Complexity Theory

Jan Křetínský

Chair for Foundations of Software Reliability
and Theoretical Computer Science
Technical University of Munich

Summer 2016

Based on slides by Jörg Kreiker

Lecture 16

IP = PSPACE

Goal and Plan

Goal

- **IP** $=$ **PSPACE**

Plan

1. **PSPACE** $\subseteq$ **IP** by showing QBF $\in$ **IP**
2. **IP** $\subseteq$ **PSPACE** by computing optimal prover strategies in polynomial space

Agenda

- arithmetization of Boolean formulas
- arithmetization of quantified formulas by linearization
- interactive protocol for QBF

Agenda

- arithmetization of Boolean formulas
- arithmetization of quantified formulas by linearization
- interactive protocol for QBF

Tomorrow

- optimal prover strategy to show **IP** $\subseteq$ **PSPACE**
- a note on graph isomorphism
- summary: interactive proofs incl further reading and context
- outlook: approximation and PCP theorem

Proof Idea

Show that QBF $\in$ **IP**.

This implies **PSPACE** $\subseteq$ **IP** because

Proof Idea

Show that QBF $\in$ **IP**.

This implies **PSPACE** $\subseteq$ **IP** because

- QBF is **PSPACE**-complete
- **IP** closed under polynomial reductions

Proof Idea

Show that QBF $\in$ **IP**.

This implies **PSPACE** $\subseteq$ **IP** because

- QBF is **PSPACE**-complete
- **IP** closed under polynomial reductions

Technique
Turn formulas into polynomials, similar to reduction from 3SAT to ILP: arithmetization.

Setting

- let $\Phi = Q_1 x_1 \ldots Q_n x_n \varphi(x_1, \ldots, x_n)$ be a quantified boolean formula, where φ is in 3CNF with m clauses
- Φ is either true or false
- running example: $\Phi_= = \forall x \exists y \ (x \vee \overline{y}) \wedge (\overline{x} \vee y)$, where the body is written $\varphi_=$
- deciding truth value of Φ is **PSPACE**-complete

Observation

- $x \wedge y$ is satisfiable iff $x \cdot y = 1$ for $x, y \in \{0, 1\}$

Observation

- $x \wedge y$ is satisfiable iff $x \cdot y = 1$ for $x, y \in \{0, 1\}$
- $\overline{x}$ is satisfiable iff $1 - x = 1$

Observation

- $x \wedge y$ is satisfiable iff $x \cdot y = 1$ for $x, y \in \{0, 1\}$
- $\overline{x}$ is satisfiable iff $1 - x = 1$
- $x \vee y$ is satisfiable iff $x + y \geq 1$

Observation

- $x \wedge y$ is satisfiable iff $x \cdot y = 1$ for $x, y \in \{0, 1\}$
- $\overline{x}$ is satisfiable iff $1 - x = 1$
- $x \vee y$ is satisfiable iff $x + y \geq 1$
- note that $x \vee y \equiv x \wedge \overline{y} \vee \overline{x} \wedge y \vee x \wedge y$

Observation

- $x \wedge y$ is satisfiable iff $x \cdot y = 1$ for $x, y \in \{0, 1\}$
- $\overline{x}$ is satisfiable iff $1 - x = 1$
- $x \vee y$ is satisfiable iff $x + y \geq 1$
- note that $x \vee y \equiv x \wedge \overline{y} \vee \overline{x} \wedge y \vee x \wedge y$

$\Rightarrow$ $x \vee y$ is satisfiable iff $x + y - xy = 1$

Arithmetization of Boolean formulas

For Boolean formula $\varphi(x_1, \ldots, x_n)$ we define $ari_\varphi(x_1, \ldots, x_n)$ such that $\varphi(x_1, \ldots, x_n)$ is satisfiable iff $ari_\varphi(x_1, \ldots, x_n)$ is 1 for satisfying assignment of x_i to true/false and the corresponding x_i.

