, \ldots, x_i = a_i$ for $a_1, \ldots, a_i \in \{0, 1\}$.

Call $a_1, \ldots, a_i$ *presets*. The remaining $x_{i+1}, \ldots, x_m$ stay as unset variables.

Let $\#\phi$ be the number of satisfying assignments of $\phi$.
Let $\#\phi(0)$ be the number of satisfying assignments of $\phi(0)$.
Let $\#\phi(a_1 \ldots a_i) = \text{the number of satisfying assignments of } \phi(a_1 \ldots a_i)$

---

**Check-in 26.1**

Let $\phi = (x_1 \lor x_2) \land (\overline{x_1} \lor \overline{x_2})$

Check all that are true:

a) $\#\phi = 1$  

b) $\#\phi = 2$

c) $\#\phi(0) = 1$  

d) $\#\phi(0) = 2$

e) $\#\phi(00) = 0$  

f) $\#\phi(00) = 1$
**Theorem:** \( \#SAT \in IP \)

**Proof:** Protocol for V and (the honest) P on input \( \langle \phi, k \rangle \)

0) P sends \( \#\phi \); V checks \( k = \#\phi \)

1) P sends \( \#\phi(0), \#\phi(1) \); V checks \( \#\phi = \#\phi(0) + \#\phi(1) \)

2) P sends \( \#\phi(00), \#\phi(01), \#\phi(10), \#\phi(11) \); V checks \( \#\phi(00) = \#\phi(00) + \#\phi(01) \)

\[ \#\phi(01) = \#\phi(10) + \#\phi(11) \]

\[ \vdots \]

\[ m \) P sends \( \#\phi(0 \cdots 0), \ldots, \#\phi(1 \cdots 1) \); V checks \( \#\phi(0 \cdots 0) = \#\phi(0 \cdots 00) + \#\phi(0 \cdots 01) \)

\[ \vdots \]

V checks \( \#\phi(1 \cdots 1) = \#\phi(1 \cdots 10) + \#\phi(1 \cdots 11) \)

\[ \vdots \]

\[ m + 1 \) V checks \( \#\phi(0 \cdots 0) = \phi(0 \cdots 0) \)

\[ \vdots \]

\[ \#\phi(1 \cdots 1) = \phi(1 \cdots 1) \]

V accepts if all checks are correct. Otherwise V rejects.

**Problem:** Exponential. Will fix.
Idea for fixing \( \#SAT \in IP \) protocol

Non-Boolean assignments to the variables of \( \phi \)
Arithmetizing Boolean formulas

Simulate $\land$ and $\lor$ with $+$ and $\times$

\[
\begin{align*}
a \land b & \rightarrow a \times b = ab \\
\overline{a} & \rightarrow (1 - a) \\
a \lor b & \rightarrow a + b - ab \\
\phi & \rightarrow p_\phi \quad \text{degree}(p_\phi) \leq |\phi|
\end{align*}
\]

Let $\mathbb{F}_q = \{0,1,\ldots,q-1\}$ for prime $q > 2^m$ be a finite field $(+, \times \mod q)$ and let $a_1,\ldots,a_i \in \mathbb{F}_q$

Let $\phi(a_1 \ldots a_i) = p_\phi$ where $x_1 \ldots x_i = a_1 \ldots a_i$ and remaining $x_{i+1},\ldots,x_m$ stay as unset variables.

