Complexity Theory

Jan Křetínský

Technical University of Munich

Summer 2019

June 3, 2019

Lecture 20

Probabilistically checkable proofs

Goal and plan

Goal

- understand probabilistically checkable proofs,
- know some examples, and
- see the relation (in fact, equivalence) between PCP and hardness of approximation

Plan

- PCP for GNI
- definition: intuition and formalization
- PCP theorem and some obvious consequences
- tool: a more general 3SAT, constraint satisfaction CSP
- PCP theorem $\implies$ gapCSP$[\rho, 1]$ is **NP**-hard
- gapCSP$[\rho, 1]$ is **NP**-hard $\implies$ PCP theorem

PCP: an intuition

What does probabilistically checkable mean?

PCP: an intuition

What does probabilistically checkable mean?

- you want to verify correctness of a proof by only looking at a few bits of it

PCP: an intuition

What does probabilistically checkable mean?

- you want to verify correctness of a proof by only looking at a few bits of it

Which proofs?

PCP: an intuition

What does probabilistically checkable mean?

- you want to verify correctness of a proof by only looking at a few bits of it

Which proofs?

- typically membership in a language

PCP: an intuition

What does probabilistically checkable mean?

- you want to verify correctness of a proof by only looking at a few bits of it

Which proofs?

- typically membership in a language

Why should I care?

PCP: an intuition

What does probabilistically checkable mean?

- you want to verify correctness of a proof by only looking at a few bits of it

Which proofs?

- typically membership in a language

Why should I care?

- because it gives you a tool to prove hardness of approximation

How can it be done?

How can it be done?

Example

- Susan picks some $0 \leq n \leq 10$, Matt wants to know which n
- problem: his vision is blurred, he only sees up to ± 5

How can it be done?

Example

- Susan picks some $0 \leq n \leq 10$, Matt wants to know which n
- problem: his vision is blurred, he only sees up to ± 5

Solution

- Matt: Hey, Susan, why don't you show me $100 \cdot n$ instead?

Can you say this more formally?

- blurred vision $\sim$ we cannot see all bits of a proof

$\Rightarrow$ we can check only a few bits

- proofs can be spread out such that wrong proofs are wrong everywhere
- the definition of PCP will require existence of a proof only
- a correct proof must always be accepted (completeness 1)
- a wrong proof must be rejected with high probability (soundness ρ)

Does it work for real problems?

Does it work for real problems?

- yes, here is a PCP for graph non-isomorphism
- we use our familiar notion of verifier and prover
- albeit both face some limitations (later)

PCP for GNI

Input: graphs G_0, G_1 with n nodes

Verifier	Proof π

PCP for GNI

Input: graphs G_0, G_1 with n nodes

Verifier	Proof π
	• an array π indexed by all graphs with n nodes • $\pi[H]$ contains a if $H \cong G_a$ • otherwise 0 or 1

PCP for GNI

Input: graphs G_0, G_1 with n nodes

Verifier	Proof π
• picks $b \in \{0, 1\}$ at random • picks random permutation $\sigma : [n] \to [n]$ • asks for $b' = \pi[\sigma(G_b)]$ • accepts iff $b' = b$	• an array π indexed by all graphs with n nodes • $\pi[H]$ contains a if $H \cong G_a$ • otherwise 0 or 1

Analysis

- $|\pi|$ is exponential in n
- verifier asks for only one bit
- verifier needs $\mathcal{O}(n)$ random bits
- verifier is a polynomial time TM
- if π is correct, the verifier always accepts
- if π is wrong (e.g. because $G_0 \cong G_1$, then verifier accepts with probability $1/2$

Agenda

- PCP for GNI ✓
- definition: intuition and formalization
- PCP theorem and some obvious consequences
- tool: a more general 3SAT, constraint satisfaction CSP
- PCP theorem $\implies$ gapCSP$[\rho, 1]$ is **NP**-hard
- gapCSP$[\rho, 1]$ is **NP**-hard $\implies$ PCP theorem

PCP system for $L \subseteq \{0,1\}^*$

Input: word $x \in \{0,1\}^n$

Verifier	Prover
1. pick $r(n)$ random bits 2. pick $q(n)$ positions/bits in π 3. based on x and random bits, compute $\Phi : \{0,1\}^{q(n)} \to \{0,1\}$ 4. after receiving proof bits $\pi_1, \ldots, \pi_{q(n)}$ output $\Phi(\pi_1, \ldots, \pi_{q(n)})$	• creates a proof π that $x \in L$ • $\lvert\pi\rvert \in 2^{r(n)} q(n)$ • on request, sends bits of π

- V is a polynomial-time TM
- if $x \in L$ then there exists a proof π s.t. V always accepts
- if $x \notin L$ then V accepts with probability $\leq 1/2$ for all proofs π

PCP$[r(n), q(n)]$

Definition

A language $L \in \{0,1\}^*$ is in PCP$[r(n), q(n)]$ iff there exists a PCP system with $c \cdot r(n)$ random bits and $d \cdot q(n)$ queries for constants $c, d > 0$.

