18.404/6.840 Lecture 26

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

Today: (Sipser §10.4)
- Arithmetization of Boolean formulas
- Finish $\#SAT \in$ IP and conclude that coNP $\subseteq$ IP

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 ↔ P) accepts w] = probability that V accepts when V interacts with P, given input w.

Defn: IP $= \{A|$ for some V and P (This P is an "honest" prover)

$w \in A \rightarrow$ Pr [(V ↔ P) accepts w] $\geq {}^{2}/_{3}$

$w \notin A \rightarrow$ for any prover $\tilde{\mathrm{P}}$ Pr [(V ↔ $\tilde{\mathrm{P}}$) accepts w] $\leq {}^{1}/_{3}$ }

Think of $\tilde{\mathrm{P}}$ as a "crooked" prover trying to make V accept when it shouldn't.

Equivalently: IP $= \{A|$ for some V

$w \in A \rightarrow \exists$P Pr [(V ↔ P) accepts w] $\geq {}^{2}/_{3}$

$w \notin A \rightarrow \nexists$P Pr [(V ↔ P) accepts w] $\geq {}^{1}/_{3}$ }

Here, we emphasize how P is similar to the certificate for NP-languages.

An amplification lemma can improve the error probability from ${}^{1}/_{3}$ to ${}^{1}/_{2^{\mathrm{poly}\,(n)}}$

coNP ⊆ IP

Surprising Theorem: IP = PSPACE

IP ⊆ PSPACE: standard simulation, similar to NP ⊆ PSPACE
PSPACE ⊆ IP: show $TQBF$ ∈ IP, we won't prove
coNP ⊆ IP: weaker but similar, show $\#SAT$ ∈ IP ($\#SAT$ is coNP-hard)

$\#SAT = \{\langle \phi, k \rangle |$ Boolean formula ϕ has exactly k satisfying assignments$\}$
Theorem: $\#SAT$ ∈ IP
Proof: First some notation. Assume ϕ has m variables $x_1, \dots, x_m$.

Let $\phi(0)$ be ϕ with $x_1 = 0$ (0 substituted for x_1) 0 = FALSE and 1 = TRUE.
Let $\phi(a_1 \dots a_i)$ be ϕ with $x_1 = a_1, \dots, x_i = a_i$ for $a_1, \dots, a_i \in \{0,1\}$.
Call $a_1, \dots, a_i$ presets. The remaining $x_{i+1}, \dots, x_m$ stay as unset variables.

Let $\#\phi$ = the number of satisfying assignments of ϕ.
Let $\#\phi(0)$ = the number of satisfying assignments of $\phi(0)$.
Let $\#\phi(a_1 \dots a_i)$ = the number of satisfying assignments of $\phi(a_1 \dots a_i)$

Check-in 26.1
Let $\phi = (x_1 \vee x_2) \wedge (x_1 \vee \overline{x_2})$
Check all that are true:

a) $\#\phi = 1$
b) $\#\phi = 2$
c) $\#\phi\ \ 0\ \ = 1$
d) $\#\phi(0) = 2$
e) $\#\phi(00) = 0$
f) $\#\phi(00) = 1$

Check-in 26.1

$\#SAT \in$ IP – 1^{st} attempt

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 $\#\phi(0) = \#\phi(00) + \#\phi(01)$

$\#\phi(1) = \#\phi(10) + \#\phi(11)$

⋮

m) P sends $\#\phi(\overbrace{0 \cdots 0}^{m}), \ldots, \#\phi(1 \cdots 1)$; V checks $\#\phi(\overbrace{0 \cdots 0}^{m-1}) = \#\phi(\overbrace{0 \cdots 00}^{m-1}) + \#\phi(0 \cdots 01)$

⋮

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

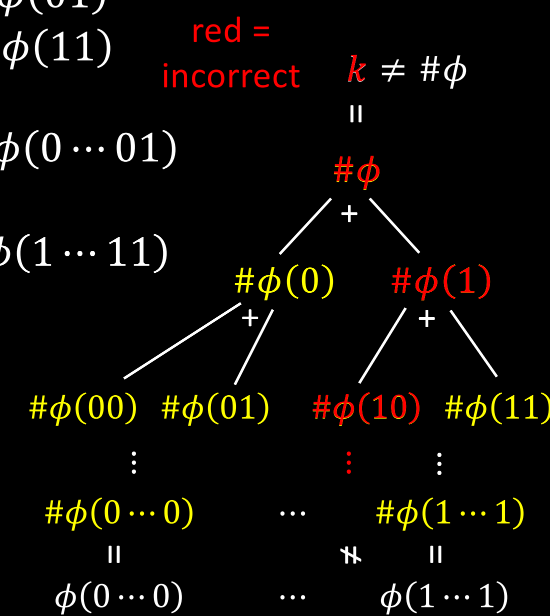
$m + 1$) V checks $\#\phi(\overbrace{0 \cdots 0}^{m}) = \phi(0 \cdots 0)$