Let $\#\phi(a_1 \ldots a_i) = \sum_{a_{i+1},\ldots,a_m \in \{0,1\}} \phi(a_1 \ldots a_m)$

Check-in 26.2

Let $\phi = (x_1 \lor x_2) \land (\overline{x_1} \lor \overline{x_2})$. Check all that are true:

a) $p_\phi = (x_1 + x_2 - x_1x_2)( (1 - x_1) + (1 - x_2) - (1 - x_1)(1 - x_2) )$

b) $p_\phi = (x_1 + x_2)( (1 - x_1) + (1 - x_2) )$

c) $p_\phi = (x_1 + x_2 - 2x_1x_2)$

identities still true

1. $\#\phi(a_1 \ldots a_i) = \#\phi(a_1 \ldots a_i0) + \#\phi(a_1 \ldots a_i1)$

2. $\#\phi(a_1 \ldots a_m) = \phi(a_1 \ldots a_m)$
Theorem: \( \#SAT \in IP \)

Proof: Protocol for V and (the honest) P on input \( \langle \phi, k \rangle \)

0) P sends \( \#\phi \); V checks \( k = \#\phi \)

1) P sends \( \#\phi(0), \#\phi(1) \); V checks \( \#\phi = \#\phi(0) + \#\phi(1) \) [by evaluating polynomial for \( \#\phi(z) \)]

V sends random \( r_1 \in \mathbb{F}_q \)

[\( P \) needs to show \( \#\phi(z) \) is correct]

2) P sends \( \#\phi(r_1z) \) as a polynomial in \( z \)

V checks \( \#\phi(r_1) = \#\phi(r_10) + \#\phi(r_11) \) [by evaluating polynomial for \( \#\phi(r_1z) \)]

V sends random \( r_2 \in \mathbb{F}_q \)

\( \vdots \)

m) P sends \( \#\phi(r_1 \cdots r_{m-1}z) \) as a polynomial in \( z \)

V checks \( \#\phi(r_1 \cdots r_{m-1}) = \#\phi(r_1 \cdots r_{m-1}0) + \#\phi(r_1 \cdots r_{m-1}1) \)

V sends random \( r_m \in \mathbb{F}_q \)

m + 1) V checks \( \#\phi(r_1 \cdots r_m) = \phi(r_1 \cdots r_m) \)

V accepts if all checks are correct. Otherwise V rejects.
#SAT ∈ IP – version 2

Input \( \langle \phi, k \rangle \)

**Prover sends**

\[ \phi \]

\[ \phi(z) = 3z^d - 5z^{d-1} + \cdots + 7 \]

\[ \phi(r_1 z) = \cdots \]

\[ \phi(r_1 r_2 z) = \cdots \]

\[ \phi(r_1 \cdots r_{m-1} z) = \cdots \]

**Verifier checks**

If \( k \) is correct, \( V \) will accept.

If \( k \) is wrong, \( V \) probably will reject, whatever \( P \) does.

**Verifier sends**

\[ \phi = k \]

\[ \phi(0) \quad \phi(1) \]

\[ \phi(r_1) \]

\[ \phi(r_1 0) \quad \phi(r_1 1) \]

\[ \phi(r_1 r_2) \]

\[ \phi(r_1 r_2 0) \quad \phi(r_1 r_2 1) \]

\[ \vdots \]

\[ \phi(r_1 \cdots r_{m-1}) \]

\[ \phi(r_1 \cdots r_{m-1} 0) \quad \phi(r_1 \cdots r_{m-1} 1) \]

\[ \phi(r_1 \cdots r_{m-1}) \]

\[ \phi(r_1 \cdots r_{m-1}) \]

\[ \phi(r_1 \cdots r_m) \]

\[ \phi(r_1 \cdots r_m) \]
Check-in 26.3

**P = NP?**

a) YES. Deep learning will do $SAT \in P$, but we won’t understand how.

b) NO. But we will never prove it.

c) NO. We will prove it but only after 100 years

d) NO. We will prove it in $n$ years, $20 \leq n \leq 100$

e) NO. We will prove it in $n$ years, $1 \leq n < 20$

f) NO. One of us is writing up the proof now...
Quick review of today

Finished $\#SAT \in IP$ and $\text{coNP} \subseteq IP$

**Additional subjects:**
18.405/6.841  Advanced complexity F2021
18.425/6.875  Cryptography F2021
6.842  Randomness and Computation

*Good luck on the final!*

*Best wishes for the holidays and the New Year!*