IP=PSPACE

Nikhil Srivastava

CPSC 468/568

Outline

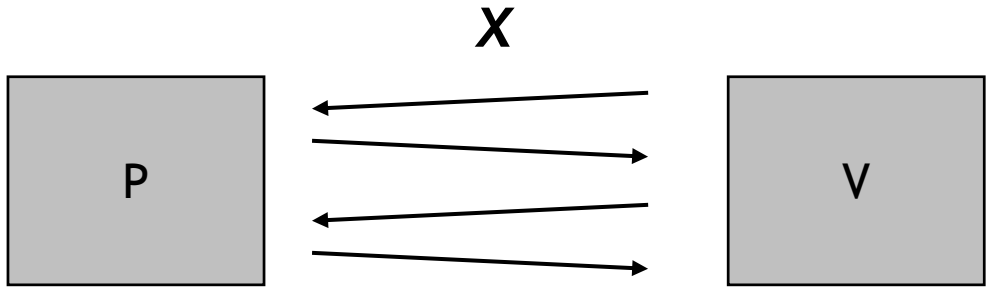
- IP
- Warmup: coNP$\subseteq$IP by arithmetization
- PSPACE
- (wrong) attempt at PSPACE$\subseteq$IP
- (revised) PSPACE$\subseteq$IP
- IP$\subseteq$PSPACE

IP

- NP: $x \in L$ iff $\exists y$ s.t. $V(x,y)=1$, V poly. TM
- IP: 2-player game between

  prover P (computationally unbounded)

  verifier V (probabilistic poly. time)

- 

If $x \in L, \exists P$ Pr[V accepts] $\geq 2/3$ (completeness)

If $x \notin L, \forall P$ Pr[V accepts] $\leq 1/3$ (soundness)

Example

- GNI in IP
- G1 and G2 graphs
- P wants to prove that they are NI

Protocol

1. V picks a random bit b
2. V sends P a random isomorphism of Gb
3. P replies to V which graph he chose

Warmup: coNP $\subseteq$IP

- UN3SAT = $\{\varphi: \neg\exists x_1 x_2 \ldots x_n\ \varphi(x_1, x_2, .., x_n)\}$
- Arithmetization, Ar:{formulas}$\rightarrow \mathbf{F}_p[x_1..x_n]$

$\neg x$	$\rightarrow$	(1-x)
$x \wedge y$	$\rightarrow$	$x \cdot y$
$x \vee y$	$\rightarrow$	x+y

- Ex:

  $Ar[x_1 \wedge (\neg\ x_1 \vee x_2)] = x_1 \cdot ((1-x_1)+x_2)$

- $\varphi(x_1 \ldots x_n)=0$ *iff* $Ar\varphi(x_1 \ldots x_n)=0$ for $x_1 \ldots x_n \in \{0,1\}$

Arithmetization and Quantifiers

Properties:

- Arithmetize $\exists x_1 \varphi(x_1)$ as $\Sigma_{x_1 \in \{0,1\}} h(x_1)$
- φ is unsatisfiable iff

$$q=\Sigma_{x1 \in \{0,1\}} \Sigma_{x2 \in \{0,1\}} \cdots \Sigma_{xn \in \{0,1\}} h(x_1 \ldots x_n)=0$$

- If m= #clauses, $\deg(h) \leq m$
- The above sum is at most 2^n 3^m
- We can work mod p (> 2^n 3^m)

Basic Property

- Q1(x) and Q2(x) two m-degree polynomials in GF(q).
- If $Q1(x) \neq Q2(x)$ then these polynomials agree in at most m points.
- $Pr[Q1(x) = Q2(x)] \leq m/q$

Protocol for UN3SAT

Protocol for UNSAT

P

V

$q_1(x) := \Sigma_{x2,x3\ldots xn \in \{0,1\}} h(x,x_2,\ldots,x_n)$ — q_1 →

check $q_1(0)+q_1(1)=0$

r_1 ← pick $r_1 \in \mathbf{F}_p$

$q_2(x) := \Sigma_{x3\ldots xn \in \{0,1\}} h(r_1,x,x_3,\ldots,x_n)$ — q_2 →

check $q_2(0)+q_2(1)=q_1(r_1)$

r_2 ← pick $r_2 \in \mathbf{F}_p$

$q_3(x) := \Sigma_{x4\ldots xn \in \{0,1\}} h(r_1,r_2,x,\ldots,x_n)$

⋮

$q_n(x) := h(r_1,r_2,\ldots r_{n-1},x)$ — q_n →

check $q_n(0)+q_n(1)=q_{n-1}(r_{n-1})$

check $h(r_1\ldots.r_n)=q_n(r_n)$

In general, the prover tries to convince the verifier that:

