بسم الله الرحمن الرحیم

نظریه علوم کامپیوتر

نظریه علوم کامپیوتر - بهار ۱۴۰۰-۱۴۰۱ - جلسه نوزدهم: سیستم اثبات تعاملی (۲)

Theory of computation - 002 - S19 - IP (2)

Legend

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*.

Review: Interactive Proofs

Defn: Pr[(V $\leftrightarrow$ P) accepts w] =

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*.

Review: Interactive Proofs

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

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

$$w \in A \rightarrow \text{Pr [(V} \leftrightarrow \text{P) accepts } w \text{]} \geq \frac{2}{3}$$

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*.

Review: Interactive Proofs

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

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

$$w \in A \rightarrow \text{Pr [(V} \leftrightarrow \text{P) accepts } w \text{]} \geq \frac{2}{3}$$

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*.

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

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

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

$$w \in A \rightarrow \quad \Pr[\,(\mathrm{V} \leftrightarrow \mathrm{P})\text{ accepts } w\,] \geq \frac{2}{3}$$

$$w \notin A \rightarrow \quad \text{for any prover } \widetilde{\mathrm{P}} \quad \Pr[\,(\mathrm{V} \leftrightarrow \widetilde{\mathrm{P}})\text{ accepts } w\,] \leq \frac{1}{3}\}$$

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

$$w \in A \rightarrow \quad \exists \mathrm{P} \quad \Pr[\,(\mathrm{V} \leftrightarrow \mathrm{P})\text{ accepts } w\,] \geq \frac{2}{3}$$

$$w \notin A \rightarrow \quad \nexists \mathrm{P} \quad \Pr[\,(\mathrm{V} \leftrightarrow \mathrm{P})\text{ accepts } w\,] \geq \frac{1}{3}\}$$

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

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

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

$w \in A \;\rightarrow\;$ Pr [(V $\leftrightarrow$ P) accepts w] $\geq \frac{2}{3}$

$w \notin A \;\rightarrow\;$ for any prover $\tilde{\text{P}}$ Pr [(V $\leftrightarrow \tilde{\text{P}}$) accepts w] $\leq \frac{1}{3} \}$

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

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

$w \in A \;\rightarrow\; \exists$P Pr [(V $\leftrightarrow$ P) accepts w] $\geq \frac{2}{3}$

$w \notin A \;\rightarrow\; \nexists$P Pr [(V $\leftrightarrow$ P) accepts w] $\geq \frac{1}{3} \}$

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

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

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

$$w \in A \;\rightarrow\; \Pr[\,(V \leftrightarrow P) \text{ accepts } w\,] \geq \frac{2}{3}$$

$$w \notin A \;\rightarrow\; \text{for any prover } \widetilde{P} \;\; \Pr[\,(V \leftrightarrow \widetilde{P}) \text{ accepts } w\,] \leq \frac{1}{3}\}$$

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

$$w \in A \;\rightarrow\; \exists P \;\; \Pr[\,(V \leftrightarrow P) \text{ accepts } w\,] \geq \frac{2}{3}$$

$$w \notin A \;\rightarrow\; \nexists P \;\; \Pr[\,(V \leftrightarrow P) \text{ accepts } w\,] \geq \frac{1}{3}\}$$

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

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 $\frac{1}{2^{\text{poly}(n)}}$

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

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

$$w \in A \rightarrow \text{Pr [(V} \leftrightarrow \text{P) accepts } w\text{]} \geq \frac{2}{3}$$

$$w \notin A \rightarrow \text{for any prover } \widetilde{\text{P}} \quad \text{Pr [(V} \leftrightarrow \widetilde{\text{P}}\text{) accepts } w\text{]} \leq \frac{1}{3} \}$$

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

$$w \in A \rightarrow \exists \text{P} \quad \text{Pr [(V} \leftrightarrow \text{P) accepts } w\text{]} \geq \frac{2}{3}$$

$$w \notin A \rightarrow \nexists \text{P} \quad \text{Pr [(V} \leftrightarrow \text{P) accepts } w\text{]} \geq \frac{1}{3} \}$$

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

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 $\frac{1}{2^{\text{poly}(n)}}$

coNP $\subseteq$ IP

coNP $\subseteq$ IP

Surprising Theorem: IP = PSPACE

coNP $\subseteq$ IP

Surprising Theorem: IP = PSPACE

IP $\subseteq$ PSPACE: standard simulation, similar to NP $\subseteq$ PSPACE

coNP $\subseteq$ IP

Surprising Theorem: IP = PSPACE

IP $\subseteq$ PSPACE: standard simulation, similar to NP $\subseteq$ PSPACE

PSPACE $\subseteq$ IP: show $TQBF \in$ IP, we won't prove

coNP $\subseteq$ IP

Surprising Theorem: IP = PSPACE

IP $\subseteq$ PSPACE: standard simulation, similar to NP $\subseteq$ PSPACE

PSPACE $\subseteq$ IP: show $TQBF \in$ IP, we won't prove

coNP $\subseteq$ IP: weaker but similar, show $\#SAT \in$ IP ($\#SAT$ is coNP-hard)

coNP $\subseteq$ IP

Surprising Theorem: IP = PSPACE

IP $\subseteq$ PSPACE: standard simulation, similar to NP $\subseteq$ PSPACE

PSPACE $\subseteq$ IP: show $TQBF \in$ IP, we won't prove

coNP $\subseteq$ IP: weaker but similar, show $\#SAT \in$ IP ($\#SAT$ is coNP-hard)

$\#SAT = \left\{ \langle \phi, k \rangle \,\middle|\, \text{Boolean formula } \phi \text{ has exactly } k \text{ satisfying assignments} \right\}$

Theorem: $\#SAT \in$ IP

coNP $\subseteq$ IP

Surprising Theorem: IP = PSPACE

IP $\subseteq$ PSPACE: standard simulation, similar to NP $\subseteq$ PSPACE

PSPACE $\subseteq$ IP: show $TQBF \in$ IP, we won't prove

coNP $\subseteq$ IP: weaker but similar, show $\#SAT \in$ IP ($\#SAT$ is coNP-hard)

$\#SAT = \{\langle \phi, k \rangle \mid$ Boolean formula ϕ has exactly k satisfying assignments$\}$

Theorem: $\#SAT \in$ IP

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

coNP ⊆ IP

Surprising Theorem: IP = PSPACE

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

$\#SAT = \{\langle \phi, k \rangle \mid$ Boolean formula ϕ has exactly k satisfying assignments$\}$

Theorem: $\#SAT \in$ IP

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

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

coNP ⊆ IP

Surprising Theorem: IP = PSPACE

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

$\#SAT = \{\langle \phi, k \rangle \mid$ Boolean formula ϕ has exactly k satisfying assignments$\}$

Theorem: $\#SAT \in$ IP

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

Let $\phi(0)$ be ϕ with $x_1 = 0$ (0 substituted for x_1) 0 = FALSE and 1 = TRUE.
Let $\phi(a_1 \ldots a_i)$ be ϕ with $x_1 = a_1$, … , $x_i = a_i$ 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 $\#\phi$ = the number of satisfying assignments of ϕ.
Let $\#\phi(0)$ = the number of satisfying assignments of $\phi(0)$.
Let $\#\phi(a_1 \ldots a_i)$ = the number of satisfying assignments of $\phi(a_1 \ldots a_i)$

coNP ⊆ IP

Surprising Theorem: IP = PSPACE

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

$\#SAT = \{\langle \phi, k\rangle \mid$ Boolean formula ϕ has exactly k satisfying assignments$\}$

Theorem: $\#SAT \in$ IP

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

Let $\phi(0)$ be ϕ with $x_1 = 0$ (0 substituted for x_1) 0 = FALSE and 1 = TRUE.

Let $\phi(a_1 \ldots a_i)$ be ϕ with $x_1 = a_1$, … , $x_i = a_i$ 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.

Two identities

Let $\#\phi$ = the number of satisfying assignments of ϕ.

Let $\#\phi(0)$ = the number of satisfying assignments of $\phi(0)$.

Let $\#\phi(a_1 \ldots a_i)$ = the number of satisfying assignments of $\phi(a_1 \ldots a_i)$

coNP ⊆ IP

Surprising Theorem: IP = PSPACE

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

$\#SAT = \{\langle \phi, k \rangle \mid$ Boolean formula ϕ has exactly k satisfying assignments$\}$

Theorem: $\#SAT \in$ IP

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

Let $\phi(0)$ be ϕ with $x_1 = 0$ (0 substituted for x_1) 0 = FALSE and 1 = TRUE.
Let $\phi(a_1 \ldots a_i)$ be ϕ with $x_1 = a_1$, … , $x_i = a_i$ 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 $\#\phi$ = the number of satisfying assignments of ϕ.
Let $\#\phi(0)$ = the number of satisfying assignments of $\phi(0)$.
Let $\#\phi(a_1 \ldots a_i)$ = the number of satisfying assignments of $\phi(a_1 \ldots a_i)$

Two identities

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

coNP ⊆ IP

Surprising Theorem: IP = PSPACE

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

$\#SAT = \{\langle \phi, k \rangle \mid$ Boolean formula ϕ has exactly k satisfying assignments$\}$

Theorem: $\#SAT \in$ IP

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

Let $\phi(0)$ be ϕ with $x_1 = 0$ (0 substituted for x_1) 0 = FALSE and 1 = TRUE.
Let $\phi(a_1 \ldots a_i)$ be ϕ with $x_1 = a_1$, … , $x_i = a_i$ 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 $\#\phi$ = the number of satisfying assignments of ϕ.
Let $\#\phi(0)$ = the number of satisfying assignments of $\phi(0)$.
Let $\#\phi(a_1 \ldots a_i)$ = the number of satisfying assignments of $\phi(a_1 \ldots a_i)$

Two identities

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

Check-in 26.1

coNP ⊆ IP

Surprising Theorem: IP = PSPACE

IP ⊆ PSPACE: standard simulation, similar to NP ⊆ PSPACE

PSPACE ⊆ IP: show $TQBF \in$ IP, we won't prove

coNP ⊆ IP: weaker but similar, show $\#SAT \in$ IP ($\#SAT$ is co

$\#SAT = \{\langle \phi, k \rangle \mid$ Boolean formula ϕ has exactly k satisfying assig

Theorem: $\#SAT \in$ IP

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

Let $\phi(0)$ be ϕ with $x_1 = 0$ (0 substituted for x_1) 0 = FALSE and 1 = TRUE.

Let $\phi(a_1 \ldots a_i)$ be ϕ with $x_1 = a_1$, … , $x_i = a_i$ 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 variable

Let $\#\phi$ = the number of satisfying assignments of ϕ.

Let $\#\phi(0)$ = the number of satisfying assignments of $\phi(0)$.