Arithmetization of Boolean formulas

$$
\begin{aligned}
ari_{x_i}(x_1,\ldots,x_n) &= x_i \\
ari_{\overline{\varphi}}(x_1,\ldots,x_n) &= 1 - ari_{\varphi}(x_1,\ldots,x_n) \\
ari_{\varphi_1\wedge\varphi_2}(x_1,\ldots,x_n) &= ari_{\varphi_1}(x_1,\ldots,x_n)\cdot ari_{\varphi_2}(x_1,\ldots,x_n) \\
ari_{\varphi_1\vee\varphi_2}(x_1,\ldots,x_n) &= ari_{\varphi_1}(x_1,\ldots,x_n) + ari_{\varphi_2}(x_1,\ldots,x_n) \\
&\quad - ari_{\varphi_1}(x_1,\ldots,x_n)\cdot ari_{\varphi_2}(x_1,\ldots,x_n)
\end{aligned}
$$

Arithmetization of Boolean formulas

$$
\begin{aligned}
ari_{x_i}(x_1, \ldots, x_n) &= x_i \\
ari_{\overline{\varphi}}(x_1, \ldots, x_n) &= 1 - ari_{\varphi}(x_1, \ldots, x_n) \\
ari_{\varphi_1 \wedge \varphi_2}(x_1, \ldots, x_n) &= ari_{\varphi_1}(x_1, \ldots, x_n) \cdot ari_{\varphi_2}(x_1, \ldots, x_n) \\
ari_{\varphi_1 \vee \varphi_2}(x_1, \ldots, x_n) &= ari_{\varphi_1}(x_1, \ldots, x_n) + ari_{\varphi_2}(x_1, \ldots, x_n) \\
&\quad - ari_{\varphi_1}(x_1, \ldots, x_n) \cdot ari_{\varphi_2}(x_1, \ldots, x_n)
\end{aligned}
$$

Example

$$
\begin{aligned}
ari_{\varphi_=}(x, y) &= (x + (1 - y) - x(1 - y)) \cdot ((1 - x) + y - (1 - x)y) \\
&= (1 - y + xy) \cdot (1 - x + xy) \\
&= 1 - x - y + 3xy - xy^2 - x^2y + x^2y^2 \\
&=: f_=(x, y)
\end{aligned}
$$

Observation

- degree of arithmetization is $\leq 3m$
- crucial for polynomial representation of formulas

What about quantification?

Intuition

- universal quantification corresponds to conjunction corresponds to multiplication
- existential quantification corresponds to disjunction corresponds to addition

What about quantification?

Intuition

- universal quantification corresponds to conjunction corresponds to multiplication
- existential quantification corresponds to disjunction corresponds to addition

- $ari_{\forall x_i.\varphi}(x_1, \ldots, x_i, \ldots, x_n)$ equals $ari_\varphi(x_1, \ldots, 0, \ldots, x_n) \cdot ari_\varphi(x_1, \ldots, 1, \ldots, x_n)$
- $ari_{\exists x_i.\varphi}(x_1, \ldots, x_i, \ldots, x_n)$ equals $ari_\varphi(x_1, \ldots, 0, \ldots, x_n) + ari_\varphi(x_1, \ldots, 1, \ldots, x_n) - ari_\varphi(x_1, \ldots, 0, \ldots, x_n) \cdot ari_\varphi(x_1, \ldots, 1, \ldots, x_n)$

Running Example

Example

$$
\begin{aligned}
ari_{\Phi_=}(x, y) &= ari_{\exists y.\varphi_=}(0, y) \cdot ari_{\exists y.\varphi_=}(1, y) \\
&= (f_=(0,0) + f_=(0,1) - f_=(0,0)f_=(0,1)) \cdot \ldots \\
&= \ldots \\
&= 1
\end{aligned}
$$

Lessons learnt

- $\Phi_{=}$ is true
- degree of polynomial might get exponential in m
- coefficients too

Lessons learnt

- $\Phi_=$ is true
- degree of polynomial might get exponential in *m*
- coefficients too

Rescue

- over $\{0, 1\}$ we have $x^c = x$
- gives rise to linearization
- to get rid of large coefficients: compute over some sufficiently small finite field

Agenda

- arithmetization of Boolean formulas ✓
- arithmetization of quantified formulas by linearization
- interactive protocol for QBF

Linearization

Linearization means reducing all exponents in polynomial to 1.