PCP$[r(n), q(n)]$

Definition

A language $L \in \{0,1\}^*$ is in **PCP**$[r(n), q(n)]$ iff there exists a PCP system with $c \cdot r(n)$ random bits and $d \cdot q(n)$ queries for constants $c, d > 0$.

Theorem (THE PCP theorem)

PCP$[\log n, 1] =$ **NP**.

Observations

- GNI $\in$ **PCP**$[poly(n), 1]$
- the soundness parameter is arbitrary and can be amplified by repetition
- **PCP**$[0, 0]$

Observations

- GNI $\in$ **PCP**$[poly(n), 1]$
- the soundness parameter is arbitrary and can be amplified by repetition
- **PCP**$[0, 0] =$ **P**
- **PCP**$[0, \log(n)]$

Observations

- GNI $\in$ **PCP**$[poly(n), 1]$
- the soundness parameter is arbitrary and can be amplified by repetition
- **PCP**$[0, 0] =$ **P**
- **PCP**$[0, \log(n)] =$ **P**
- **PCP**$[0, poly(n)]$

Observations

- GNI $\in$ **PCP**$[poly(n), 1]$
- the soundness parameter is arbitrary and can be amplified by repetition
- **PCP**$[0, 0] =$ **P**
- **PCP**$[0, \log(n)] =$ **P**
- **PCP**$[0, poly(n)] =$ **NP**
- **PCP**$[r(n), q(n)] \subseteq$ **NTIME**$(2^{\mathcal{O}(r(n))} q(n))$

Observations

- GNI $\in$ **PCP**$[poly(n), 1]$
- the soundness parameter is arbitrary and can be amplified by repetition
- **PCP**$[0, 0]$ $=$ **P**
- **PCP**$[0, \log(n)]$ $=$ **P**
- **PCP**$[0, poly(n)]$ $=$ **NP**
- **PCP**$[r(n), q(n)] \subseteq$ **NTIME**$(2^{\mathcal{O}(r(n))} q(n))$

$\Rightarrow$ **PCP**$[\log n, 1] \subseteq$ **NP**

- every problem in **NP** has a polynomial sized proof (certificate), of which we need to check only a constant number of bits
- for 3SAT (and hence for all!) as low as 3!

More remarks

- the Cook-Levin reduction does not suffice to prove the PCP theorem
  - because of soundness
  - even for $x \notin L$, almost all clauses are satisfiable
  - because they describe acceptable computations

More remarks

- the Cook-Levin reduction does not suffice to prove the PCP theorem
  - because of soundness
  - even for $x \notin L$, almost all clauses are satisfiable
  - because they describe acceptable computations
- PCP is inherently different from **IP**
  - proofs can be exponential in PCP
  - PCP: restrictions on queries and random bits
  - IP: restrictions on total message length

$\Rightarrow$ **PCP**$[poly(n), poly(n)] \supseteq$ **IP** $=$ **PSPACE** (in fact equal to **NEXP**)

Agenda

- PCP for GNI ✓
- definition: intuition and formalization ✓
- PCP theorem and some obvious consequences ✓
- tool: a more general 3SAT, constraint satisfaction CSP
- PCP theorem $\implies$ gapCSP[$\rho, 1$] is **NP**-hard
- gapCSP[$\rho, 1$] is **NP**-hard $\implies$ PCP theorem

Constraint satisfaction

3SAT	qCSP
• n Boolean variables	• n Boolean variables
• m clauses	• m general constraints
• each clause has 3 variables	• each constraint is over q variables

CSP remarks

- one can define the fraction of simultaneously satisfiable clauses just as for max3SAT
- each constraint represents a function $\{0,1\}^q \rightarrow \{0,1\}$
- we may assume that all variables are used: $n \leq qm$

$\Rightarrow$ a qCSP instance can be represented using $mq \log(n) 2^q$ bits (polynomial in n, m)

gap-CSP

Definition

$\text{gap} - \text{qCSP}[\rho, 1]$ is **NP**-hard if for every $L \in$ **NP** there is a gap-producing reduction f such that

- $x \in L \implies f(x)$ is satisfiable
- $x \notin L \implies$ at most ρ constraints of $f(x)$ are satisfiable (at the same time)

Agenda

- PCP for GNI ✓
- definition: intuition and formalization ✓
- PCP theorem and some obvious consequences ✓
- tool: a more general 3SAT, constraint satisfaction CSP ✓
- PCP theorem $\implies$ gapCSP[$\rho, 1$] is **NP**-hard
- gapCSP[$\rho, 1$] is **NP**-hard $\implies$ PCP theorem

PCP $\Leftrightarrow$ Hardness of approximation

Theorem

The following two statements are equivalent.

