COMPSCI 501: Formal Language Theory

Lecture 37: IP = PSPACE

Marius Minea
marius@cs.umass.edu

University of Massachusetts Amherst

26 April 2019

Review: Interactive Proofs

Interactive Proofs = Prover (not trustworthy) + Verifier (poly-time)

Verifier V: three inputs, outputs new message m_{i+1}

1. Input string w: decide $w \in A$ or not.
2. Random input: like probabilistic choice (bits from coin flips)
3. Message history: new choice based on past dialog
   $m_1\#m_2\#\ldots\#m_i$
   Output: next message m_{i+1} (to prover), or *accept*, or *reject*

Prover P: takes only input and message history (private coin)

A language A is in **IP** if there exists a verifier (polynomially computable function) V such that for evey string w

1. for *some* function P, $w \in A \implies \Pr[V \leftrightarrow P \text{ accepts } w] \geq \frac{2}{3}$
2. for *any* function $\tilde{P}$, $w \notin A \implies \Pr[V \leftrightarrow P \text{ accepts } w] \leq \frac{1}{3}$

We've seen **IP** $\subseteq$ **PSPACE** (construct "most convincing" prover)

A Simpler Problem: Counting Satisfying Assignments

#SAT $= \{\langle \phi, k\rangle \mid \phi$ is a CNF formula with exactly k satisfuying assignments$\}$

Let's prove SAT $\in$ IP.

Idea: let prover provide truth count, and then "probe" deeper, for truth assignments of *one* variable.

Define $f_i(a_1, \ldots a_i)$ = SAT-count of ϕ with first i variables fixed: $x_1 = a_1, \ldots x_i = a_i$. Our count is $f_0()$.

Then $f_i(a_1, \ldots a_i) = f_{i+1}(a_1, \ldots, a_i, 0) + f_{i+1}(a_1, \ldots, a_i, 1)$.

A Deterministic Protocol

Recall input: $\langle \phi, k\rangle$

Phase 0: $P \to V$: $f_0()$
V checks $k = f_0()$, *rejects* if not.

Phase 1: $P \to V$: $f_1(0), f_1(1)$
V checks $f_0() = f_1(0) + f_1(1)$, *rejects* if not.

Phase 2: $P \to V$: $f_2(0,0), f_2(0,1), f_2(1,0), f_2(1,1)$
V checks $f_1(0) = f_2(0,0) + f_2(0,1)$, $f_1(1) = f_2(1,0) + f_2(1,1)$, *rejects* if not.

. . . What is the problem?

Size of input grows exponentially $\implies$ EXPTIME to process them

However, protocol is correct. Honest prover will lead to accept.
Prover dishonest on some f_i needs to lie on one of two f_{i+1} values $\implies$ will be caught in the end.

Introducing Randomness

What if we check the prover's answer on one of the branches?

Random bitstring: $r_1 r_2 \ldots r_m$.

Phase 0: $P \to V$: $f_0()$
V checks $k = f_0()$, *rejects* if not.

Phase 1: $P \to V$: $f_1(0), f_1(1)$
V checks $f_0() = f_1(0) + f_1(1)$, *rejects* if not, else sends r_1

Phase 2: $P \to V$: $f_2(r_1, 0), f_2(r_1, 1)$
V checks $f_1(r_1) = f_1(r_1, 0) + f_1(r_1, 1)$, *rejects* if not, else sends r_2

What is the probability of catching a dishonest prover?

Does not work.

Arithmetization of Formulas

Checking for booleans does not provide sufficient info.

Assign polynomial $p(x_1, x_2, \ldots, x_m)$ to formula ϕ so it evaluates the same on booleans.

$p(x_i) = x_i$
$p(\neg\alpha) = 1 - p(\alpha)$
$p(\alpha \wedge \beta) = p(\alpha) \cdot p(\beta)$

Define $\vee$ consistent with $\neg$, $\vee$: DeMorgan rules:
$p(\alpha \vee \beta) = p(\neg(\neg\alpha \wedge \neg\beta)) = 1 - (1 - p(\alpha))(1 - p(\beta))$

Important: degree grows polynomially ($\wedge$, $\vee$: sum of degrees).

Also redefine functions f_i as polynomials.

New Protocol for $\#SAT$

Random sequence $r_1 r_2 \dots r_m$, each from (large) field $\mathcal{F}$

Phase 0: $P \to V$: $f_0()$
V checks $k = f_0()$, *rejects* if not.

Phase 1: $P \to V$: polynomial $f_1(z)$ (coefficients)
V checks $f_0() = f_1(0) + f_1(1)$, *rejects* if not, else sends r_1

Phase 2: $P \to V$: polynomial $f_2(r_1, z)$ (coefficients)
V checks $f_1(r_1) = f_1(r_1, 0) + f_1(r_1, 1)$, *rejects* if not, else sends r_2

Does this decide $\#SAT$? Clearly accept if prover P is honest.

IP protocol for $\#$ST: Analysis

If value $\tilde{f}_0()$ in Phase 0 is incorrect, then function $\tilde{f}_1(z)$ sent in Phase 1 is also incorrect.
(Otherwise, sum of two correct values would not match $\tilde{f}_0()$)

Claim: For random value f_1, $\tilde{f}_1(r_1)$ likely incorrect too.

$$\Pr[\tilde{f}_1(r_1) = f_1(r_1)] < \frac{d}{|\mathcal{F}|}$$

Degree $d \le n$, choose field with $|\mathcal{F}| \ge 2^n$ ($\mathbb{Z}_p$ for some prime)
Then $\frac{n}{2^n} < \frac{1}{n^2}$ if $n \ge 10$.

Probability of getting lucky overall: $m \cdot \frac{1}{n^2}$.
Choosing $n > m$ we get $< \frac{1}{n}$, can make arbitrarily small.

Thus, $\#SAT \in$ IP

Finally: PSPACE $\subseteq$ IP

Take PSPACE-complete language *TQBF*, show $\in$ IP.
(*TQBF* = is a quantified boolean formula a tautology?)

$\psi = Q_1 x_1 Q_2 x_2 \cdot Q_m x_m[\phi]$ $\qquad Q_i$ are $\forall$ or $\exists$

Again, define $f_i(a_1, a_2, \dots a_i)$ fixing $a_1, a_2, \dots a_m$
(other quantifiers remain $\implies$ value is 1 or 0).

Arithmetization for quantifiers:

$p(\forall x f(x)) = p(f(0)) \cdot p(f(1))$
$p(\exists x f(x)) = p(\neg \forall x \neg f(x)) = 1 - (1 - p(f(0))) \cdot (1 - p(f(1)))$

Problem: degree *squares* at each quantifier!

Size n formula: $D(n) = D(n/2)^2$ solves to? $D(n) = 2^n$ exponential

Avoiding Exponential Blowup

Linearization: $x^k = x$ for $x \in \{0, 1\}$.

$\implies$ instead of multiplication, use

$L_i(p) = x_i p(x_1, \dots, 1, \dots, x_m) + (1 - x_i) p(x_1, \dots 0, \dots, x_m)$

Linearize between rounds on each variable to get equivalent form.

Protocol rounds: check product for f_i ($\forall$) or $1 - f_i$ ($\exists$)

Choose field $\mathcal{F}$ of size $\ge n^4$ in formula size n.

$O(n^2)$ rounds, degree $\le n$, cheating probability $\le \frac{n \cdot n^2}{n^4} = \frac{1}{n}$