- $L_y(f(x,y)) = f(x,1)\cdot y + f(x,0)\cdot(1-y)$
- $L_y(f(x,y)$ is linear in y
- $L_y(f(x,y))$ is equivalent to $f(x,y)$ over $\{0,1\}^2$

Example

$$
\begin{aligned}
L_y(f_=(x,y)) &= L_y(1-x-y+3xy-xy^2-x^2y+x^2y^2)\\
&= (1-y)(1-x)+y\cdot(-x+3x-x-x^2+x^2)\\
&= 1-x-y+2xy
\end{aligned}
$$

General form

$$
\begin{aligned}
L_j(f(x_1,\ldots,x_j,\ldots,x_n)) &= f(x_1,\ldots,1,\ldots,x_k)x_j \\
&+ f(x_1,\ldots,0,\ldots,x_k)(1-x_j)
\end{aligned}
$$

Arithmetization

1. arithmetize Boolean body of formula
2. linearize all variables
3. for innermost quantifier apply $ari_\forall x$ (resp. $ari_\exists x$)
4. linearize all but x
5. repeat from 3.

Recursive definition of general arithmetization

$$
\begin{aligned}
f_{n,n}(x_1,\ldots,x_n) &:= ari_{\varphi}(x_1,\ldots,x_n) \\
f_{i,i}(x_1,\ldots,x_i) &:= f_{i+1,0}(x_1,\ldots,x_i,0) f_{i+1,0}(x_1,\ldots,x_i,1) \\
&\quad \text{if } x_{i+1} \text{ universal} \\
f_{i,i}(x_1,\ldots,x_i) &:= f_{i+1,0}(x_1,\ldots,x_i,0) + f_{i+1,0}(x_1,\ldots,x_i,1) \\
&\quad - f_{i+1,0}(x_1,\ldots,x_i,0) f_{i+1,0}(x_1,\ldots,x_i,1) \\
&\quad \text{if } x_{i+1} \text{ existential} \\
f_{i,j}(x_1,\ldots,x_i) &= L_{j+1}(f_{i,j+1}(x_1,\ldots,x_i))
\end{aligned}
$$

Observations

- there are $\mathcal{O}(n^2)$ functions $f_{\cdot,\cdot}$
- functions $f_{n,\cdot}$ have degree at most $3m$
- all other functions have degree of each variable at most 2
- $f_{0,0} = 1$ iff $\Phi \in$ QBF

Agenda

- arithmetization of Boolean formulas ✓
- arithmetization of quantified formulas by linearization ✓
- interactive protocol for QBF

Protocol intuition

- V accepts if $f_{0,0} = 1$
- P needs to convince V of that fact by iterating over all $f_{i,j}$
- V challenges P by choosing random values from a finite field
- P inserts these values into polynomials and return linear function
- V checks that functions adhere to recursive scheme

Initialization

- verifier and prover agree on prime p such that $12|\Phi|^2 < p \leq 24|\Phi|^2$
- all polynomials will be computed in $\mathbb{Z}/p\mathbb{Z}$
- this is a range, where linear functions can be polynomially represented and evaluated
- start: P sends $f_{0,0}$, the prime and the primality proof
- if $f_{0,0} = 1$ then iterate from $i = 1$ and $j = 0$ until both reach n; otherwise reject

$\Rightarrow$ $\mathcal{O}(n^2)$ rounds

Quantor case $j = 0$

- V asks for $f_{i,0}(r_1, \ldots, r_{i-1}, x_i)$
- P sends $f_{i,0}(r_1, \ldots, r_{i-1}, x_i)$
- if x_i is universally quantified, V checks whether

$$f_{i,0}(r_1, \ldots, r_{i-1}, 0) f_{i,0}(r_1, \ldots, r_{i-1}, 1) \equiv_p f_{i-1,i-1}(r_1, \ldots, r_{i-1})$$