$$q_i(r)=q_{i+1}(0)+q_{+1}(1)$$

Completeness: $\varphi \in$ UNSAT, q=0

P

V

$q_1(x) := \Sigma_{x2,x3\ldots xn \in \{0,1\}} h(x, x_2, \ldots, x_n)$

q_1

check $q_1(0)+q_1(1)=0$ ✓

r_1

pick $r_1 \in \mathbf{F}_p$

$q_2(x) := \Sigma_{x3\ldots xn \in \{0,1\}} h(r_1, x, x_3, \ldots, x_n)$

q_2

check $q_2(0)+q_2(1)=q_1(r_1)$ ✓

r_2

pick $r_2 \in \mathbf{F}_p$

$q_3(x) := \Sigma_{x4\ldots xn \in \{0,1\}} h(r_1, r_2, x, \ldots, x_n)$

$q_n(x) := h(r_1, r_2, \ldots r_{n-1}, x)$

q_n

check $q_n(0)+q_n(1)=q_{n-1}(r_{n-1})$ ✓

check $h(r_1 \ldots . r_n)=q_n(r_n)$ ✓

ACCEPT

P

Soundness: $\varphi \notin$ UN3SAT, q>0

V

$q_1(x) := \Sigma_{x2,x3\ldots xn \in \{0,1\}} h(x,x_2,\ldots,x_n)$ q'_1

must send $q'_1 \neq q_1$

check $q'_1(0)+q'_1(1)=0$

Soundness: $\varphi \notin$ UNSAT, q>0

P

V

$q_1(x) := \Sigma_{x2,x3\ldots xn \in \{0,1\}} h(x, x_2, \ldots, x_n)$

send $q'_1 \neq q_1$

q'_1

check $q'_1(0)+q'_1(1)=0$

r_1

pick $r_1 \in \mathbf{F}_p$

$q_2(x) := \Sigma_{x3\ldots xn \in \{0,1\}} h(r_1, x, x_3, \ldots, x_n)$

q'_2

check $q'_2(0)+q'_2(1)=q'_1(r_1)$

$Pr[q'_1(r_1) = q_1(r_1)] \leq m/p$

-> Yay! Send true $q_2'=q_2$

$Pr[q'_1(r_1) \neq q_1(r_1)] \geq 1-m/p$

-> Must lie again…, send $q_2' \neq q_2$

Soundness: $\varphi \notin$ UNSAT, q>0

P

V

$q_1(x) := \Sigma_{x2,x3\ldots xn \in \{0,1\}} h(x,x_2,\ldots,x_n)$

send $q'_1 \neq q_1$

q'_1

check $q'_1(0)+q'_1(1)=0$

r_1

pick $r_1 \in \mathbf{F}_p$

$q_2(x) := \Sigma_{x3\ldots xn \in \{0,1\}} h(r_1,x,x_3,\ldots,x_n)$

Pr[send $q'_2=q_2$] $\leq m/p$,

else send $q'_2 \neq q_2$

q'_2

check $q'_2(0)+q'_2(1)=q'_1(r_1)$

r_2

pick $r_2 \in \mathbf{F}_p$

$q_3(x) := \Sigma_{x4\ldots xn \in \{0,1\}} h(r_1,r_2,x,\ldots,x_n)$

$Pr[q'_2(r_2) = q_2(r_2)] \leq m/p$

-> Yay! Send true $q_3'=q_3$

$Pr[q'_2(r_2) \neq q_2(r_2)] \geq 1-m/p$

-> Must lie again..., send $q_3' \neq q_3$

Soundness: $\varphi \notin$ UNSAT, q>0

P

V

$q_1(x) := \Sigma_{x2,x3\ldots xn \in \{0,1\}} h(x,x_2,\ldots,x_n)$

send $q'_1 \neq q_1$

q'_1

check $q'_1(0)+q'_1(1)=0$

r_1

pick $r_1 \in \mathbf{F}_p$

$q_2(x) := \Sigma_{x3\ldots xn \in \{0,1\}} h(r_1,x,x_3,\ldots,x_n)$

Pr[send $q'_2=q_2$]$\leq m/p$,

else send $q'_2 \neq q_2$

q'_2

check $q'_2(0)+q'_2(1)=q'_1(r_1)$

r_2

pick $r_2 \in \mathbf{F}_p$

$q_3(x) := \Sigma_{x4\ldots xn \in \{0,1\}} h(r_1,r_2,x,\ldots,x_n)$

Pr[send $q'_3=q_3$]$\leq m/p$,

else send $q'_3 \neq q_3$