Let $\#\phi(a_1 \ldots a_i)$ = the number of satisfying assignments of $\phi(a_1 \ldots a_i)$

Check-in 26.1

Let $\phi = (x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2})$

Check all that are true:

a) $\#\phi = 1$ b) $\#\phi = 2$

c) $\#\phi(0) = 1$ d) $\#\phi(0) = 2$

e) $\#\phi(00) = 0$ f) $\#\phi(00) = 1$

Two identities

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

Check-in 26.1

$\#SAT \in$ IP – 1st attempt

Theorem: $\#SAT \in$ IP

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

$\#SAT \in$ IP – 1st 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 $k = \#\phi$

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

$\#SAT \in$ IP – 1st 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 $\#\phi(0) = \#\phi(00) + \#\phi(01)$
   $\#\phi(1) = \#\phi(10) + \#\phi(11)$

$\#SAT \in$ IP – 1st 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 $\#\phi(0) = \#\phi(00) + \#\phi(01)$

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

$\vdots$

m) P sends $\#\phi(\overbrace{0\cdots0}^{m}), \ldots, \#\phi(1\cdots1)$; V checks $\#\phi(0\cdots0) = \#\phi(\overbrace{0\cdots0}^{m-1}0) + \#\phi(\overbrace{0\cdots0}^{m-1}1)$

$\vdots$

V checks $\#\phi(1\cdots1) = \#\phi(1\cdots10) + \#\phi(1\cdots11)$

$\#SAT \in$ IP – 1st 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 $\#\phi(0) = \#\phi(00) + \#\phi(01)$

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

$\vdots$

m) P sends $\#\phi(\overbrace{0\cdots0}^{m}), \ldots, \#\phi(1\cdots1)$; V checks $\#\phi(0\cdots0) = \#\phi(\overbrace{0\cdots00}^{m-1}) + \#\phi(\overbrace{0\cdots01}^{m-1})$

$\vdots$

V checks $\#\phi(1\cdots1) = \#\phi(1\cdots10) + \#\phi(1\cdots11)$

$m+1$) V checks $\#\phi(\overbrace{0\cdots0}^{m}) = \phi(0\cdots0)$

$\vdots$

$$\#\phi(1\cdots1) = \phi(1\cdots1)$$

$\#SAT \in$ IP – 1st 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 $\#\phi(0) = \#\phi(00) + \#\phi(01)$

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

$\vdots$

m) P sends $\#\phi(\overbrace{0\cdots0}^{m}), \ldots, \#\phi(1\cdots1)$; V checks $\#\phi(0\cdots0) = \#\phi(\overbrace{0\cdots00}^{m-1}) + \#\phi(\overbrace{0\cdots01}^{m-1})$

$\vdots$

V checks $\#\phi(1\cdots1) = \#\phi(1\cdots10) + \#\phi(1\cdots11)$

$m+1$) V checks $\#\phi(\overbrace{0\cdots0}^{m}) = \phi(0\cdots0)$

$\vdots$

$$\#\phi(1\cdots1) = \phi(1\cdots1)$$

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

$\#SAT \in$ IP – 1st 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 $\#\phi(0) = \#\phi(00) + \#\phi(01)$

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

k

$=$

$\#\phi$

$\vdots$

m) P sends $\#\phi(\overbrace{0\cdots0}^{m}), \ldots, \#\phi(1\cdots1)$; V checks $\#\phi(0\cdots0) = \#\phi(\overbrace{0\cdots0}^{m-1}0) + \#\phi(\overbrace{0\cdots0}^{m-1}1)$

$\vdots$

V checks $\#\phi(1\cdots1) = \#\phi(1\cdots10) + \#\phi(1\cdots11)$

$m+1$) V checks $\#\phi(\overbrace{0\cdots0}^{m}) = \phi(0\cdots0)$

$\vdots$

$\#\phi(1\cdots1) = \phi(1\cdots1)$

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

$\#SAT \in$ IP – 1st 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 $\#\phi(0) = \#\phi(00) + \#\phi(01)$

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

$\vdots$

m) P sends $\#\phi(\overbrace{0\cdots0}^{m}), \ldots, \#\phi(1\cdots1)$; V checks $\#\phi(0\cdots0) = \#\phi(\overbrace{0\cdots0}^{m-1}0) + \#\phi(\overbrace{0\cdots0}^{m-1}1)$

$\vdots$

V checks $\#\phi(1\cdots1) = \#\phi(1\cdots10) + \#\phi(1\cdots11)$

$m+1$) V checks $\#\phi(\overbrace{0\cdots0}^{m}) = \phi(0\cdots0)$

$\vdots$

$$\#\phi(1\cdots1) = \phi(1\cdots1)$$

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

k

$\shortparallel$

$\#\phi$

$+$

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

$\#SAT \in$ IP – 1st 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 $\#\phi(0) = \#\phi(00) + \#\phi(01)$

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

$\vdots$

m) P sends $\#\phi(\overbrace{0\cdots0}^{m}), \ldots, \#\phi(1\cdots1)$; V checks $\#\phi(0\cdots0) = \#\phi(\overbrace{0\cdots0}^{m-1}0) + \#\phi(\overbrace{0\cdots0}^{m-1}1)$

$\vdots$

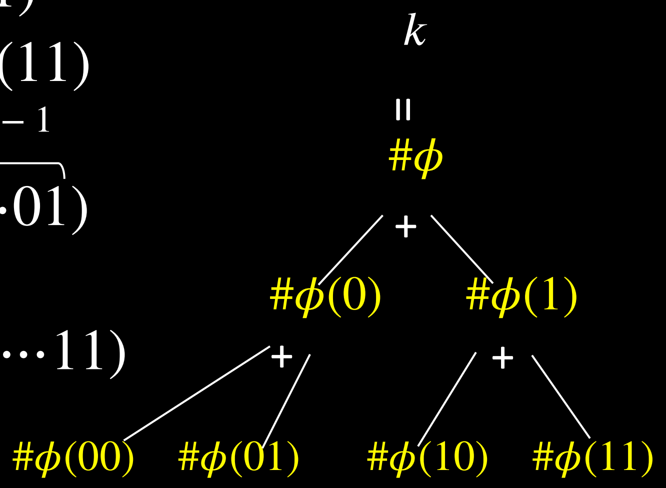
V checks $\#\phi(1\cdots1) = \#\phi(1\cdots10) + \#\phi(1\cdots11)$

$m+1$) V checks $\#\phi(\overbrace{0\cdots0}^{m}) = \phi(0\cdots0)$

$\vdots$

$\#\phi(1\cdots1) = \phi(1\cdots1)$

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



$\#SAT \in$ IP – 1st 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 $\#\phi(0) = \#\phi(00) + \#\phi(01)$

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

$\vdots$

m) P sends $\#\phi(\overbrace{0\cdots0}^{m}), \ldots, \#\phi(1\cdots1)$; V checks $\#\phi(0\cdots0) = \#\phi(\overbrace{0\cdots00}^{m-1}) + \#\phi(\overbrace{0\cdots01}^{m-1})$

$\vdots$

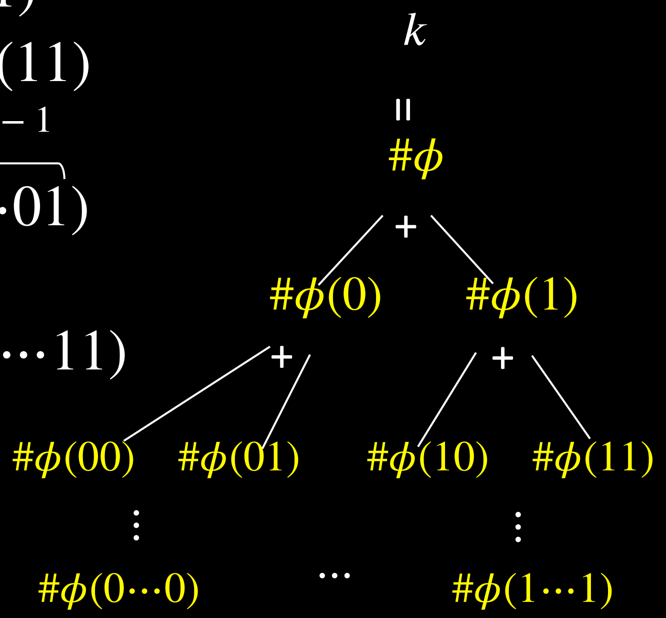
V checks $\#\phi(1\cdots1) = \#\phi(1\cdots10) + \#\phi(1\cdots11)$

$m+1$) V checks $\#\phi(\overbrace{0\cdots0}^{m}) = \phi(0\cdots0)$

$\vdots$

$\#\phi(1\cdots1) = \phi(1\cdots1)$

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



$\#SAT \in$ IP – 1st 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 $\#\phi(0) = \#\phi(00) + \#\phi(01)$

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

$\vdots$

m) P sends $\#\phi(\overbrace{0\cdots0}^{m}), \ldots, \#\phi(1\cdots1)$; V checks $\#\phi(0\cdots0) = \#\phi(\overbrace{0\cdots00}^{m-1}) + \#\phi(\overbrace{0\cdots01}^{m-1})$

$\vdots$

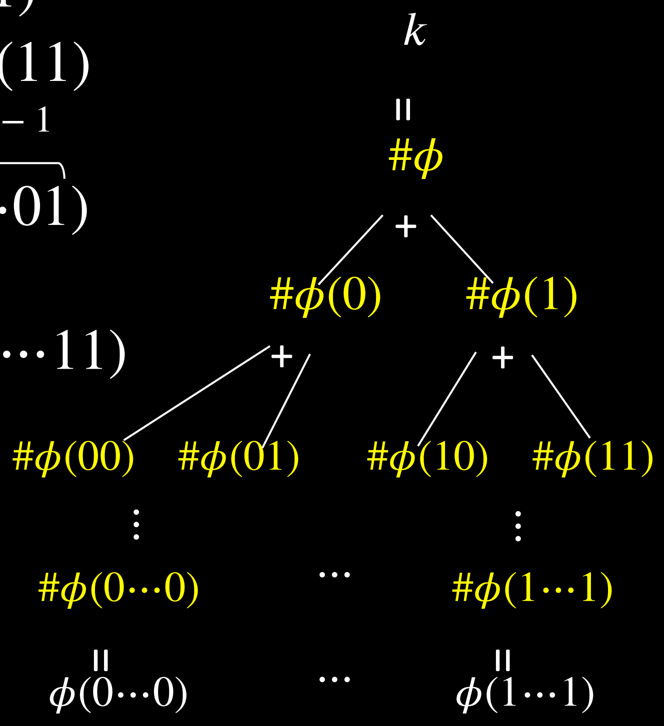
V checks $\#\phi(1\cdots1) = \#\phi(1\cdots10) + \#\phi(1\cdots11)$

$m+1$) V checks $\#\phi(\overbrace{0\cdots0}^{m}) = \phi(0\cdots0)$

$\vdots$

$\#\phi(1\cdots1) = \phi(1\cdots1)$

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



$\#SAT \in$ IP – 1st 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 $\#\phi(0) = \#\phi(00) + \#\phi(01)$

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

$\vdots$

m) P sends $\#\phi(\overbrace{0\cdots0}^{m}), \ldots, \#\phi(1\cdots1)$; V checks $\#\phi(0\cdots0) = \#\phi(\overbrace{0\cdots00}^{m-1}) + \#\phi(\overbrace{0\cdots01}^{m-1})$

$\vdots$

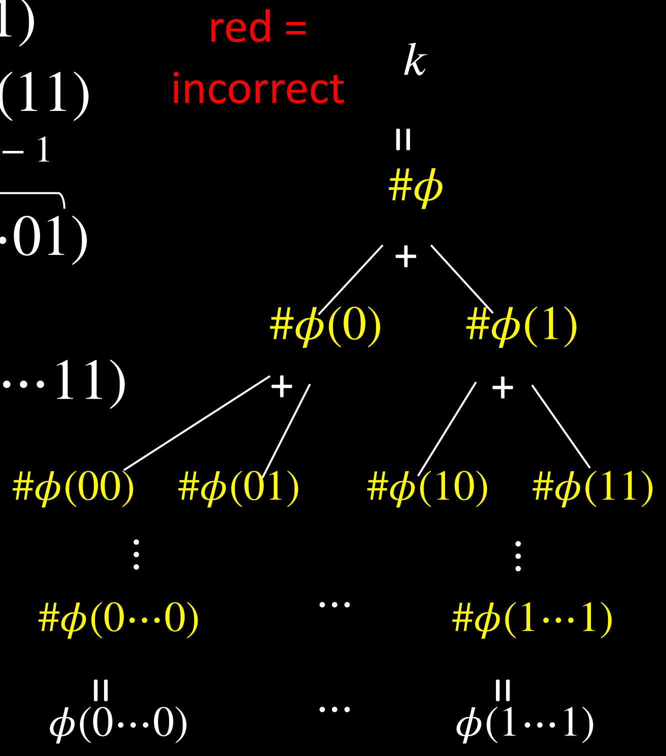
V checks $\#\phi(1\cdots1) = \#\phi(1\cdots10) + \#\phi(1\cdots11)$

$m+1$) V checks $\#\phi(\overbrace{0\cdots0}^{m}) = \phi(0\cdots0)$

$\vdots$

$\#\phi(1\cdots1) = \phi(1\cdots1)$

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



$\#SAT \in$ IP – 1st 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 $\#\phi(0) = \#\phi(00) + \#\phi(01)$