- if x_i is existentially quantified, V checks

$$f_{i,0}(r_1, \ldots, r_{i-1}, 0) + f_{i,0}(r_1, \ldots, r_{i-1}, 1) - f_{i,0}(r_1, \ldots, r_{i-1}, 0) f_{i,0}(r_1, \ldots, r_{i-1}, 1) \equiv_p f_{i-1,i-1}(r_1, \ldots, r_{i-1})$$

- V picks random number $r_i \in \mathbb{Z}/p\mathbb{Z}$ and set j to 1

Linearization case $j > 0$

- V asks fo $f_{i,j}(r_1, \ldots, x_j, \ldots, r_i)$
- P sends $f_{i,j}(r_1, \ldots, x_j, \ldots, r_i)$
- V checks

$$
\begin{gathered}
(1 - r_j) f_{i,j}(r_1, \ldots, 0, \ldots, r_i) + \\
r_j f_{i,j}(r_1, \ldots, 1, \ldots, r_i) \\
\equiv_p \\
f_{i,j-1}(r_1, \ldots, r_i)
\end{gathered}
$$

- V picks r_j at random and increases j (or sets j to 0 and increases i)

Finally . . .

P tests whether

$$ari_{\varphi}(r_1, \ldots, r_n) \equiv_p f_{n,n}(r_1, \ldots, r_n)$$

Observations

- P only sends linear functions
- total message length still polynomial
- V can compute linear functions in $\mathbb{Z}/p\mathbb{Z}$
- if $\Phi \in$ QBF P can always convince V by sending correct polynomials

$\Rightarrow$ perfect completeness

- we have public coins

What if $\Phi \notin$ QBF?

An honest prover admits this fact.

A cheating prover can try to send forged polynomials $g_{i,j}(x)$ instead of $f_{i,j}(x_1, \ldots, x, \ldots, x_i)$.

For soundness P must fail to convince V with high probability.

Soundness

- P can cheat in round (i,j) iff $f_{i,j}(x_1, \ldots, x, \ldots, x_i) - g_{i,j}(x) \equiv_p 0$
- that is: iff V by chance picks a root r_k of a polynomial

Soundness

- P can cheat in round (i,j) iff $f_{i,j}(x_1, \ldots, x, \ldots, x_i) - g_{i,j}(x) \equiv_p 0$
- that is: iff V by chance picks a root r_k of a polynomial
- probability to do so in round (i, j) is $q_{i,j} \leq deg(f_{i,j})/p$ since polynomials of degree n have at most n roots

Soundness

- P can cheat in round (i,j) iff $f_{i,j}(x_1, \ldots, x, \ldots, x_i) - g_{i,j}(x) \equiv_p 0$
- that is: iff V by chance picks a root r_k of a polynomial
- probability to do so in round (i, j) is $q_{i,j} \leq deg(f_{i,j})/p$ since polynomials of degree n have at most n roots
- $f_{n,\cdot}$ have degree at most $3m$
- $f_{i<n,\cdot}$ have degree at most 2
- there are $(n+1)(n+2)/2$ polynomials, $n+1$ large ones

$$
\begin{aligned}
Pr[\text{P cheats}] &\leq \Sigma_{i=1}^{n} \Sigma_{j=0}^{i} q_{i,j} \\
&\leq \frac{3m(n+1)}{p} + \frac{n(n+1)}{p} \\
&\leq \frac{4|\Phi|^2}{p} \\
&\leq 1/3
\end{aligned}
$$

Agenda

- arithmetization of Boolean formulas ✓
- arithmetization of quantified formulas by linearization ✓
- interactive protocol for QBF ✓

Agenda

- arithmetization of Boolean formulas ✓
- arithmetization of quantified formulas by linearization ✓
- interactive protocol for QBF ✓

Tomorrow

- optimal prover strategy to show **IP** $\subseteq$ **PSPACE**
- a note on grpah isomorphism
- summary: interactive proofs incl further reading and context
- outlook: approximation and PCP theorem
- evaluation