$q_n(x) := h(r_1,r_2,\ldots r_{n-1},x)$

Pr[send $q'_n=q_n$]$\leq m/p$,

else send $q'_n \neq q_n$

q'_n

check $q'_n(0)+q'_n(1)=q'_{n-1}(r_{n-1})$

check $h(r_1\ldots.r_n)=q'_n(r_n)$

Soundness: $\varphi \notin UNSAT, q>0$

P

V

$q_1(x) := \Sigma_{x2,x3...xn \in \{0,1\}} h(x,x_2,...,x_n)$

send $q'_1 \neq q_1$

q'_1

check $q'_1(0)+q'_1(1)=0$

r_1

pick $r_1 \in \mathbf{F}_p$

$q_2(x) := \Sigma_{x3...xn \in \{0,1\}} h(r_1,x,x_3,...,x_n)$

Pr[send $q'_2=q_2$]$\leq m/p$,

else send $q'_2 \neq q_2$

q'_2

check $q'_2(0)+q'_2(1)=q'_1(r_1)$

r_2

pick $r_2 \in \mathbf{F}_p$

$q_3(x) := \Sigma_{x4...xn \in \{0,1\}} h(r_1,r_2,x,...,x_n)$

Pr[send $q'_3=q_3$]$\leq m/p$,

else send $q'_3 \neq q_3$

$q_n(x) := h(r_1,r_2,...r_{n-1},x)$

Pr[send $q'_n=q_n$]$\leq m/p$,

else send $q'_n \neq q_n$

q'_n

check $q'_n(0)+q'_n(1)=q'_{n-1}(r_{n-1})$

check $h(r_1....r_n)=q'_n(r_n)$

$Pr[h(r_1....r_n)=q_n(r_n)=q'_n(r_n)] \leq m/p$

$\Sigma Pr \leq mn/p$

Soundness: $\varphi \notin$ UNSAT, q>0

P

V

$q_1(x) := \Sigma_{x2,x3\ldots xn \in \{0,1\}} h(x,x_2,\ldots,x_n)$

send $q'_1 \neq q_1$

q'_1

check $q'_1(0)+q'_1(1)=0$

r_1

pick $r_1 \in \mathbf{F}_p$

$q_2(x) := \Sigma_{x3\ldots xn \in \{0,1\}} h(r_1,x,x_3,\ldots,x_n)$

Pr[send $q'_2=q_2$]$\leq m/p$,

else send $q'_2 \neq q_2$

q'_2

check $q'_2(0)+q'_2(1)=q'_1(r_1)$

r_2

pick $r_2 \in \mathbf{F}_p$

$q_3(x) := \Sigma_{x4\ldots xn \in \{0,1\}} h(r_1,r_2,x,\ldots,x_n)$

Pr[send $q'_3=q_3$]$\leq m/p$,

else send $q'_3 \neq q_3$

$q_n(x) := h(r_1,r_2,\ldots r_{n-1},x)$

Pr[send $q'_n=q_n$]$\leq m/p$,

else send $q'_n \neq q_n$

q'_n

check $q'_n(0)+q'_n(1)=q'_{n-1}(r_{n-1})$

check $h(r_1\ldots.r_n)=q'_n(r_n)$

$Pr[h(r_1\ldots.r_n)=q_n(r_n)=q'_n(r_n)] \leq m/p$

Soundness: $\varphi \notin$ UNSAT, q>0

P

V

$q_1(x) := \Sigma_{x2,x3\ldots xn \in \{0,1\}} h(x,x_2,\ldots,x_n)$

send $q'_1 \neq q_1$

q'_1

check $q'_1(0)+q'_1(1)=0$

r_1 pick $r_1 \in \mathbf{F}_p$

$q_2(x) := \Sigma_{x3\ldots xn \in \{0,1\}} h(r_1,x,x_3,\ldots,x_n)$

Pr[send $q'_2=q_2$]$\leq m/p$,

else send $q'_2 \neq q_2$

q'_2

check $q'_2(0)+q'_2(1)=q'_1(r_1)$

r_2 pick $r_2 \in \mathbf{F}_p$

$q_3(x) := \Sigma_{x4\ldots xn \in \{0,1\}} h(r_1,r_2,x,\ldots,x_n)$

Pr[send $q'_3=q_3$]$\leq m/p$,

else send $q'_3 \neq q_3$

$q_n(x) := h(r_1,r_2,\ldots r_{n-1},x)$

Pr[send $q'_n=q_n$]$\leq m/p$,

else send $q'_n \neq q_n$

q'_n

check $q'_n(0)+q'_n(1)=q'_{n-1}(r_{n-1})$