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

$\vdots$

m) P sends $\#\phi(\overbrace{0\cdots0}^{m}), \ldots, \#\phi(1\cdots1)$; V checks $\#\phi(0\cdots0) = \#\phi(\overbrace{0\cdots00}^{m-1}) + \#\phi(\overbrace{0\cdots01}^{m-1})$

$\vdots$

V checks $\#\phi(1\cdots1) = \#\phi(1\cdots10) + \#\phi(1\cdots11)$

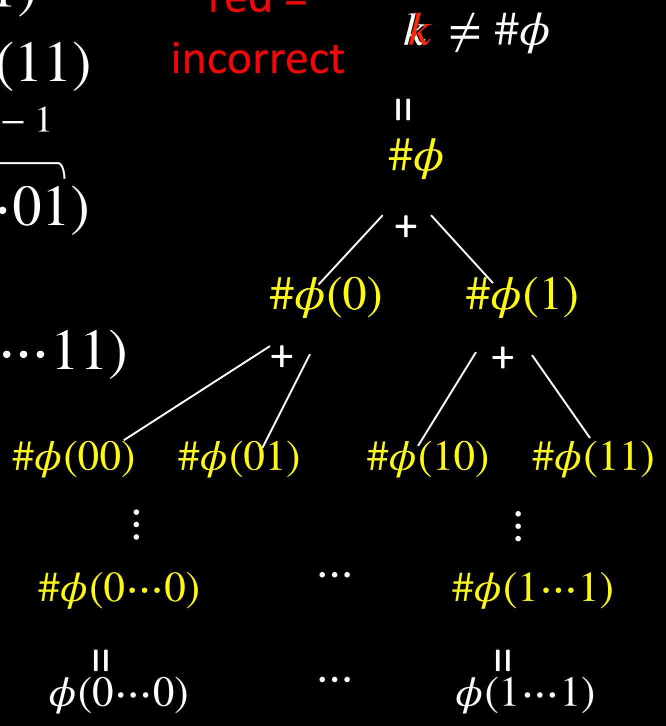
$m + 1$) V checks $\#\phi(\overbrace{0\cdots0}^{m}) = \phi(0\cdots0)$

$\vdots$

$\#\phi(1\cdots1) = \phi(1\cdots1)$

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

red = incorrect



$\#SAT \in$ IP – 1st 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 $\#\phi(0) = \#\phi(00) + \#\phi(01)$

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

$\vdots$

m) P sends $\#\phi(\overbrace{0\cdots0}^{m}), \ldots, \#\phi(1\cdots1)$; V checks $\#\phi(0\cdots0) = \#\phi(\overbrace{0\cdots00}^{m-1}) + \#\phi(\overbrace{0\cdots01}^{m-1})$

$\vdots$

V checks $\#\phi(1\cdots1) = \#\phi(1\cdots10) + \#\phi(1\cdots11)$

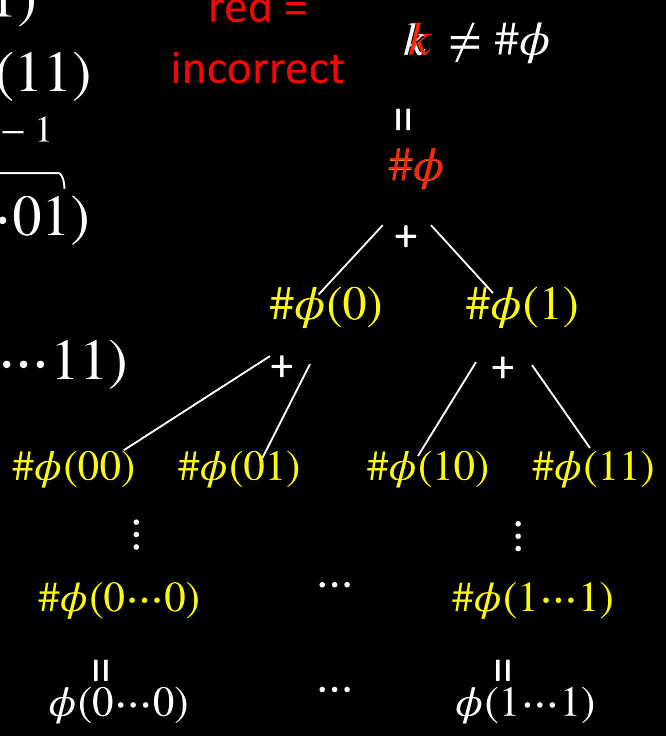
$m+1$) V checks $\#\phi(\overbrace{0\cdots0}^{m}) = \phi(0\cdots0)$

$\vdots$

$\#\phi(1\cdots1) = \phi(1\cdots1)$

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

red = incorrect



$\#SAT \in$ IP – 1st 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 $\#\phi(0) = \#\phi(00) + \#\phi(01)$

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

$\vdots$

m) P sends $\#\phi(\overbrace{0\cdots0}^{m}), \ldots, \#\phi(1\cdots1)$; V checks $\#\phi(0\cdots0) = \#\phi(\overbrace{0\cdots00}^{m-1}) + \#\phi(\overbrace{0\cdots01}^{m-1})$

$\vdots$

V checks $\#\phi(1\cdots1) = \#\phi(1\cdots10) + \#\phi(1\cdots11)$

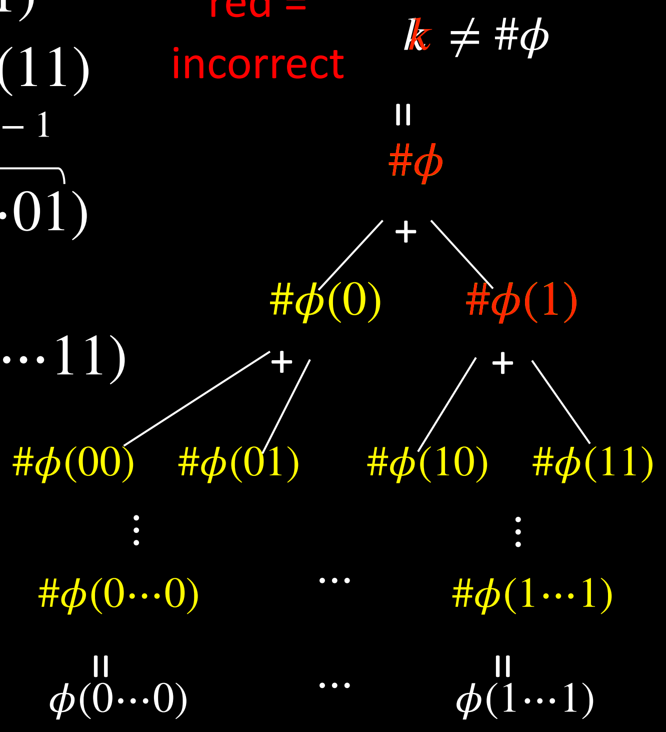
$m+1$) V checks $\#\phi(\overbrace{0\cdots0}^{m}) = \phi(0\cdots0)$

$\vdots$

$\#\phi(1\cdots1) = \phi(1\cdots1)$

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

red = incorrect



$\#SAT \in$ IP – 1st 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 $\#\phi(0) = \#\phi(00) + \#\phi(01)$

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

$\vdots$

m) P sends $\#\phi(\overbrace{0\cdots0}^{m}), \ldots, \#\phi(1\cdots1)$; V checks $\#\phi(0\cdots0) = \#\phi(\overbrace{0\cdots0}^{m-1}0) + \#\phi(\overbrace{0\cdots0}^{m-1}1)$

$\vdots$

V checks $\#\phi(1\cdots1) = \#\phi(1\cdots10) + \#\phi(1\cdots11)$

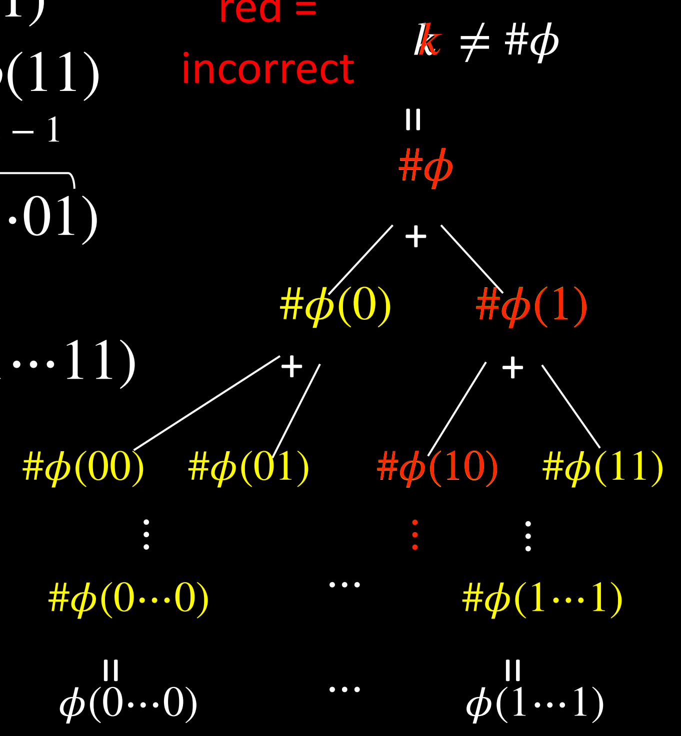
$m+1$) V checks $\#\phi(\overbrace{0\cdots0}^{m}) = \phi(0\cdots0)$

$\vdots$

$\#\phi(1\cdots1) = \phi(1\cdots1)$

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

red = incorrect



$\#SAT \in$ IP – 1st 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 $\#\phi(0) = \#\phi(00) + \#\phi(01)$

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

$\vdots$

m) P sends $\#\phi(\overbrace{0\cdots0}^{m}), \ldots, \#\phi(1\cdots1)$; V checks $\#\phi(0\cdots0) = \#\phi(\overbrace{0\cdots00}^{m-1}) + \#\phi(\overbrace{0\cdots01}^{m-1})$

$\vdots$

V checks $\#\phi(1\cdots1) = \#\phi(1\cdots10) + \#\phi(1\cdots11)$

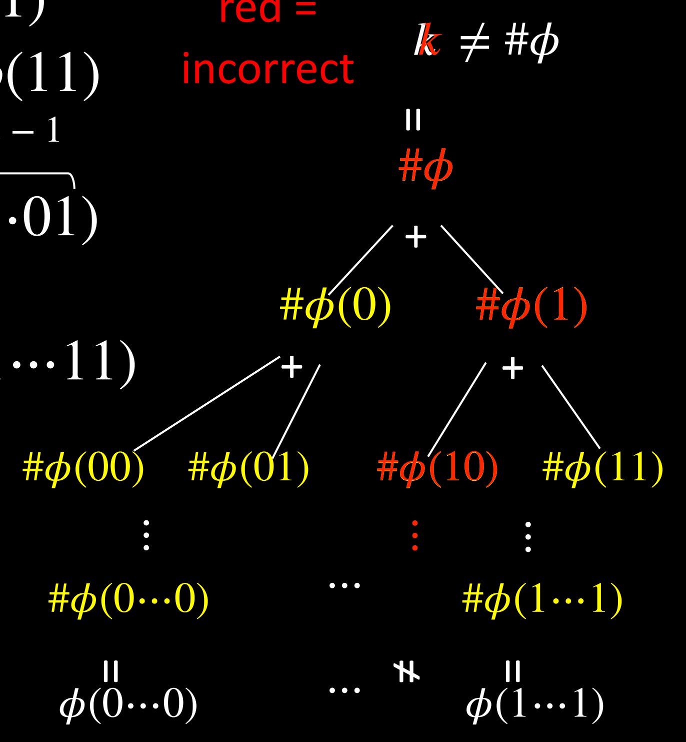
$m+1$) V checks $\#\phi(\overbrace{0\cdots0}^{m}) = \phi(0\cdots0)$