⋮

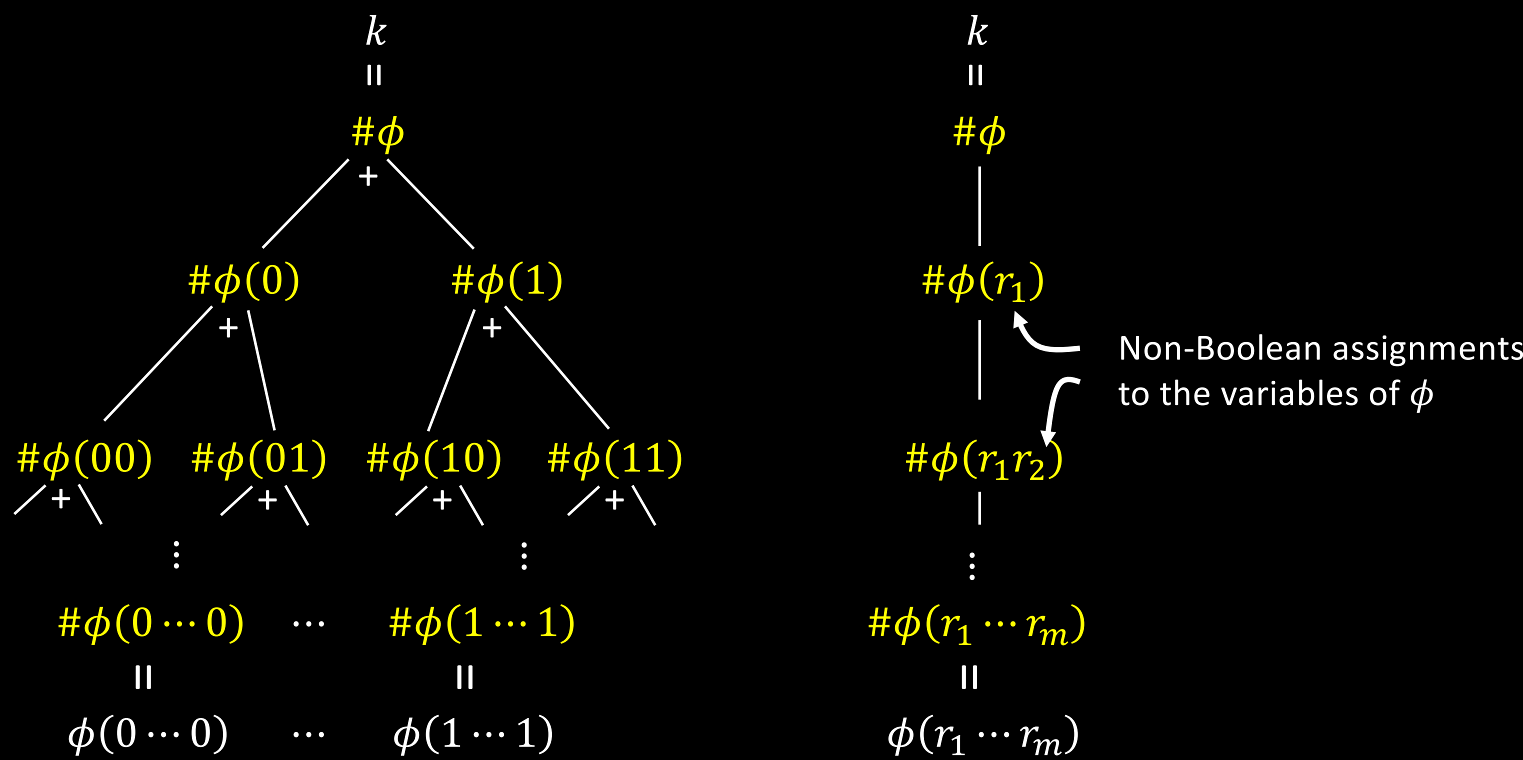
$\#\phi(1 \cdots 1) = \phi(1 \cdots 1)$

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

Problem: Exponential. Will fix.



Idea for fixing $\#SAT \in$ IP protocol



Arithmetizing Boolean formulas

Simulate $\wedge$ and $\vee$ with $+$ and $\times$

$$
\begin{array}{ccl}
a \wedge b & \rightarrow & a \times b = ab \\
\overline{a} & \rightarrow & (1-a) \\
a \vee b & \rightarrow & a + b - ab \\
\phi & \rightarrow & p_\phi \quad \text{degree}(p_\phi) \leq |\phi|
\end{array}
$$

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

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

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

identities still true

1. $\#\phi(a_1 \dots a_i) = \#\phi(a_1 \dots a_i 0) + \#\phi(a_1 \dots a_i 1)$
2. $\#\phi(a_1 \dots a_m) = \phi(a_1 \dots a_m)$

Check-in 26.2

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

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

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

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

Check-in 26.2

$\#SAT \in$ IP – version 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 $\#\phi = \#\phi(0) + \#\phi(1)$

1) P sends $\#\phi(z)$ as a polynomial in z [sends coefficients – recall deg $p_\phi \leq |\phi|$]

V checks $\#\phi = \#\phi(0) + \#\phi(1)$ [by evaluating polynomial for $\#\phi(z)$]

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

V sends random $r_1 \in \mathbb{F}_q$

2) P sends $\#\phi(r_1 z)$ as a polynomial in z

V checks $\#\phi(r_1) = \#\phi(r_1 0) + \#\phi(r_1 1)$ [by evaluating polynomial for $\#\phi(r_1 z)$]

V sends random $r_2 \in \mathbb{F}_q$

⋮

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

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*.

$\#SAT \in$ 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 sends

r_1

r_2

r_m

reject

Verifier checks

$\#\phi = k$

$+$

$\#\phi(0) \quad \#\phi(1)$

$\#\phi(r_1)$

$+$

$\#\phi(r_1 0) \quad \#\phi(r_1 1)$

$\#\phi(r_1 r_2)$

$+$

$\#\phi(r_1 r_2 0) \quad \#\phi(r_1 r_2 1)$

$\vdots$

$\#\phi(r_1 \cdots r_{m-1})$

$+$

$\#\phi(r_1 \cdots r_{m-1} 0) \quad \#\phi(r_1 \cdots r_{m-1} 1)$

$\#\phi(r_1 \cdots r_m)$

$\not\parallel$

$\phi(r_1 \cdots r_m)$

If k is correct, V will accept.

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

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…

Check-in 26.3

Quick review of today

Finished $\#SAT \in$ IP and 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!