check $h(r_1\ldots.r_n)=q'_n(r_n)$

$Pr[h(r_1\ldots.r_n)=q_n(r_n)=q'_n(r_n)] \leq m/p$

$\Sigma Pr \leq mn/p$

□

PSPACE

- PSPACE = {L decidable by poly. space TM}
- Go, Chess $\in$ PSPACE, PH $\subseteq$PSPACE
- Quantified Boolean Formula:

$$\psi=\forall x_1\ \exists x_2\ \forall x_3 \ldots \exists x_n\ \ \varphi(x_1,x_2,\ldots,x_n)$$

- TQBF = {true ψ}

TQBF is PSPACE-complete

- It is easy to see that it is in PSPACE
- Suppose *L* in PSPACE. Then its TM M uses n^k space and runs in at most 2^(n^k) steps.
- Define $\varphi_i(a,b)$ if configuration *b* is reachable from *a* in at most 2^*i* steps.
- Set *a* the initial configuration of TM M with input x and *b* the unique accepting configuration (if x in L)

- Like in Savitch's Theorem:

$\varphi_{i+1}(a,b)$ iff $\exists c(\varphi_i(a,c) \wedge \varphi_i(c,b))$

- However, this would result in exponentially many clauses.
- Therefore,

$\varphi_{i+1}(a,b)$ iff $\exists c \forall x,y((x,y)=(a,c) \vee (x,y)=(c,b)) \rightarrow \varphi_i(x,y))$

- Larger by a constant additive factor
- Convert to CNF

PSPACE$\subseteq$IP, Naïve attempt

- Use arithmetization for QBFs such that $Q(x_1,x_2,\ldots,x_n)$ =1 or 0, if the QBF is true or false respectively

$$\neg x \rightarrow (1-x)$$

$$x \wedge y \rightarrow x \cdot y$$

$$x \vee y \vee z \rightarrow 1-(1-x)(1-y)(1-z)$$

$$\exists\, x_n\, \varphi(x_1,x_2,\ldots,x_n) \rightarrow 1-(1-Q(..,0))(1-Q(..,1))$$

$$= \sum_{x\in\{0,1\}} Q(x_1,x_2,\ldots,x_n)$$

$$\forall\, x_n\, \varphi(x_1,x_2,\ldots,x_n) \rightarrow Q(..,0)Q(..,1)$$

$$= \prod_{x\in\{0,1\}} Q(x_1,x_2,\ldots,x_n)$$

- Peel off one Σ or Π at a time, checking $1-(1-q_{i+1}(0))(1-q_{i+1}(1))=q_i(r)$ and $q_{i+1}(0)q_{i+1}(1)=q_i(r)$ in alternate rounds.

Problem: Σ and Π double degree

$$q= \sum_{x_1\in\{0,1\}} \prod_{x_2\in\{0,1\}} \ldots \prod_{x_{n-2}\in\{0,1\}} \sum_{x_{n-1}\in\{0,1\}} \prod_{x_n\in\{0,1\}} h(x_1,\ldots,x_n)$$

$2^n(m)$ $2^{n-1}(m)$ … 8(m) 4(m) 2(m) m

Degree Reduction Operator

- $Rx_iU(x_1,\ldots x_i,\ldots,x_n)$ = U with all $x_i^k \rightarrow x_i$
- Formally,

  $Rx_iU(x_1,\ldots x_i,\ldots,x_n) = x_i \cdot U(x_1 \ldots x_{i-1},1,x_{i+1},\ldots x_n) +$

  $(1-x_i)\cdot U(x_1 \ldots x_{i-1},0,x_{i+1},\ldots x_n)$
- Let

  $q=Ex_1Rx_1Ax_2Rx_1Rx_2Ex_3Rx_1Rx_2Rx_3\ldots Ex_nRx_1Rx_2\ldots Rx_n\ h(x_1,\ldots,x_n)$
- Let old *q* (without R)=q_o. Then *q=0* iff q_0*=0*.