$\vdots$

$\#\phi(1\cdots1) = \phi(1\cdots1)$

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

red = incorrect



$\#SAT \in$ IP – 1st 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 $\#\phi(0) = \#\phi(00) + \#\phi(01)$

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

$\vdots$

m) P sends $\#\phi(\overbrace{0\cdots0}^{m}), \ldots, \#\phi(1\cdots1)$; V checks $\#\phi(0\cdots0) = \#\phi(\overbrace{0\cdots00}^{m-1}) + \#\phi(\overbrace{0\cdots01}^{m-1})$

$\vdots$

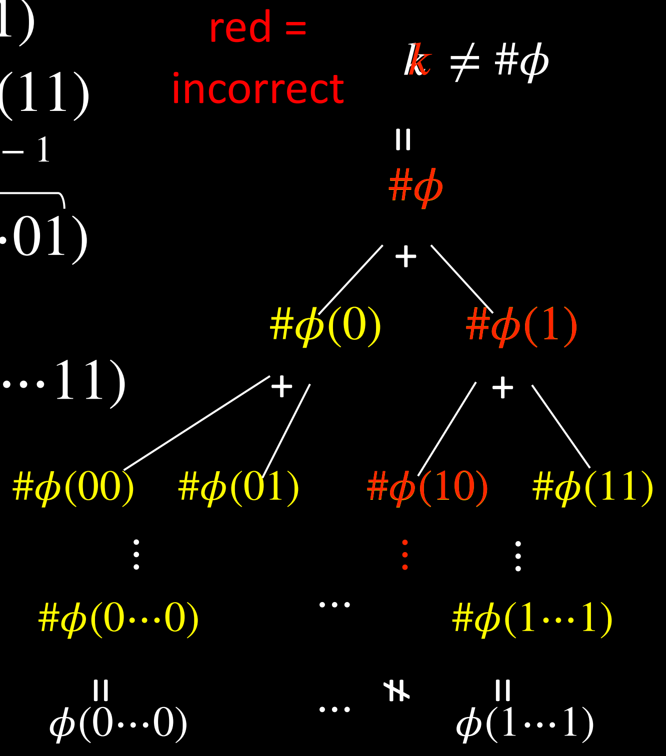
V checks $\#\phi(1\cdots1) = \#\phi(1\cdots10) + \#\phi(1\cdots11)$

$m+1$) V checks $\#\phi(\overbrace{0\cdots0}^{m}) = \phi(0\cdots0)$

$\vdots$

$\#\phi(1\cdots1) = \phi(1\cdots1)$

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



Problem: Exponential. Will fix.

$\#SAT \in$ IP – 1st 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 $\#\phi(0) = \#\phi(00) + \#\phi(01)$

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

$\vdots$

m) P sends $\#\phi(\overbrace{0\cdots0}^{m}), \ldots, \#\phi(1\cdots1)$; V checks $\#\phi(0\cdots0) = \#\phi(\overbrace{0\cdots00}^{m-1}) + \#\phi(\overbrace{0\cdots01}^{m-1})$

$\vdots$

V checks $\#\phi(1\cdots1) = \#\phi(1\cdots10) + \#\phi(1\cdots11)$

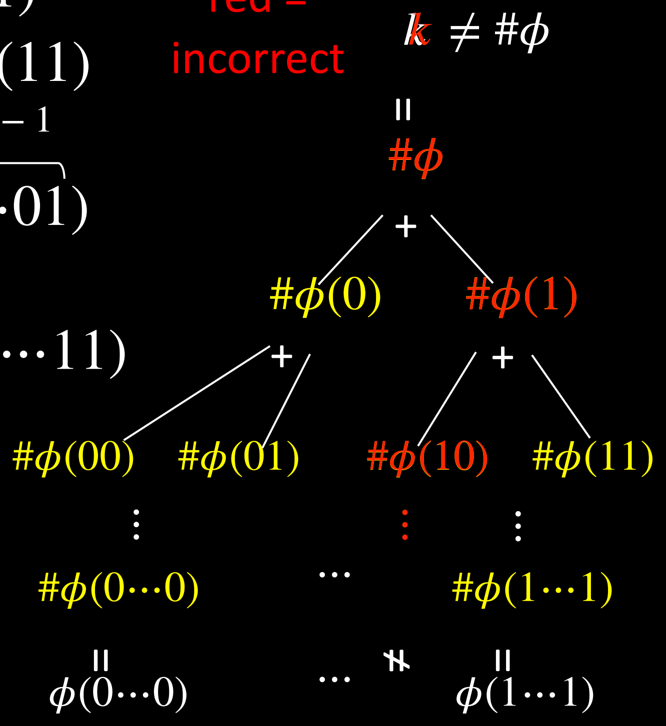
$m+1$) V checks $\#\phi(\overbrace{0\cdots0}^{m}) = \phi(0\cdots0)$

$\vdots$

$\#\phi(1\cdots1) = \phi(1\cdots1)$

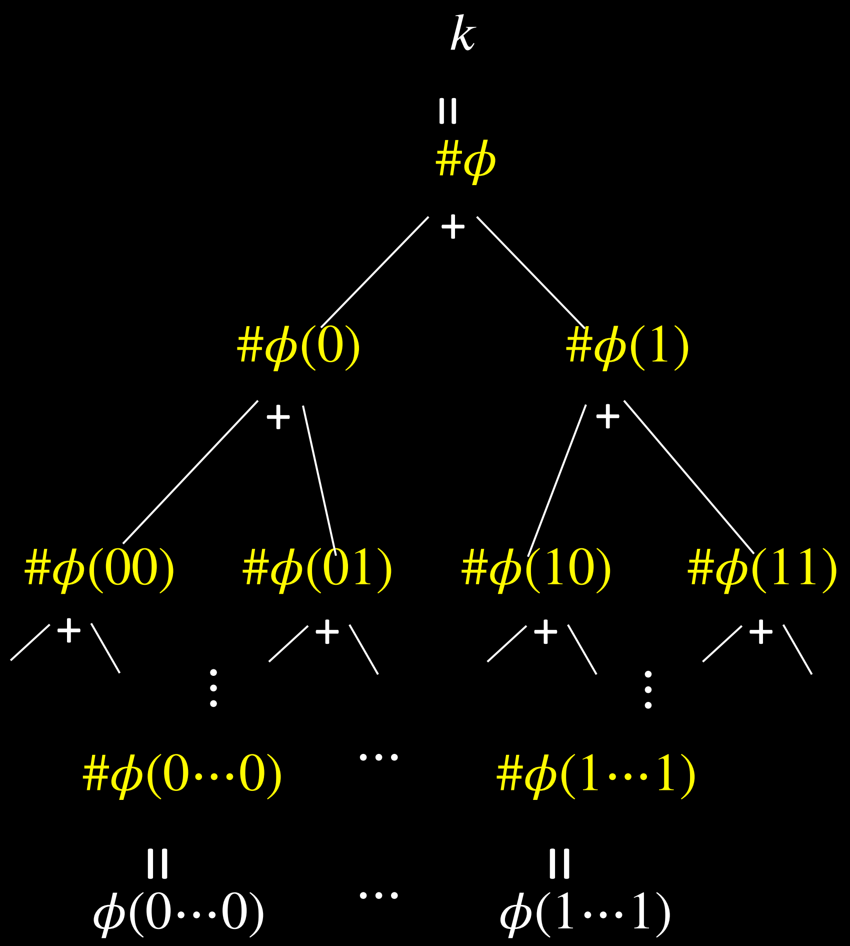
V *accepts* if all checks are correct. Otherwise V *rejects*.

red = incorrect

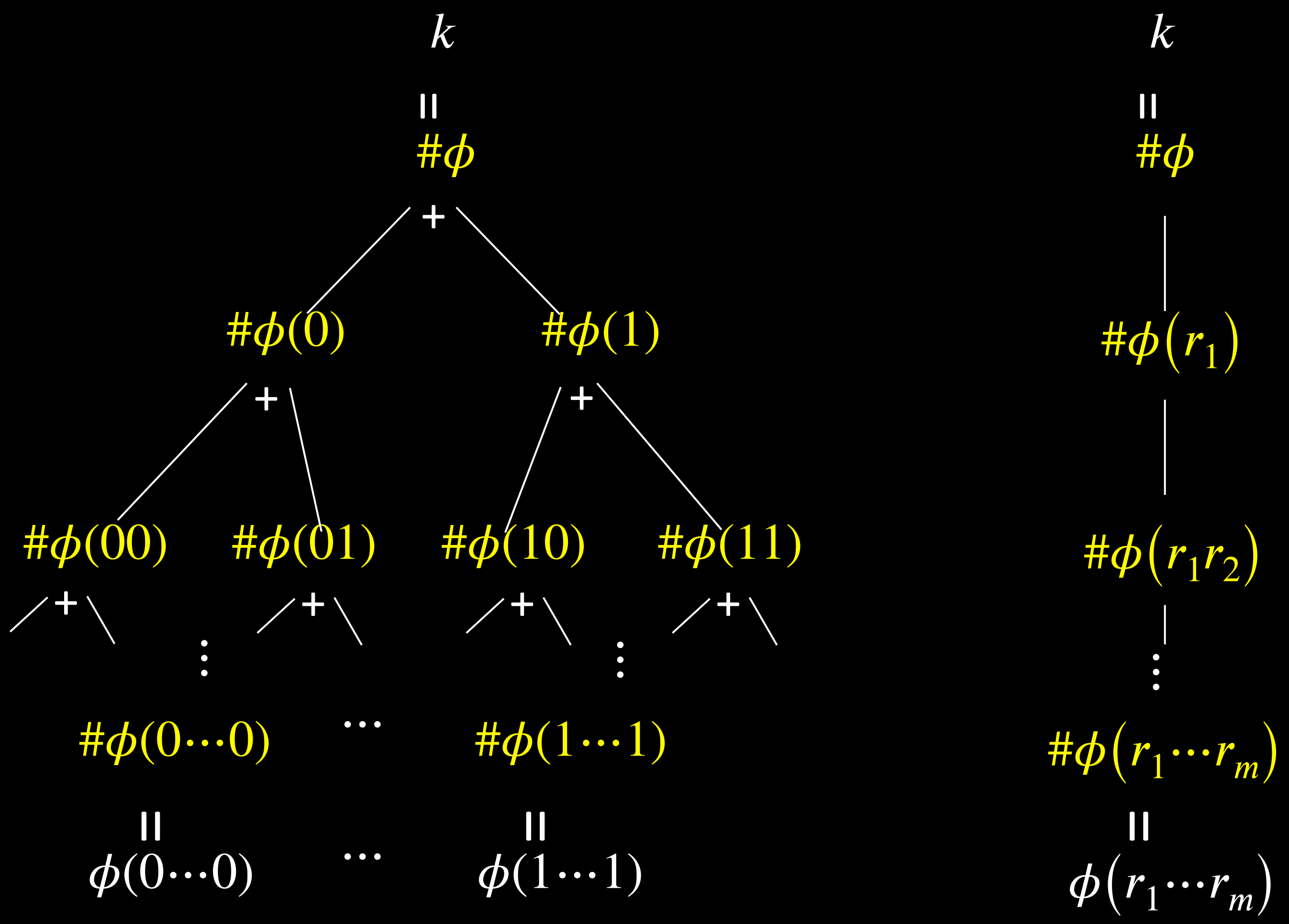


Problem: Exponential. Will fix.

