CS286.2 Lecture 1: The PCP theorem, hardness of approximation, and multiplayer games

Scribe: Thomas Vidick

The PCP theorem was proved as the result of a long, intense line of work in the 90s exploring the power of interaction. Since then the theorem has been given a second proof, and has found many applications, in particular to hardness of approximation. In this lecture we give three equivalent formulations of the theorem. All three state that a certain problem is NP-complete.

Proof checking version

Our first formulation gives the theorem its name (PCP = Probabilistically Checkable Proof). It states that, provided that one is willing to settle for a probabilistic decision process that errs with small probability, all languages in NP have proofs that can be verified very efficiently: only a constant number of symbols of the proof need to be evaluated! To state this formally we first need the notion of a PCP verifier.

Definition 1 (PCP verifier)**.** *Let $r,q : \mathbb{N} \to \mathbb{N}$ and let $\Sigma = (\Sigma_n)_{n\in\mathbb{N}}$ be a sequence of finite sets.*[1] *A $(r,q)_\Sigma - \mathrm{PCP}$ verifier V is a probabilistic polynomial-time algorithm that takes as input a string $x \in \{0,1\}^*$ and has access to a special tape containing a* proof *$\Pi \in \Sigma_n^*$, where $n = |x|$ is the length of x. The verifier flips at most $r(n)$ random coins, makes at most $q(n)$ queries to the proof, and outputs a single "accept" or "reject" symbol. Given the specification of a verifier V and $x \in \{0,1\}^*$ we let $\omega(V_x)$ be the maximum acceptance probability of V, when its input is x and the maximum is taken over all possible proofs $\Pi \in \Sigma^*$.*

We define an associated complexity class:

Definition 2. *Let $c,s : \mathbb{N} \to [0,1]$ be arbitrary (computable) functions and r,q,Σ as in Definition 1. A language L is in $\mathrm{PCP}_{c,s}[r,q]_\Sigma$ if there exists a $(r,q)_\Sigma - \mathrm{PCP}$ verifier V such that, for every $x \in \{0,1\}^*$:*

- *(Completeness.) If $x \in L$ then $\omega(V_x) \geq c$, i.e. there exists a proof Π such that V_x accepts Π with probability at least c,*
- *(Soundness.) If $x \notin L$ then $\omega(V_x) \leq s$, i.e. for every proof Π, V_x accepts Π with probability at most s.*

[1] Most often we will have $\Sigma_n = \Sigma$ a constant-sized set for all n.

When $c = 1$, $s = 1/2$ and $\Sigma = \{0,1\}$ we use the shorthand $\mathrm{PCP}[r,q]$ *for* $\mathrm{PCP}_{c,s}[r,q]_\Sigma$.

The PCP theorem states that all languages in NP have very efficient verifiers, namely ones that only read a *constant* number of bits from the proof:

Theorem 3 (PCP, proof-checking variant)**.** *The inclusion* $\mathrm{NP} \subseteq \mathrm{PCP}(O(\log n), O(1))$ *holds. Equivalently, given a* $(O(\log n), O(1))_{\{0,1\}}$-PCP *verifier V the problem of deciding between* $\omega(V_x) = 1$ *and* $\omega(V_x) \leq 1/2$ *is* NP-*hard.*

Multiplayer games

We state a second variant of the PCP theorem that is closer to its origins in the study of multiprover interactive proof systems. The line of work that led to the PCP theorem (see Ryan O'Donnell's notes for more on the history of the theorem) was motivated by the study of the power of *interaction*. Starting from NP, allowing the verifier to use randomness as well as to follow a polynomial-time question/answer interaction with an infinitely powerful prover leads to the class IP (Interactive Proofs), which was shown to equal PSPACE by Shamir (building on work by Lund, Fortnow, Karloff and Nisan). Adding a second prover, not allowed to communicate with the first, gives MIP (Multiprover Interactive Proofs). Building upon techniques introduced in the proof of IP = PSPACE (most importantly, *arithmetization*), Babai, Fortnow and Lund proved that MIP = NEXP — an "exponentially larger" class than the class NP from which we started from.

The discovery of the surprising power of randomization, interaction, and multiple provers eventually led to the proof of the PCP theorem. Here we give a statement of the theorem in the language of *multiplayer games*, which provide a convenient framework in which to express "scaled-down" variants (corresponding to interactions involving a single round of question/answers, and a polynomial number of possible questions) of the MIP = NEXP result.

Definition 4. *Let q be an integer. A q-player game G is specified by the following:*

- *Finite question sets* $Q_1, \ldots, Q_q$,
- *A probability distribution* π *on* $Q_1 \times \cdots Q_q$,
- *Finite answer sets* $A_1, \ldots, A_q$,
- *A decision predicate* $V : (A_1 \times \cdots \times A_q) \times (Q_1 \times \cdots \times Q_q) \to \{0,1\}$.

Given a game G, the value $\omega(G)$ of G is defined as the maximum success probability of players $P_1, \ldots, P_q$ in the game, where player P_j is simply a deterministic function $f_j : Q_j \to A_j$.[2] *Formally,*

$$\omega(G) \,=\, \max_{f_j : Q_j \to A_j} \sum_{(q_1,\ldots,q_q) \in Q_1 \times \cdots \times Q_q} \pi(q_1, \ldots, q_q)\, V\big(f_1(q_1), \ldots, f_q(q_q); q_1, \ldots, q_q\big).$$

Our second formulation of the PCP theorem is as follows.

