Complexity Theory

Jan Křetínský

Technical University of Munich

Summer 2019

May 28, 2019

Lecture 17

IP = PSPACE (2)

Goal and Plan

Goal

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

Plan

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

Agenda

- optimal prover strategy to show **IP** $\subseteq$ **PSPACE**
- summary and further reading
- outlook: approximation and PCP theorem

Definition recap

L is in **IP** iff

1. there exists a polynomial p and
2. there exists a poly-time, randomized verifier V

Definition recap

L is in **IP** iff

1. there exists a polynomial p and
2. there exists a poly-time, randomized verifier V

such that for all words $x \in \{0,1\}^*$ holds

- if $x \in L$ then there exists a prover P such that $Pr[out_V\langle P, V\rangle(x) = 1] \geq 2/3$
- if $x \notin L$ then for all provers P holds that $Pr[out_V\langle P, V\rangle(x) = 1] \leq 1/3$

Definition recap

L is in **IP** iff

1. there exists a polynomial p and
2. there exists a poly-time, randomized verifier V

such that for all words $x \in \{0,1\}^*$ holds

- if $x \in L$ then there exists a prover P such that
  $Pr[out_V\langle P, V\rangle(x) = 1] \geq 2/3$
- if $x \notin L$ then for all provers P holds that
  $Pr[out_V\langle P, V\rangle(x) = 1] \leq 1/3$

Moreover, the following is bounded by $p(|x|)$

- the number of random bits chosen by V
- the number of rounds
- the length of each message

Optimal Prover

Let $L \in$ **IP** be arbitrary, we need to show that $L \in$ **PSPACE**.

Optimal Prover

Let $L \in$ **IP** be arbitrary, we need to show that $L \in$ **PSPACE**.
We know that there exist V and p according to definition on previous slide.

Optimal Prover

Let $L \in$ **IP** be arbitrary, we need to show that $L \in$ **PSPACE**.
We know that there exist V and p according to definition on previous slide.
For $x \in \{0, 1\}^n$, we need to compute in polynomial space whether $x \in L$ or $x \notin L$.

Optimal Prover

Let $L \in$ **IP** be arbitrary, we need to show that $L \in$ **PSPACE**.
We know that there exist V and p according to definition on previous slide.
For $x \in \{0,1\}^n$, we need to compute in polynomial space whether $x \in L$ or $x \notin L$.

$$z := \max_{P}\{Pr[out_V\langle P, V\rangle(x) = 1] \mid P \text{ is any prover for } L\}$$

Optimal Prover

Let $L \in$ **IP** be arbitrary, we need to show that $L \in$ **PSPACE**.
We know that there exist V and p according to definition on previous slide.
For $x \in \{0,1\}^n$, we need to compute in polynomial space whether $x \in L$ or $x \notin L$.

$$z := \max_P \{Pr[out_V\langle P, V\rangle(x) = 1] \mid P \text{ is any prover for } L\}$$

z is acceptance probability of optimal prover, inducing the error probability.

Optimal Prover

Let $L \in$ **IP** be arbitrary, we need to show that $L \in$ **PSPACE**.
We know that there exist V and p according to definition on previous slide.
For $x \in \{0,1\}^n$, we need to compute in polynomial space whether $x \in L$ or $x \notin L$.

$$z := \max_P \{Pr[out_V\langle P, V\rangle(x) = 1] \mid P \text{ is any prover for } L\}$$

z is acceptance probability of optimal prover, inducing the error probability.

- if $z \leq 1/3$ then $x \notin L$
- if $z \geq 2/3$ then $x \in L$
- since $L \in$ **IP** other z cannot occur
- maximum taken over finitely many provers for a given x

Recursive computation of z

If we can compute z in polynomial space, we are done.

Recursive computation of z

If we can compute z in polynomial space, we are done.

Recursive algorithm:

- simulate V branching on
  - each random choice of V
  - each possible response of P
- count
  - accepting branches produced by P's optimal response
  - total number of branches
- ratio is z

Doable in polynomial space?

- recursion depth: $p(n)$
- total number of branches: $p(n)^{p(n)}$

$\Rightarrow$ requires polynomially many bits only

- can manage both counters and current branch with a **PSPACE** machine

Agenda

- optimal prover strategy to show **IP** $\subseteq$ **PSPACE** ✓
- summary and further reading
- outlook: approximation and PCP theorem

Summary

- **IP** $=$ **PSPACE**
- **PSPACE** has short interactive proofs (certificates)
- proof of **IP** $\supseteq$ **PSPACE** also showed that we can have
  - public coins
  - perfect completeness

  for each $L \in$ **IP**
- interaction plus randomization seem to add power, whereas each in isolation seemingly does not

Further Reading

- interactive proofs defined in 1985 by *Goldwasser, Micali, Rackoff*. The knowledge complexity of interactive proof systems. SIAM Journal on Computing archive. Volume 18 (1)(1989).
- public coins: *L. Babai* Trading group theory for randomness. STOC 1985.
- survey book: *Oded Goldreich* Computational Complexity. A Conceptual Perspective. `http://www.wisdom.weizmann.ac.il/~oded/cc-drafts.html`

Further Reading

- *Adi Shamir*. IP=PSPACE. Journal of the ACM v.39 n.4, p.878-880.
- outline here followed lecture notes from Brown university: A detailed proof that IP=PSPACE. `http://www.cs.brown.edu/courses/gs019/papers/ip.pdf`
- also nice: Michael Sipser's book Introduction to the Theory of Computation
- essentially covered 8.1 and 8.2 from Arora-Barak book
- an entertaining survey about the development in the beginning of the 90s by *L. Babai*. Transparent proofs and limits to approximations. First European Congress of Mathematicians. 1994.

Outlook

In the beginning of the 90s a lot of things happened quickly...

- Shamir proved that **IP** $=$ **PSPACE**
- one can also allow multiple provers which leads to the complexity class **MIP**
- one accepts only if provers agree
- **MIP** $=$ **NEXP**
- lead to the notion of **PCP**$[q, r]$, where one checks only r entries in a table of answer/query pairs of size 2^q
- it was then shown that **PCP**$[poly, poly]$ $=$ **NEXP** and **PCP**$[\log n, \mathcal{O}(1)]$ $=$ **NP**
- which yields strong results about approximation of **NP**-complete problems
- for instance: consider a $7/8$ approximation of 3SAT

Block structure of lecture

- basic complexity classes
- probabilistic TMs and randomization
- interactive proofs
- approximations and PCP
- parallelization
  - **NC**
  - circuits
  - descriptive complexity