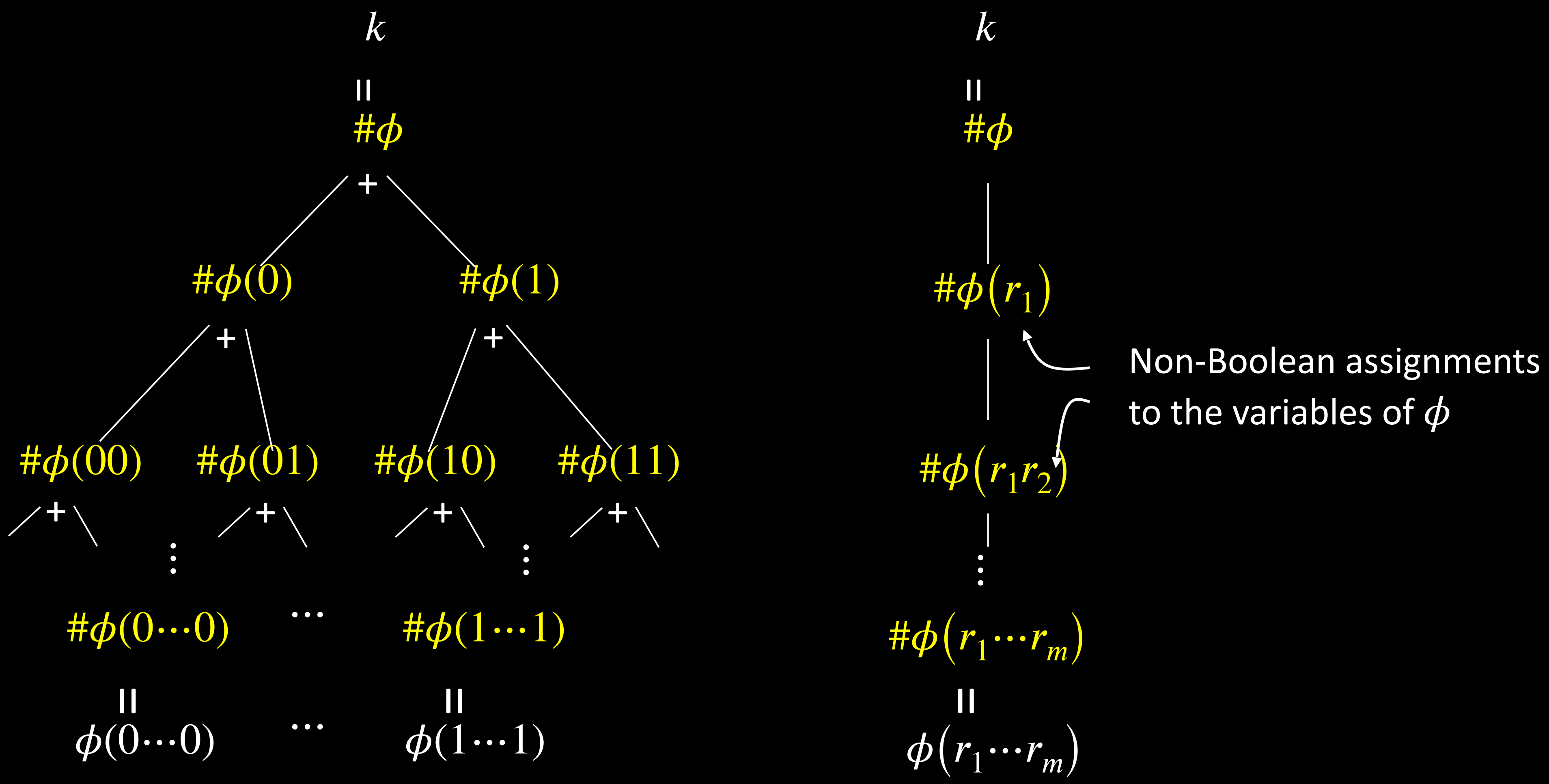
Idea for fixing $\#SAT \in$ IP protocol



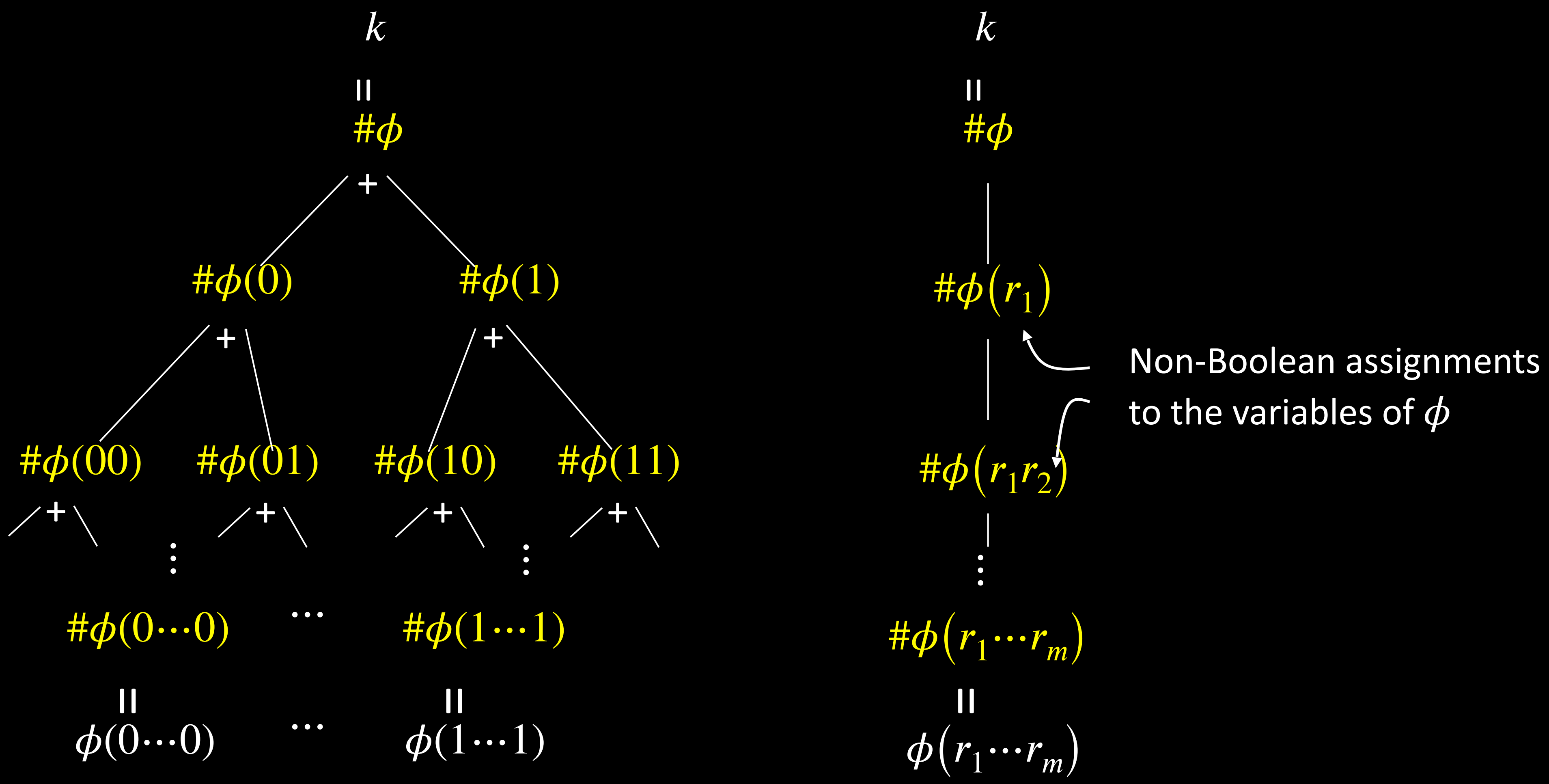
Idea for fixing $\#SAT \in$ IP protocol



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



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



Arithmetizing Boolean formulas

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

$$
\begin{aligned}
a \wedge b \quad &\rightarrow \quad a \times b = ab \\
\overline{a} \quad &\rightarrow \quad (1 - a) \\
a \vee b \quad &\rightarrow \quad a + b - ab
\end{aligned}
$$

Arithmetizing Boolean formulas

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

$$
\begin{aligned}
a \wedge b &\rightarrow a \times b = ab \\
\overline{a} &\rightarrow (1 - a) \\
a \vee b &\rightarrow a + b - ab \\
\phi &\rightarrow p_\phi
\end{aligned}
$$

Arithmetizing Boolean formulas

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

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

Arithmetizing Boolean formulas

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

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

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

Arithmetizing Boolean formulas

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

$$a \wedge b \;\rightarrow\; a \times b = ab$$
$$\bar{a} \;\rightarrow\; (1 - a)$$
$$a \vee b \;\rightarrow\; a + b - ab$$
$$\phi \;\rightarrow\; p_\phi \;\; \text{degree}(p_\phi) \le |\phi|$$

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 $x_1 \cdots x_i = a_1 \cdots a_i$ and remaining $x_{i+1}, \ldots, x_m$ stay as unset variables.

Arithmetizing Boolean formulas

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

$$a \wedge b \;\rightarrow\; a \times b = ab$$
$$\bar{a} \;\rightarrow\; (1-a)$$
$$a \vee b \;\rightarrow\; a + b - ab$$
$$\phi \;\rightarrow\; p_\phi \;\; \text{degree}(p_\phi) \leq \left|\phi\right|$$

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 $x_1 \cdots x_i = a_1 \cdots a_i$ and remaining $x_{i+1},\; \ldots,\; x_m$ stay as unset variables.

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

Arithmetizing Boolean formulas

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

$$a \wedge b \;\rightarrow\; a \times b = ab$$
$$\overline{a} \;\rightarrow\; (1 - a)$$
$$a \vee b \;\rightarrow\; a + b - ab$$
$$\phi \;\rightarrow\; p_\phi \;\; \text{degree}(p_\phi) \;\leq\; \left|\phi\right|$$

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

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

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

Important: For Boolean $a_1 \ldots\, a_i$
the values of $\phi\left(a_1 \ldots a_i\right)$ and $\#\phi(a_1 \ldots\, a_i)$
are unchanged from the previous definition.

Arithmetizing Boolean formulas

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

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

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 $x_1 \cdots x_i = a_1 \cdots a_i$ and remaining $x_{i+1}, \ldots, x_m$ stay as unset variables.

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

Important: For Boolean $a_1 \ldots a_i$ the values of $\phi(a_1 \ldots a_i)$ and $\#\phi(a_1 \ldots a_i)$ are unchanged from the previous definition.

We have <u>extended</u> these functions to non-Boolean values

Arithmetizing Boolean formulas

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

$$a \wedge b \quad \rightarrow \quad a \times b = ab$$
$$\overline{a} \quad \rightarrow \quad (1-a)$$
$$a \vee b \quad \rightarrow \quad a + b - ab$$
$$\phi \quad \rightarrow \quad p_\phi \ \text{degree}(p_\phi) \ \leq \ \left|\phi\right|$$

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

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

identities still true

1. $\#\phi(a_1 \ldots a_i) =$
   $\#\phi\left(a_1 \ldots a_i 0\right) + \#\phi\left(a_1 \ldots a_i 1\right)$
2. $\#\phi(a_1 \ldots a_m) = \ \phi(a_1 \ldots\ a_m)$

Important: For Boolean $a_1 \ldots\ a_i$ the values of $\phi\left(a_1 \ldots a_i\right)$ and $\#\phi(a_1 \ldots\ a_i)$ are unchanged from the previous definition.

We have <u>extended</u> these functions to non-Boolean values

Arithmetizing Boolean formulas

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

$$a \wedge b \rightarrow a \times b = ab$$
$$\bar{a} \rightarrow (1 - a)$$
$$a \vee b \rightarrow a + b - ab$$
$$\phi \rightarrow p_\phi \text{ degree}(p_\phi) \le |\phi|$$

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 $x_1 \cdots x_i = a_1 \cdots a_i$ and remaining $x_{i+1}, \ldots, x_m$ stay as unset variables.

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

identities still true

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

Important: For Boolean $a_1 \ldots a_i$ the values of $\phi(a_1 \ldots a_i)$ and $\#\phi(a_1 \ldots a_i)$ are unchanged from the previous definition.

We have <u>extended</u> these functions to non-Boolean values

Check-in 26.2

Arithmetizing Boolean formulas

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

$$a \wedge b \;\rightarrow\; a \times b = ab$$
$$\bar{a} \;\rightarrow\; (1-a)$$
$$a \vee b \;\rightarrow\; a + b - ab$$
$$\phi \;\rightarrow\; p_\phi \;\; \text{degree}(p_\phi) \le |\phi|$$

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 $x_1 \cdots x_i = a_1 \cdots a_i$ and remaining $x_{i+1},\ \ldots,\ x_m$ stay as unset variables.