[2] It is not hard to see that allowing the players to use private or even shared randomness would not improve their chances of winning; see the exercises.

Theorem 5 (PCP, games variant)**.** *For any $L \in \mathrm{NP}$ there exists a polynomial-time mapping $x \in \{0,1\}^* \mapsto G_x$ from strings x to q-player games G_x, where $q = O(1)$, such that $x \in L \implies \omega(G_x) = 1$ and $x \notin L \implies \omega(G_x) \leq 1/2$. Equivalently, there exists a constant q such that the problem of deciding whether, given a q-player game G, $\omega(G) = 1$ or $\omega(G) \leq 1/2$, is NP-hard.*

Constraint satisfaction problems

Our third formulation of the PCP theorem uses the language of constraint satisfaction problems, and is the most useful for applications to hardness of approximation.

Definition 6 (CSP)**.** *Let $m, q : \mathbb{N} \to \mathbb{N}$ and let $\Sigma = (\Sigma_n)$ be a sequence of finite sets. A $(m,q)_\Sigma - \mathrm{CSP}$ φ is a collection of m constraints C_j, each acting on at most q out of a total of n variables. If $m = \mathrm{poly}(n)$ and $\Sigma = \{0,1\}$ we refer to $(m,q)_\Sigma - \mathrm{CSP}$s as $q - \mathrm{CSP}$s.*

Given a CSP φ, its value $\omega(\varphi)$ is the maximum fraction of constraints that an assignment can satisfy:

$$\omega(\varphi) = \max_{x_1,\ldots,x_n} \frac{\#\{j : C_j \text{ is satisfied by } (x_i)\}}{m}.$$

The Cook-Levin theorem states that 3-SAT is NP-complete. In other words, the problem of deciding whether $\omega(\varphi) = 1$ or $\omega(\varphi) \leq 1 - 1/m$ for $(m,3)_{\{0,1\}} - \mathrm{CSP}$s is NP-complete. The PCP theorem shows that an a priori much easier, approximation version of this problem remains just as hard:

Theorem 7 (PCP, CSP variant)**.** *For any $L \in \mathrm{NP}$ there exists a polynomial-time mapping $x \in \{0,1\}^* \mapsto \varphi_x$ from strings x to $(m,q) - \mathrm{CSP}$s φ_x, where $m = \mathrm{poly}(n)$ and $q = O(1)$, such that $x \in L \implies \omega(\varphi_x) = 1$ and $x \notin L \implies \omega(\varphi_x) \leq 1/2$. Equivalently, there exists a constant q such that given a $q - \mathrm{CSP}$ φ the problem of deciding between $\omega(\varphi) = 1$ and $\omega(\varphi) \leq 1/2$ is NP-complete.*

With some more work, Hastad showed the following:

Theorem 8. *Let $\varepsilon > 0$. Given a 3-SAT formula φ, it is NP-hard to distinguish between $\omega(\varphi) = 1$ and $\omega(\varphi) \leq 7/8 + \varepsilon$.*

The theorem is optimal in the sense that given any 3-SAT formula there *always* exists an assignment satisfying a fraction $7/8$ of the clauses. To see this, note that a randomly chosen assignment satisfies any 3-SAT clause with probability exactly $7/8$. By linearity of expectation, the expected fraction of clauses satisfied by a random assignment is $7/8$, hence there must exist an assignment satisfying as many clauses.

Equivalence between the three versions

We show that the three versions of the PCP theorem are equivalent.

Theorem 3 $\Longrightarrow$ Theorem 7. Let L be any language in NP and $x \in \{0,1\}^*$. By Theorem 3 we know that there exists a PCP verifier $V = V_x$ that flips $r = r(|x|)$ random coins, makes $q = q(|x|)$ queries to a proof Π, and is such that, if $x \in L$ then there exists a proof Π that V_x accepts with probability 1, whereas if $x \notin L$ any proof Π is accepted by V_x with probability at most $1/2$.

Consider the following constraint satisfaction problem φ_x. φ_x has as many variables as there are locations in the proof Π_x, which without loss of generality is at most $q2^r = \text{poly}(|x|)$. For each possible string $R \in \{0,1\}^r$ we introduce a constraint C_R which corresponds to the check performed by V_x on this choice of random bits. Precisely, whenever the verifier's random bits correspond to R, he looks up q entries of the proof Π and accepts if and only if these entries satisfy a certain constraint. The constraint C_j is defined to evaluate to 1 if and only if the variables on which C_j acts are assigned values that would have made the verifier V_x accept, if the entries of Π he looks up were assigned matching values.

If $x \in L$ we know that there exists a proof Π that is accepted by V_x. In other words, there is an assignment to the variables of φ_x (the corresponding entries of Π) that satisfies all the clauses (the verifier's test on any possible random string). Hence $\omega(\varphi_x) = 1$.

If $x \notin L$ then every proof Π is rejected by V_x with probability at least $1/2$. Suppose for contradiction that $\omega(\varphi_x) > 1$. This means that there would exist an assignment to all variables that satisfies strictly more than $1/2$ of the constraints. We can define a corresponding proof Π that lists all the variables' values. By definition of φ_x from V_x the proof Π would satisfy strictly more than $1/2$ of the verifier's possible tests, contradicting our assumption that V_x rejects any proof with probability at least $1/2$. Hence it must be that $\omega(\varphi_x) \leq 1/2$, which concludes the proof.