- **NP** $=$ **PCP**$[\log n, 1]$
- *there exist* $0 < \rho < 1$ *and* $q \in \mathbb{N}$ *such that* $\mathrm{gap} - \mathrm{qCSP}[\rho, 1]$ *is* **NP**-*hard.*

PCP $\Leftrightarrow$ Hardness of approximation

Theorem

The following two statements are equivalent.

- $\mathbf{NP} = \mathbf{PCP}[\log n, 1]$
- *there exist* $0 < \rho < 1$ *and* $q \in \mathbb{N}$ *such that* $\mathrm{gap} - \mathrm{qCSP}[\rho, 1]$ *is* **NP**-*hard.*

- this formalizes the equivalence of probabilistically checkable proofs and hardness of approximation
- this is why the PCP theorem was a breakthrough in inapproximability
- gap preservation from CSP to 3SAT is not hard but omitted

$\Rightarrow$

- show that there is a gap-producing reduction f from 3SAT to $\mathrm{gap} - q\mathrm{CSP}[1/2, 1]$

$\Rightarrow$

- show that there is a gap-producing reduction f from 3SAT to $\mathrm{gap} - \mathrm{qCSP}[1/2, 1]$
- by PCP, 3SAT has PCP system with poly. time verifier V, a constant q queries, using $c \log n$ random bits

$\Rightarrow$

- show that there is a gap-producing reduction f from 3SAT to gap – qCSP$[1/2, 1]$
- by PCP, 3SAT has PCP system with poly. time verifier V, a constant q queries, using $c \log n$ random bits
- define $f(x) = \{\psi_r : \{0,1\}^q \rightarrow \{0,1\} \mid r \in \{0,1\}^{c \log n}\}$ such that
- $\psi_r(b_1, \ldots, b_q) = 1$ if V accepts the bits from proof π given by r

$\Rightarrow$

- show that there is a gap-producing reduction f from 3SAT to gap – qCSP$[1/2, 1]$
- by PCP, 3SAT has PCP system with poly. time verifier V, a constant q queries, using $c \log n$ random bits
- define $f(x) = \{\psi_r : \{0,1\}^q \rightarrow \{0,1\} \mid r \in \{0,1\}^{c \log n}\}$ such that
- $\psi_r(b_1, \ldots, b_q) = 1$ if V accepts the bits from proof π given by r
- $f(x)$ is a qCSP of size $2^{c \log n} \in \mathcal{O}(n)$, representable and computable in poly time

$\Rightarrow$

- show that there is a gap-producing reduction f from 3SAT to $\text{gap} - \text{qCSP}[1/2, 1]$
- by PCP, 3SAT has PCP system with poly. time verifier V, a constant q queries, using $c \log n$ random bits
- define $f(x) = \{\psi_r : \{0,1\}^q \to \{0,1\} \mid r \in \{0,1\}^{c \log n}\}$ such that
- $\psi_r(b_1, \ldots, b_q) = 1$ if V accepts the bits from proof π given by r
- $f(x)$ is a qCSP of size $2^{c \log n} \in \mathcal{O}(n)$, representable and computable in poly time
- if $x \in$ 3SAT then there exists proof π s.t. $f(x)$ is satisfiable

$\Rightarrow$

- show that there is a gap-producing reduction f from 3SAT to gap – qCSP$[1/2, 1]$
- by PCP, 3SAT has PCP system with poly. time verifier V, a constant q queries, using $c \log n$ random bits
- define $f(x) = \{\psi_r : \{0,1\}^q \to \{0,1\} \mid r \in \{0,1\}^{c \log n}\}$ such that
- $\psi_r(b_1, \ldots, b_q) = 1$ if V accepts the bits from proof π given by r
- $f(x)$ is a qCSP of size $2^{c \log n} \in \mathcal{O}(n)$, representable and computable in poly time
- if $x \in$ 3SAT then there exists proof π s.t. $f(x)$ is satisfiable
- if $x \notin$ 3SAT then all proofs π satisfy at most $1/2$ of $f(x)$'s constraints