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

identities still true

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

Check-in 26.2

Let $\phi = (x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2})$. 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) $p_\phi = (x_1 + x_2 - 2x_1 x_2)$

$\#SAT \in$ IP – version 1

$\#SAT \in$ IP – version 1

Theorem: $\#SAT \in$ IP

$\#SAT \in$ IP – version 1

Theorem: $\#SAT \in$ IP

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

$\#SAT \in$ 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 $k = \#\phi$

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

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

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

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

$\#SAT \in$ IP – version 1

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

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

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

$\#SAT \in$ IP – version 1

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

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 $\#\phi(z)$ 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]

$\#SAT \in$ IP – version 1

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

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

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

V sends random $r_1 \in \mathbb{F}_q$ [P needs to show $\#\phi(r_1)$ is correct]

$\#SAT \in$ IP – version 1

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

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

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

   V sends random $r_1 \in \mathbb{F}_q$ [P needs to show $\#\phi\left(r_1\right)$ is correct]

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

$\#SAT \in$ IP – version 1

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

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

   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 $\#\phi(r_1)$ is correct]

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 $\#\phi(r_1 z)$]

$\#SAT \in$ IP – version 1

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

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

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

V sends random $r_1 \in \mathbb{F}_q$ [P needs to show $\#\phi(r_1)$ is correct]

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 $r_2 \in \mathbb{F}_q$ [P needs to show $\#\phi(r_1 r_2)$ is correct]

$\#SAT \in$ IP – version 1

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

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

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

   V sends random $r_1 \in \mathbb{F}_q$ [P needs to show $\#\phi(r_1)$ is correct]

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 $r_2 \in \mathbb{F}_q$ [P needs to show $\#\phi(r_1 r_2)$ is correct]

$\vdots$

$\#SAT \in$ IP – version 1

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

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

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

V sends random $r_1 \in \mathbb{F}_q$ [P needs to show $\#\phi(r_1)$ is correct]

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 $r_2 \in \mathbb{F}_q$ [P needs to show $\#\phi(r_1 r_2)$ is correct]

⋮

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

$\#SAT \in$ IP – version 1

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

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

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

V sends random $r_1 \in \mathbb{F}_q$ [P needs to show $\#\phi(r_1)$ is correct]

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 $r_2 \in \mathbb{F}_q$ [P needs to show $\#\phi(r_1 r_2)$ is correct]

⋮

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

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

$\#SAT \in$ IP – version 1

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

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

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 $\#\phi(r_1)$ is correct]

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 $r_2 \in \mathbb{F}_q$ [P needs to show $\#\phi(r_1 r_2)$ is correct]

$\vdots$

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 $r_m \in \mathbb{F}_q$

$\#SAT \in$ IP – version 1

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

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

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

   V sends random $r_1 \in \mathbb{F}_q$ [P needs to show $\#\phi\left(r_1\right)$ is correct]

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

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

   V sends random $r_2 \in \mathbb{F}_q$ [P needs to show $\#\phi\left(r_1 r_2\right)$ is correct]

⋮

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

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

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

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

$\#SAT \in$ IP – version 1

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

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

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

V sends random $r_1 \in \mathbb{F}_q$ [P needs to show $\#\phi\left(r_1\right)$ is correct]

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

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

V sends random $r_2 \in \mathbb{F}_q$ [P needs to show $\#\phi\left(r_1 r_2\right)$ is correct]

$\vdots$

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

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

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

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

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

$\#SAT \in$ IP – version 1

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

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

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

V sends random $r_1 \in \mathbb{F}_q$ [P needs to show $\#\phi(r_1)$ is correct]

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 $r_2 \in \mathbb{F}_q$ [P needs to show $\#\phi(r_1 r_2)$ is correct]

⋮

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

$\#SAT \in$ IP – version 2

Input $\langle \phi, k \rangle$

$\#SAT \in$ IP – version 2

Input $\langle \phi, k \rangle$

Prover sends **Verifier sends**

$\#SAT \in$ IP – version 2

Input $\langle \phi, k \rangle$

Prover sends **Verifier sends**

$\#\phi$

$\#SAT \in$ IP – version 2

Input $\langle \phi, k \rangle$

Prover sends

$\#\phi$

Verifier sends

Verifier checks

$\#\phi = k$

$\#SAT \in$ IP – version 2

Input $\langle \phi, k \rangle$

Prover sends

$\#\phi$

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

Verifier sends

Verifier checks

$\#\phi = k$

$\#SAT \in$ IP – version 2

Input $\langle \phi, k \rangle$

Prover sends

Verifier sends

Verifier checks

$\#\phi$

$\#\phi = k$

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

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

$\#SAT \in$ IP – version 2

Input $\langle \phi, k \rangle$

Prover sends

$\#\phi$

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

Verifier sends

Verifier checks

$\#\phi = k$

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

$\#SAT \in$ IP – version 2

Input $\langle \phi, k \rangle$

Prover sends

$\#\phi$

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

Verifier sends

r_1

Verifier checks

$\#\phi = k$

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

$\#SAT \in$ IP – version 2

Input $\langle \phi, k \rangle$

Prover sends

$\#\phi$

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

Verifier sends

r_1

Verifier checks

$\#\phi = k$

$+$

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

$\#\phi\left(r_1\right)$

$\#SAT \in$ IP – version 2

Input $\langle \phi, k \rangle$

Prover sends

$\#\phi$

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

$\#\phi(r_1 z) = \cdots$

Verifier sends

r_1

Verifier checks

$\#\phi = k$

+

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

$\#\phi(r_1)$

$\#SAT \in$ IP – version 2

Input $\langle \phi, k \rangle$

Prover sends

$\#\phi$

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

$\#\phi(r_1 z) = \cdots$

Verifier sends

r_1

Verifier checks

$\#\phi = k$

$+$

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

$\#\phi(r_1)$

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

$\#SAT \in$ IP – version 2

Input $\langle \phi, k \rangle$

Prover sends

$\#\phi$

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

$\#\phi(r_1 z) = \cdots$

Verifier sends

r_1

Verifier checks

$\#\phi = k$

$+$

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

$\#\phi(r_1)$

$+$

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

$\#SAT \in$ IP – version 2

Input $\langle \phi, k \rangle$

Prover sends

$\#\phi$

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

$\#\phi(r_1 z) = \cdots$

Verifier sends

r_1

r_2

Verifier checks

$\#\phi = k$

$+$

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

$\#\phi(r_1)$

$+$

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

$\#SAT \in$ IP – version 2

Input $\langle \phi, k \rangle$

Prover sends

$\#\phi$

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

$\#\phi(r_1 z) = \cdots$

Verifier sends

r_1

r_2

Verifier checks

$\#\phi = k$

$+$

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

$\#\phi(r_1)$

$+$

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

$\#\phi(r_1 r_2)$

$\#SAT \in$ IP – version 2

Input $\langle \phi, k \rangle$

Prover sends

$\#\phi$

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

$\#\phi(r_1 z) = \cdots$

$\#\phi(r_1 r_2 z) = \cdots$

Verifier sends

r_1

r_2

Verifier checks

$\#\phi = k$

$+$

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

$\#\phi(r_1)$

$+$

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

$\#\phi(r_1 r_2)$

$\#SAT \in$ IP – version 2

Input $\langle \phi, k \rangle$

Prover sends

Verifier sends

Verifier checks

$\#\phi$

$\#\phi = k$

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

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

$\#\phi(r_1)$

r_1

$\#\phi(r_1 z) = \cdots$

$\#\phi(r_1) = \#\phi(r_1 0) + \#\phi(r_1 1)$

$\#\phi(r_1 r_2)$

r_2

$\#\phi(r_1 r_2 z) = \cdots$

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

$\#SAT \in$ IP – version 2

Input $\langle \phi, k \rangle$

Prover sends

$\#\phi$

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

$\#\phi(r_1 z) = \cdots$

$\#\phi(r_1 r_2 z) = \cdots$

Verifier sends

r_1

r_2

Verifier checks

$\#\phi = k$

$+$

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

$\#\phi(r_1)$

$+$

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

$\#\phi(r_1 r_2)$

$+$

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

$\#SAT \in$ IP – version 2

Input $\langle \phi, k \rangle$

Prover sends

$\#\phi$

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

$\#\phi(r_1 z) = \cdots$

$\#\phi(r_1 r_2 z) = \cdots$

Verifier sends

r_1

r_2

Verifier checks

$\#\phi = k$

$+$

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

$\#\phi(r_1)$

$+$

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

$\#\phi(r_1 r_2)$

$+$

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

$\vdots$

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

$\#SAT \in$ IP – version 2

Input $\langle \phi, k \rangle$

Prover sends

Verifier sends

Verifier checks

$\#\phi$

$\#\phi = k$

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

$\#\phi(0) \; + \; \#\phi(1)$

$\#\phi(r_1)$

r_1

$\#\phi(r_1 z) = \cdots$

$\#\phi(r_1 0) \; + \; \#\phi(r_1 1)$

$\#\phi(r_1 r_2)$

r_2

$\#\phi(r_1 r_2 z) = \cdots$

$\#\phi(r_1 r_2 0) \; + \; \#\phi(r_1 r_2 1)$

$\vdots$

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

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

$\#SAT \in$ 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

Verifier checks

$$\#\phi = k$$

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

$$\#\phi(r_1)$$

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

$$\#\phi(r_1 r_2)$$

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

$$\vdots$$

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

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

$\#SAT \in$ IP – version 2

Input $\langle \phi, k \rangle$

Prover sends

Verifier sends

Verifier checks

$\#\phi$

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

r_1

$\#\phi(r_1 z) = \cdots$

r_2

$\#\phi(r_1 r_2 z) = \cdots$

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

$$\#\phi = k$$

$$\#\phi(0) \; + \; \#\phi(1)$$

$$\#\phi(r_1)$$

$$\#\phi(r_1 0) \; + \; \#\phi(r_1 1)$$

$$\#\phi(r_1 r_2)$$

$$\#\phi(r_1 r_2 0) \; + \; \#\phi(r_1 r_2 1)$$

$$\vdots$$

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

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

$\#SAT \in$ 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

Verifier checks

$\#\phi = k$

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

$\#\phi(r_1)$

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

$\#\phi(r_1 r_2)$

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

$\vdots$

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

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

$\#SAT \in$ 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

Verifier checks

$\#\phi = k$

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

$\#\phi(r_1)$

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

$\#\phi(r_1 r_2)$

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

$\vdots$

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

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

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

$\#SAT \in$ 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

Verifier checks

$\#\phi = k$

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

$\#\phi(r_1)$

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

$\#\phi(r_1 r_2)$

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

$\vdots$

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

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

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

$\|$

$\phi(r_1 \cdots r_m)$

$\#SAT \in$ 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

accept

Verifier checks

$\#\phi = k$

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

$\#\phi(r_1) = \#\phi(r_1 0) + \#\phi(r_1 1)$

$\#\phi(r_1 r_2) = \#\phi(r_1 r_2 0) + \#\phi(r_1 r_2 1)$

$\vdots$

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

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

$\#SAT \in$ 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

accept

Verifier checks

$\#\phi = k$

$+$

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

$\#\phi(r_1)$

$+$

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

$\#\phi(r_1 r_2)$

$+$

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

$\vdots$

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

$+$

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

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

$\|$

$\phi(r_1 \cdots r_m)$

If k is correct, V will accept.

$\#SAT \in$ 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

accept

Verifier checks

$\#\phi = k$

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

$$\#\phi(r_1) = \#\phi(r_1 0) + \#\phi(r_1 1)$$

$$\#\phi(r_1 r_2) = \#\phi(r_1 r_2 0) + \#\phi(r_1 r_2 1)$$

$\vdots$

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

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

If k is correct, V will accept.