PSPACE$\subseteq$IP, revised protocol

- To verify $U(v_1,\ldots v_n)=k$
  - Case $U=ExS(x,v_1,\ldots,v_{n-1})$:
    ask prover for $g(x)=S(x,v_1,\ldots v_n)$.
    check $1-(1-g(0))(1-g(1))=k$
    pick random $r\in\mathbf{F}_p$, send to prover
    recursively verify $S(r,v_1,\ldots v_{n-1})=g(r)$
  - Case $U=AxS(x,v_1,\ldots,v_{n-1})$:
    identical, but test $g(0)g(1)=k$

PSPACE$\subseteq$IP, revised protocol

- To verify $U(v_1,\ldots v_n)=k$
  - Case $U=RxS(x,v_1,\ldots,v_{n-1})$:

    let $g_0(x)$ be previous g, r_0 previous r

    ask prover for $g(x)=S(x,v_1,\ldots,v_n)$

    check $Rxg(x)[r_0]=s_0[r_0]$

    pick random $r\in\mathbf{F}_p$, send to prover

    recursively verify $S(r,v_1,\ldots v_{n-1})=g(r)$

Soundness: q=0

- In each round, an E|A|R is eliminated
- Pr[fooling verifier in a round]≤(degree)/p
- Pr[V accepts]≤(sum of degrees)/p

$$q=\underbrace{Ex_1Rx_1Ax_2Rx_1Rx_2Ex_3Rx_1Rx_2Rx_3\ldots Ex_nRx_1}_{\text{alternating 1 and 2}}\underbrace{Rx_2\ldots Rx_n\ h(x_1,\ldots,x_n)}_{\leq m}$$

- Pr[V accepts]≤$(3nm+n^2)/p\leq 1/3$ for large p

Soundness: q=0

- In each round, an E|A|R is eliminated
- Pr[fooling verifier in a round]≤(degree)/p
- Pr[V accepts]≤(sum of degrees)/p

$$q=\underbrace{Ex_1Rx_1Ax_2Rx_1Rx_2Ex_3Rx_1Rx_2Rx_3\ldots Ex_nRx_1}_{\text{alternating 1 and 2}}\underbrace{Rx_2\ldots Rx_n\ h(x_1,\ldots,x_n)}_{\leq m}$$

- Pr[V accepts]≤$(3nm+n^2)/p\leq 1/3$ for large p

□

IP$\subseteq$PSPACE

- Suppose L $\in$ IP, fix protocol.
- Goal: find $M=\max_P \Pr_P[V \text{ accepts } x]$
- If $M \leq 1/3$ reject, otherwise accept.

IP$\subseteq$PSPACE

- Compute P with `optimal strategy'
- Assume V uses coin flips r
- Transcript := $q_1,a_1,q_2,a_2\ldots.q_n,a_n$

IP$\subseteq$PSPACE

- Define

  $F(q_1,a_1 \ldots q_i) = \max_P \Pr[V \text{ accepts} | q_1,a_1 \ldots q_i]$

IP$\subseteq$PSPACE

- Define

  $F(q_1,a_1 \ldots q_i) = \max_P \Pr[V \text{ accepts} | q_1,a_1 \ldots q_i]$
- *Know that:*

  $F(q_1,\ldots q_n,a_n) = 1$ if V accepts, 0 otherwise
- *Want:*

  $F(q_1) = \max \Pr[V \text{ accepts}] = M$

IP⊆PSPACE

- $F(q_1,a_1,\ldots q_i)=\max_{ai} F(q_1,a_1,\ldots q_i,a_i)$

- $F(q_1,\ldots a_i)= \Sigma_{qi+1} \Pr[q_{i+1} \mid q_1 \ldots a_i] F(q_1,\ldots a_i,q_{i+1})$

IP⊆PSPACE

- $F(q_1,a_1,\ldots q_i)=\max_{ai} F(q_1,a_1,\ldots q_i,a_i)$

- $F(q_1,\ldots a_i)= \Sigma_{qi+1} \Pr[q_{i+1} \mid q_1 \ldots a_i] F(q_1,\ldots a_i,q_{i+1})$

Enumerate $2^{|r|}$ coin flip sequences, simulate V

Compute recursively

IP⊆PSPACE

- $F(q_1,a_1,\ldots q_i)=\max_{ai} F(q_1,a_1,\ldots q_i,a_i)$

- $F(q_1,\ldots a_i)= \Sigma_{qi+1} \Pr[q_{i+1} \mid q_1 \ldots a_i] F(q_1,\ldots a_i,q_{i+1})$

Enumerate $2^{|r|}$ coin flip sequences, simulate V

Compute recursively

□