$\Rightarrow$

- show that there is a gap-producing reduction f from 3SAT to $\text{gap} - \text{qCSP}[1/2, 1]$
- by PCP, 3SAT has PCP system with poly. time verifier V, a constant q queries, using $c \log n$ random bits
- define $f(x) = \{\psi_r : \{0,1\}^q \to \{0,1\} \mid r \in \{0,1\}^{c \log n}\}$ such that
- $\psi_r(b_1, \ldots, b_q) = 1$ if V accepts the bits from proof π given by r
- $f(x)$ is a qCSP of size $2^{c \log n} \in \mathcal{O}(n)$, representable and computable in poly time
- if $x \in$ 3SAT then there exists proof π s.t. $f(x)$ is satisfiable
- if $x \notin$ 3SAT then all proofs π satisfy at most $1/2$ of $f(x)$'s constraints

$\Rightarrow$ f is gap-producing

$\Longleftarrow$

- show that for $L \in$ **NP**, there exists a PCP system

$\Leftarrow$

- show that for $L \in$ **NP**, there exists a PCP system
- by assumption there is a gap-producing reduction f from L to $\text{gap} - \text{qCSP}[\rho, 1]$ for some q and ρ

$\Leftarrow$

- show that for $L \in \mathbf{NP}$, there exists a PCP system
- by assumption there is a gap-producing reduction f from L to $\text{gap}-\text{qCSP}[\rho, 1]$ for some q and ρ
  - for $x \in L$: $f(x)$ is satisfiable qCSP $\{\psi_i\}_{i=1}^{m}$
  - for $x \notin L$ at most ρm constraints satisfiable

$\Leftarrow$

- show that for $L \in \mathbf{NP}$, there exists a PCP system
- by assumption there is a gap-producing reduction f from L to $\text{gap} - \text{qCSP}[\rho, 1]$ for some q and ρ
  - for $x \in L$: $f(x)$ is satisfiable qCSP $\{\psi_i\}_{i=1}^{m}$
  - for $x \notin L$ at most ρm constraints satisfiable
- on input x the PCP verifier
  - computes $f(x)$
  - expects proof π to be assignment to $f(x)$'s n variables
  - picks $1 \leq j \leq m$ at random (needs $\log m$ bits!)
  - sets $\Phi = \psi_j$
  - asks for value of q variables of ψ_j

$\Leftarrow$

- show that for $L \in$ **NP**, there exists a PCP system
- by assumption there is a gap-producing reduction f from L to $\mathrm{gap}-\mathrm{qCSP}[\rho, 1]$ for some q and ρ
  - for $x \in L$: $f(x)$ is satisfiable qCSP $\{\psi_i\}_{i=1}^{m}$
  - for $x \notin L$ at most ρm constraints satisfiable
- on input x the PCP verifier
  - computes $f(x)$
  - expects proof π to be assignment to $f(x)$'s n variables
  - picks $1 \leq j \leq m$ at random (needs $\log m$ bits!)
  - sets $\Phi = \psi_j$
  - asks for value of q variables of ψ_j
- if $x \in L$ then V accepts with prob. 1
- if $x \notin L$ then V accepts with prob. ρ

$\Leftarrow$

- show that for $L \in \mathbf{NP}$, there exists a PCP system
- by assumption there is a gap-producing reduction f from L to $\text{gap} - \text{qCSP}[\rho, 1]$ for some q and ρ
  - for $x \in L$: $f(x)$ is satisfiable qCSP $\{\psi_i\}_{i=1}^m$
  - for $x \notin L$ at most ρm constraints satisfiable
- on input x the PCP verifier
  - computes $f(x)$
  - expects proof π to be assignment to $f(x)$'s n variables
  - picks $1 \leq j \leq m$ at random (needs $\log m$ bits!)
  - sets $\Phi = \psi_j$
  - asks for value of q variables of ψ_j
- if $x \in L$ then V accepts with prob. 1
- if $x \notin L$ then V accepts with prob. ρ
- ρ can be amplified to soundness error at most $1/2$ by constant number of repetitions

Recap: Two views of the PCP theorem

prob. checkable proofs		hardness of approximation
PCP verifier V	$\leftrightarrow$	CSP instance
proof π	$\leftrightarrow$	variable assignment
$\lvert\pi\rvert$	$\leftrightarrow$	number of variables in CSP
number of random bits	$\leftrightarrow$	$\log m$, where m is number of clauses
number of queries	$\leftrightarrow$	arity of constraints

What have we learnt?

- probabilistically checkable proofs are proofs with restrictions on the verifier's number of random bits and the number of proof bits queried
- yields a new, robust characterization of **NP**
- is equivalent to **NP**-hardness of gap – qCSP$[\rho, 1]$
- hence to **NP**-hardness of gap – 3SAT$[\rho, 1]$
- hence to **NP**-hardness of approximation for many problems in **NP** (previous lecture)

Up next: Prove that **NP** $\subseteq$ **PCP**$[poly(n), 1]$