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

$\#SAT \in$ IP – version 2

Input $\langle \phi, k \rangle$

Prover sends	**Verifier sends**	**Verifier checks**
$\#\phi$		$\#\phi = k$
$\#\phi(z) = 3z^d - 5z^{d-1} + \cdots + 7$		$\#\phi(0) + \#\phi(1)$
	r_1	$\#\phi(r_1)$
$\#\phi(r_1 z) = \cdots$		$\#\phi(r_1 0) + \#\phi(r_1 1)$
	r_2	$\#\phi(r_1 r_2)$
$\#\phi(r_1 r_2 z) = \cdots$		$\#\phi(r_1 r_2 0) + \#\phi(r_1 r_2 1)$
		$\vdots$
		$\#\phi(r_1 \cdots r_{m-1})$
$\#\phi(r_1 \cdots r_{m-1} z) = \cdots$		$\#\phi(r_1 \cdots r_{m-1} 0) + \#\phi(r_1 \cdots r_{m-1} 1)$
	r_m	$\#\phi(r_1 \cdots r_m)$
	accept	$\Vert$
		$\phi(r_1 \cdots r_m)$

If k is correct, V will accept.

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

$\#SAT \in$ 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

accept

Verifier checks

$\#\phi = k$

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

$\#\phi(r_1)$

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

$\#\phi(r_1 r_2)$

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

$\vdots$

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

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

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

$\|$

$\phi(r_1 \cdots r_m)$

If k is correct, V will accept.

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

$\#SAT \in$ 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

accept

Verifier checks

$\#\phi = k$

$+$

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

$\#\phi(r_1)$

$+$

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

$\#\phi(r_1 r_2)$

$+$

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

$\vdots$

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

$+$

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

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

$\parallel$

$\phi(r_1 \cdots r_m)$

If k is correct, V will accept.

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

$\#SAT \in$ 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

accept

Verifier checks

$\#\phi = k$

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

$$\#\phi(r_1)$$

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

$$\#\phi(r_1 r_2)$$

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

$$\vdots$$

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

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

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

If k is correct, V will accept.

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

$\#SAT \in$ 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

accept

Verifier checks

$\#\phi = k$

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

$\#\phi(r_1)$

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

$\#\phi(r_1 r_2)$

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

$\vdots$

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

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

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

$\|$

$\phi(r_1 \cdots r_m)$

If k is correct, V will accept.

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

$\#SAT \in$ 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

accept

Verifier checks

$\#\phi = k$

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

$\#\phi(r_1)$

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

$\#\phi(r_1 r_2)$

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

$\vdots$

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

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

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

$\| $

$\phi(r_1 \cdots r_m)$

If k is correct, V will accept.

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

$\#SAT \in$ 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

accept

Verifier checks

$\#\phi = k$

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

$\#\phi(r_1)$

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

$\#\phi(r_1 r_2)$

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

$\vdots$

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

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

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

$\|$

$\phi(r_1 \cdots r_m)$

If k is correct, V will accept.

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

$\#SAT \in$ 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

accept

Verifier checks

$\#\phi = k$

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

$\#\phi(r_1)$

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

$\#\phi(r_1 r_2)$

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

$\vdots$

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

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

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

$\|$

$\phi(r_1 \cdots r_m)$

If k is correct, V will accept.

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

$\#SAT \in$ 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

accept

Verifier checks

$\#\phi = k$

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

$\#\phi(r_1)$

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

$\#\phi(r_1 r_2)$

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

$\vdots$

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

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

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

If k is correct, V will accept.

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

$\#SAT \in$ 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

accept

Verifier checks

$\#\phi = k$

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

$\#\phi(r_1)$

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

$\#\phi(r_1 r_2)$

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

$\vdots$

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

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

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

$\|$

$\phi(r_1 \cdots r_m)$

If k is correct, V will accept.

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

$\#SAT \in$ 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

accept

Verifier checks

$\#\phi = k$

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

$\#\phi(r_1)$

$\#\phi(r_1) = \#\phi(r_1 0) + \#\phi(r_1 1)$

$\#\phi(r_1 r_2)$

$\#\phi(r_1 r_2) = \#\phi(r_1 r_2 0) + \#\phi(r_1 r_2 1)$

$\vdots$

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

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

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

If k is correct, V will accept.

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

$\#SAT \in$ 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

accept

Verifier checks

$\#\phi = k$

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

$\#\phi(r_1)$

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

$\#\phi(r_1 r_2)$

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

$\vdots$

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

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

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

If k is correct, V will accept.

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

$\#SAT \in$ 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

accept

Verifier checks

$\#\phi = k$

$+$

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

$\#\phi(r_1)$

$+$

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

$\#\phi(r_1 r_2)$

$+$

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

$\vdots$

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

$+$

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

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

$\|$

$\phi(r_1 \cdots r_m)$

If k is correct, V will accept.

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

$\#SAT \in$ 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

accept

Verifier checks

$\#\phi = k$

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

$\#\phi(r_1)$

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

$\#\phi(r_1 r_2)$

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

$\vdots$

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

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

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

If k is correct, V will accept.

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

$\#SAT \in$ 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

accept

Verifier checks

$\#\phi = k$

$+$

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

$\#\phi(r_1)$

$+$

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

$\#\phi(r_1 r_2)$

$+$

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

$\vdots$

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

$+$

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

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

$\|$

$\phi(r_1 \cdots r_m)$

If k is correct, V will accept.

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

$\#SAT \in$ IP – version 2

Input $\langle \phi, k \rangle$

Prover sends	**Verifier sends**	**Verifier checks**
$\#\phi$		$\#\phi = k$
$\#\phi(z) = 3z^d - 5z^{d-1} + \cdots + 7$		$\#\phi(0) + \#\phi(1)$
	r_1	$\#\phi(r_1)$
$\#\phi(r_1 z) = \cdots$		$\#\phi(r_1 0) + \#\phi(r_1 1)$
	r_2	$\#\phi(r_1 r_2)$
$\#\phi(r_1 r_2 z) = \cdots$		$\#\phi(r_1 r_2 0) + \#\phi(r_1 r_2 1)$
		$\vdots$
		$\#\phi(r_1 \cdots r_{m-1})$
$\#\phi(r_1 \cdots r_{m-1} z) = \cdots$		$\#\phi(r_1 \cdots r_{m-1} 0) + \#\phi(r_1 \cdots r_{m-1} 1)$
	r_m	$\#\phi(r_1 \cdots r_m)$
	reject	$\neq$ $\phi(r_1 \cdots r_m)$

If k is correct, V will accept.

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

$\#SAT \in$ 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

$\#\phi = k$

$+$

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

$\#\phi(r_1)$

$+$

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

$\#\phi(r_1 r_2)$

$+$

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

$\vdots$

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

$+$

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

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

$\|$

$\phi(r_1 \cdots r_m)$

If k is correct, V will accept.

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

$\#SAT \in$ IP – correctness

Claim: (1) $\langle \phi, k \rangle \in \#SAT \rightarrow$ Pr [(V $\leftrightarrow$ P) accepts $\langle \phi, k \rangle$] $\geq \frac{2}{3}$

(2) $\langle \phi, k \rangle \notin \#SAT \rightarrow$ for any prover $\widetilde{\text{P}}$ Pr [(V $\leftrightarrow$ $\widetilde{\text{P}}$) accepts $\langle \phi, k \rangle$] $\leq \frac{1}{3}$ }

$\#SAT \in$ IP – correctness

Claim: (1) $\langle \phi, k \rangle \in \#SAT \rightarrow$ Pr [(V $\leftrightarrow$ P) accepts $\langle \phi, k \rangle$] $\geq \frac{2}{3}$

(2) $\langle \phi, k \rangle \notin \#SAT \rightarrow$ for any prover $\tilde{\text{P}}$ Pr [(V $\leftrightarrow \tilde{\text{P}}$) accepts $\langle \phi, k \rangle$] $\leq \frac{1}{3}$ }

Proof: (1) All Verifier checks are correct (with the honest P) so V always accepts (Pr = 1).

$\#SAT \in$ IP – correctness

Claim: (1) $\langle \phi, k \rangle \in \#SAT \rightarrow$ Pr [(V $\leftrightarrow$ P) accepts $\langle \phi, k \rangle$] $\geq \frac{2}{3}$

(2) $\langle \phi, k \rangle \notin \#SAT \rightarrow$ for any prover $\widetilde{\text{P}}$ Pr [(V $\leftrightarrow$ $\widetilde{\text{P}}$) accepts $\langle \phi, k \rangle$] $\leq \frac{1}{3}$ }

Proof: (1) All Verifier checks are correct (with the honest P) so V always accepts (Pr = 1).

(2) If V accepts then $\#\phi\left(r_1 \cdots r_m\right)$ is correct even though $\#\phi$ is wrong

$\#SAT \in$ IP – correctness

Claim: (1) $\langle \phi, k \rangle \in \#SAT \rightarrow$ Pr [(V ↔ P) accepts $\langle \phi, k \rangle$] $\geq \frac{2}{3}$

(2) $\langle \phi, k \rangle \notin \#SAT \rightarrow$ for any prover $\widetilde{\text{P}}$ Pr [(V ↔ $\widetilde{\text{P}}$) accepts $\langle \phi, k \rangle$] $\leq \frac{1}{3}$ }

Proof: (1) All Verifier checks are correct (with the honest P) so V always accepts (Pr = 1).

(2) If V accepts then $\#\phi\left(r_1 \cdots r_m\right)$ is correct even though $\#\phi$ is wrong

So $\#\phi\left(r_1 \cdots r_i\right)$ is correct at some first stage i.

$\#SAT \in$ IP – correctness

Claim: (1) $\langle \phi, k \rangle \in \#SAT \to$ Pr [(V ↔ P) accepts $\langle \phi, k \rangle$] $\geq \frac{2}{3}$

(2) $\langle \phi, k \rangle \notin \#SAT \to$ for any prover $\tilde{\text{P}}$ Pr [(V ↔ $\tilde{\text{P}}$) accepts $\langle \phi, k \rangle$] $\leq \frac{1}{3}$ }

Proof: (1) All Verifier checks are correct (with the honest P) so V always accepts (Pr = 1).

(2) If V accepts then $\#\phi(r_1 \cdots r_m)$ is correct even though $\#\phi$ is wrong

So $\#\phi(r_1 \cdots r_i)$ is correct at some first stage i.

$\vdots$

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

$\#SAT \in$ IP – correctness

Claim: (1) $\langle \phi, k\rangle \in \#SAT \rightarrow$ Pr [(V ↔ P) accepts $\langle \phi, k\rangle$] $\geq \frac{2}{3}$

(2) $\langle \phi, k\rangle \notin \#SAT \rightarrow$ for any prover $\widetilde{\text{P}}$ Pr [(V ↔ $\widetilde{\text{P}}$) accepts $\langle \phi, k\rangle$] $\leq \frac{1}{3}$ }

Proof: (1) All Verifier checks are correct (with the honest P) so V always accepts (Pr = 1).

(2) If V accepts then $\#\phi\left(r_1 \cdots r_m\right)$ is correct even though $\#\phi$ is wrong

So $\#\phi\left(r_1 \cdots r_i\right)$ is correct at some first stage i.

$\vdots$

$\#\phi\left(r_1 \cdots r_{i-1}\right)$

$\#\phi\left(r_1 \cdots r_i\right)$

$\vdots$

$\#SAT \in$ IP – correctness

Claim: (1) $\langle \phi, k \rangle \in \#SAT \rightarrow$ Pr [(V $\leftrightarrow$ P) accepts $\langle \phi, k \rangle$] $\geq \frac{2}{3}$

(2) $\langle \phi, k \rangle \notin \#SAT \rightarrow$ for any prover $\widetilde{\text{P}}$ Pr [(V $\leftrightarrow$ $\widetilde{\text{P}}$) accepts $\langle \phi, k \rangle$] $\leq \frac{1}{3}$ }

Proof: (1) All Verifier checks are correct (with the honest P) so V always accepts (Pr = 1).

(2) If V accepts then $\#\phi(r_1 \cdots r_m)$ is correct even though $\#\phi$ is wrong

So $\#\phi(r_1 \cdots r_i)$ is correct at some first stage i.

$\vdots$

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

$+$

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

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

$\vdots$

$\#SAT \in$ IP – correctness

Claim: (1) $\langle \phi, k \rangle \in \#SAT \rightarrow$ Pr [(V $\leftrightarrow$ P) accepts $\langle \phi, k \rangle$] $\geq \frac{2}{3}$

(2) $\langle \phi, k \rangle \notin \#SAT \rightarrow$ for any prover $\widetilde{\text{P}}$ Pr [(V $\leftrightarrow$ $\widetilde{\text{P}}$) accepts $\langle \phi, k \rangle$] $\leq \frac{1}{3}$ }

Proof: (1) All Verifier checks are correct (with the honest P) so V always accepts (Pr = 1).

(2) If V accepts then $\#\phi(r_1 \cdots r_m)$ is correct even though $\#\phi$ is wrong

So $\#\phi(r_1 \cdots r_i)$ is correct at some first stage i.

$\vdots$

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

$+$

$\#\phi(r_1 \cdots r_{i-1} z) \quad \longleftarrow \quad \#\phi(r_1 \cdots r_{i-1} 0) \qquad \#\phi(r_1 \cdots r_{i-1} 1)$

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

$\vdots$

$\#SAT \in$ IP – correctness

Claim: (1) $\langle \phi, k\rangle \in \#SAT \rightarrow$ Pr [(V $\leftrightarrow$ P) accepts $\langle \phi, k\rangle$] $\geq \frac{2}{3}$

(2) $\langle \phi, k\rangle \notin \#SAT \rightarrow$ for any prover $\widetilde{\text{P}}$ Pr [(V $\leftrightarrow \widetilde{\text{P}}$) accepts $\langle \phi, k\rangle$] $\leq \frac{1}{3}$ }

Proof: (1) All Verifier checks are correct (with the honest P) so V always accepts (Pr = 1).

(2) If V accepts then $\#\phi(r_1 \cdots r_m)$ is correct even though $\#\phi$ is wrong

So $\#\phi(r_1 \cdots r_i)$ is correct at some first stage i.

$\vdots$

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

$+$

$\#\phi(r_1 \cdots r_{i-1} z) \qquad \#\phi(r_1 \cdots r_{i-1} z) \longleftarrow \#\phi(r_1 \cdots r_{i-1} 0) \qquad \#\phi(r_1 \cdots r_{i-1} 1)$

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

$\vdots$

$\#SAT \in$ IP – correctness

Claim: (1) $\langle \phi, k \rangle \in \#SAT \rightarrow$ Pr [(V ↔ P) accepts $\langle \phi, k \rangle$] $\geq \frac{2}{3}$

(2) $\langle \phi, k \rangle \notin \#SAT \rightarrow$ for any prover $\tilde{P}$ Pr [(V ↔ $\tilde{P}$) accepts $\langle \phi, k \rangle$] $\leq \frac{1}{3}$ }

Proof: (1) All Verifier checks are correct (with the honest P) so V always accepts (Pr = 1).

(2) If V accepts then $\#\phi(r_1 \cdots r_m)$ is correct even though $\#\phi$ is wrong

So $\#\phi(r_1 \cdots r_i)$ is correct at some first stage i.

$$\vdots$$

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

$$+$$

$$\#\phi(r_1 \cdots r_{i-1} z) \qquad \#\phi(r_1 \cdots r_{i-1} z) \longleftarrow \#\phi(r_1 \cdots r_{i-1} 0) \qquad \#\phi(r_1 \cdots r_{i-1} 1)$$

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

$$\vdots$$

Pr [agree at random $r_i \in \mathbb{F}_q$] $\leq \frac{\text{degree}}{q} \leq \frac{|\phi|}{2^m}$

$\#SAT \in$ IP – correctness

Claim: (1) $\langle \phi, k \rangle \in \#SAT \rightarrow$ Pr [(V $\leftrightarrow$ P) accepts $\langle \phi, k \rangle$] $\geq \frac{2}{3}$

(2) $\langle \phi, k \rangle \notin \#SAT \rightarrow$ for any prover $\widetilde{\text{P}}$ Pr [(V $\leftrightarrow \widetilde{\text{P}}$) accepts $\langle \phi, k \rangle$] $\leq \frac{1}{3}$ }

Proof: (1) All Verifier checks are correct (with the honest P) so V always accepts (Pr = 1).

(2) If V accepts then $\#\phi(r_1 \cdots r_m)$ is correct even though $\#\phi$ is wrong

So $\#\phi(r_1 \cdots r_i)$ is correct at some first stage i.

$$\vdots$$

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

$$+$$

$$\#\phi(r_1 \cdots r_{i-1} z) \qquad \#\phi(r_1 \cdots r_{i-1} z) \longleftarrow \#\phi(r_1 \cdots r_{i-1} 0) \qquad \#\phi(r_1 \cdots r_{i-1} 1)$$

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

$$\vdots$$

Pr [agree at random $r_i \in \mathbb{F}_q$] $\leq \frac{\text{degree}}{q} \leq \frac{|\phi|}{2^m}$

Agreement is necessary for $\#\phi(r_1 \cdots r_i)$ to be correct

$\#SAT \in$ IP – correctness

Claim: (1) $\langle \phi, k \rangle \in \#SAT \to$ Pr [(V $\leftrightarrow$ P) accepts $\langle \phi, k \rangle$] $\geq \frac{2}{3}$

(2) $\langle \phi, k \rangle \notin \#SAT \to$ for any prover $\widetilde{\mathrm{P}}$ Pr [(V $\leftrightarrow$ $\widetilde{\mathrm{P}}$) accepts $\langle \phi, k \rangle$] $\leq \frac{1}{3}$ }

Proof: (1) All Verifier checks are correct (with the honest P) so V always accepts (Pr = 1).

(2) If V accepts then $\#\phi(r_1 \cdots r_m)$ is correct even though $\#\phi$ is wrong

So $\#\phi(r_1 \cdots r_i)$ is correct at some first stage i.

$$\vdots$$

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

$$+$$

$$\#\phi(r_1 \cdots r_{i-1} z) \qquad \#\phi(r_1 \cdots r_{i-1} z) \longleftarrow \#\phi(r_1 \cdots r_{i-1} 0) \qquad \#\phi(r_1 \cdots r_{i-1} 1)$$

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

$$\vdots$$

Pr [agree at random $r_i \in \mathbb{F}_q$] $\leq \dfrac{\text{degree}}{q} \leq \dfrac{|\phi|}{2^m}$

Agreement is necessary for $\#\phi(r_1 \cdots r_i)$ to be correct

So P is "lucky" at stage i if agreement occurs.

$\#SAT \in$ IP – correctness

Claim: (1) $\langle \phi, k \rangle \in \#SAT \rightarrow$ Pr [(V ↔ P) accepts $\langle \phi, k \rangle$] $\geq \frac{2}{3}$

(2) $\langle \phi, k \rangle \notin \#SAT \rightarrow$ for any prover $\widetilde{\mathrm{P}}$ Pr [(V ↔ $\widetilde{\mathrm{P}}$) accepts $\langle \phi, k \rangle$] $\leq \frac{1}{3}$ }

Proof: (1) All Verifier checks are correct (with the honest P) so V always accepts (Pr = 1).

(2) If V accepts then $\#\phi(r_1 \cdots r_m)$ is correct even though $\#\phi$ is wrong

So $\#\phi(r_1 \cdots r_i)$ is correct at some first stage i.

$$\vdots$$

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

$$+$$

$$\#\phi(r_1 \cdots r_{i-1}z) \qquad \#\phi(r_1 \cdots r_{i-1}z) \longleftarrow \#\phi(r_1 \cdots r_{i-1}0) \qquad \#\phi(r_1 \cdots r_{i-1}1)$$

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

$$\vdots$$

Pr [agree at random $r_i \in \mathbb{F}_q$] $\leq \frac{\text{degree}}{q} \leq \frac{|\phi|}{2^m}$

Agreement is necessary for $\#\phi(r_1 \cdots r_i)$ to be correct

So P is "lucky" at stage i if agreement occurs.

Pr [agree at <u>some</u> stage i] $\leq \frac{m|\phi|}{2^m} \leq \frac{1}{m}$ for large m

$\#SAT \in$ IP – correctness

Claim: (1) $\langle \phi, k \rangle \in \#SAT \rightarrow$ Pr [(V ↔ P) accepts $\langle \phi, k \rangle$] $\geq \frac{2}{3}$

(2) $\langle \phi, k \rangle \notin \#SAT \rightarrow$ for any prover $\widetilde{\text{P}}$ Pr [(V ↔ $\widetilde{\text{P}}$) accepts $\langle \phi, k \rangle$] $\leq \frac{1}{3}$ }

Proof: (1) All Verifier checks are correct (with the honest P) so V always accepts (Pr = 1).

(2) If V accepts then $\#\phi(r_1 \cdots r_m)$ is correct even though $\#\phi$ is wrong

So $\#\phi(r_1 \cdots r_i)$ is correct at some first stage i.

$$\vdots$$

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

$$+$$

$$\#\phi(r_1 \cdots r_{i-1}z) \qquad \#\phi(r_1 \cdots r_{i-1}z) \longleftarrow \#\phi(r_1 \cdots r_{i-1}0) \qquad \#\phi(r_1 \cdots r_{i-1}1)$$

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

$$\vdots$$

Pr [agree at random $r_i \in \mathbb{F}_q$] $\leq \frac{\text{degree}}{q} \leq \frac{|\phi|}{2^m}$

Agreement is necessary for $\#\phi(r_1 \cdots r_i)$ to be correct

So P is "lucky" at stage i if agreement occurs.

Pr [agree at <u>some</u> stage i] $\leq \frac{m|\phi|}{2^m} \leq \frac{1}{m}$ for large m

Therefore, for any prover $\widetilde{\text{P}}$ Pr [(V ↔ $\widetilde{\text{P}}$) accepts $\langle \phi, k \rangle$] $\leq \frac{1}{3}$ }

$\#SAT \in$ IP – correctness

Claim: (1) $\langle \phi, k \rangle \in \#SAT \rightarrow$ Pr [(V ↔ P) accepts $\langle \phi, k \rangle$] $\geq \frac{2}{3}$

(2) $\langle \phi, k \rangle \notin \#SAT \rightarrow$ for any prover $\tilde{\text{P}}$ Pr [(V ↔ $\tilde{\text{P}}$) accepts $\langle \phi, k \rangle$] $\leq \frac{1}{3}$ }

Proof: (1) All Verifier checks are correct (with the honest P) so V always accepts (Pr = 1).

(2) If V accepts then $\#\phi(r_1 \cdots r_m)$ is correct even though $\#\phi$ is wrong

So $\#\phi(r_1 \cdots r_i)$ is correct at some first stage i.

$$\vdots$$
$$\#\phi(r_1 \cdots r_{i-1})$$
$$+$$
$$\#\phi(r_1 \cdots r_{i-1} z) \qquad \#\phi(r_1 \cdots r_{i-1} z) \longleftarrow \#\phi(r_1 \cdots r_{i-1} 0) \qquad \#\phi(r_1 \cdots r_{i-1} 1)$$
$$\#\phi(r_1 \cdots r_i)$$
$$\vdots$$

Pr [agree at random $r_i \in \mathbb{F}_q$] $\leq \frac{\text{degree}}{q} \leq \frac{|\phi|}{2^m}$

Agreement is necessary for $\#\phi(r_1 \cdots r_i)$ to be correct

So P is “lucky” at stage i if agreement occurs.

Pr [agree at <u>some</u> stage i] $\leq \frac{m|\phi|}{2^m} \leq \frac{1}{m}$ for large m

Therefore, for any prover $\tilde{\text{P}}$ Pr [(V ↔ $\tilde{\text{P}}$) accepts $\langle \phi, k \rangle$] $\leq \frac{1}{3}$ }

Quick review of today

Quick review of today

Finished $\#SAT \in$ IP and coNP $\subseteq$ IP

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 ?

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 ?