

Lecture 2

Lecturer: Irit Dinur

Scribe: Ariel Procaccia

Last week we defined an (r, q) -restricted verifier as a verifier which uses $O(r)$ random bits, queries $O(q)$ bits from the proof, and runs in polynomial time. We also stated that a language L is in $\mathbf{PCP}[r, q]$ if and only if there exists an (r, q) -restricted verifier V which satisfies the following conditions:

- **Completeness:** $x \in L \Rightarrow \exists \pi \text{ s.t. } \Pr_R[V^\pi(x, R) = YES] = 1$
- **Soundness:** $x \notin L \Rightarrow \forall \pi : \Pr_R[V^\pi(x, R) = YES] < \frac{1}{2}$

Notice that clearly $\mathbf{NP} = \mathbf{PCP}[0, \text{poly}(n)]$. Moreover, we proved that $\mathbf{PCP}[\log(n), 1] \subset \mathbf{NP}$. Our eventual goal is to show the opposite inclusion, thus proving:

Theorem 1 (PCP Theorem) $\mathbf{PCP}[\log(n), 1] = \mathbf{NP}$.

1 Equivalent Formulations of the PCP Theorem

1.1 Introduction to Constraint Satisfaction Problems

CSPs (Constraint Satisfaction Problems) are a category of optimization problems; we encounter CSPs throughout this lecture, as well as later in the course. An instance of a CSP is $\phi = \{C_1, \dots, C_m\}$, where each C_i is a constraint defined on a subset of size q of the B -ary variables $V = \{x_1, \dots, x_n\}$. Formally, each C_i is specified by a function $f_i : \{0, 1\}^q \rightarrow \{0, 1\}$ and a q -tuple $(i_1, \dots, i_q) \in [n]^q$ (the indices of the variables on which the constraint is defined); C_i is satisfied by an assignment A if and only if $f_i(A(x_{i_1}), \dots, A(x_{i_q})) = 1$.

Max-CSP is the problem of finding an assignment which maximizes the number of satisfied constraints in a given CSP. The corresponding decision problem is whether all constraints can be satisfied. An important example of Max-CSP is Max-3SAT: given m clauses, each a disjunction of three literals, find an assignment which satisfies the maximal number of clauses. In this case $B = 2$, $q = 3$.

Denote by $\mathbf{CSP}(\mathcal{F})$ the problem of satisfying CSPs where the functions are members of \mathcal{F} . Returning to our previous example, if \mathcal{F} is the family of functions over three binary variables which return a disjunction of the variables or their negations, then $\mathbf{CSP}(\mathcal{F}) = \text{Max-3SAT}$.

We now introduce a surprising theorem:

Fact 2 (Schaeffer's Dichotomy Theorem) [3] *For any family \mathcal{F} over binary variables, $\mathbf{CSP}(\mathcal{F})$ is either in \mathbf{P} or is \mathbf{NP} -Complete.*

Note that this theorem was strengthened to deal with ternary variables, but is not known for general constant size variables. However, if $\mathbf{P} \neq \mathbf{NP}$, it is not likely to be true for very large (non-constant) variables.

1.2 Proving an Equivalence

Armed with knowledge of CSPs, we are ready to tackle the proof of the main lemma of this lecture. (**Irit: This is repeating the proof from the previous lecture**).

Lemma 3 *The following conditions are equivalent:*

1. $PCP[\log(n), 1] = NP$.
2. *There exists $\alpha < 1$ such that, given a 3CNF formula χ , it is NP-hard to decide between:*
 - χ is satisfiable.
 - Any assignment satisfies at most α fraction of the clauses of χ .

Proof (1 \Rightarrow 2) Since $SAT \in NP$, by the assumption there exists a $(\log(n), 1)$ -restricted verifier V , such that given a formula ϕ :

- $\phi \in SAT \Rightarrow \exists \pi$ s.t. $\Pr_R[V^\pi(\phi, R) = YES] = 1$
- $\phi \notin SAT \Rightarrow \forall \pi : \Pr_R[V^\pi(\phi, R) = YES] < \frac{1}{2}$

Denote the length of the proof by m , and let x_1, \dots, x_m be the bits in the proof. For every choice R of $r = \log(n)$ random bits (and a fixed input ϕ), V queries q addresses in the proof, $i_1^R, \dots, i_q^R \in [m]$; these q addresses are deterministically selected according to R . V now accepts if and only if certain conditions are met by the q queried bits. In other words, V answers according to a boolean function ψ_R over $\{0, 1\}^q$. There are $2^r = 2^{\log(n)} = n$ such functions ψ_R .

Every ψ_R , together with a choice of i_1^R, \dots, i_q^R , may be regarded as a constraint. We now have a CSP whose variables are x_1, \dots, x_m , and whose constraints C_R are specified by ψ_R and (i_1^R, \dots, i_q^R) .

If $\phi \in SAT$, by the completeness condition there is a proof π that convinces V ; formally, there is an assignment $A : \{x_1, \dots, x_m\} \rightarrow \{0, 1\}$, corresponding to the bits in π , which satisfies all the constraints. However, if $\phi \notin SAT$, then by the soundness condition for any assignment at most $\frac{1}{2}$ of the constraints are satisfied.

The last step is to transform the above CSP into a 3CNF formula, while maintaining a bounded probability of error. Specifically, we wish to transform the set of functions $\{\psi_R\}_R$ to 3CNF. Transforming such a function to a q -CNF formula is easy: Simply take as clauses all of the disjunctions of the complements of literals y_1, \dots, y_q which satisfy $\psi_R(y_1, \dots, y_q) = 0$. Transforming a q -CNF to a 3CNF is also easy. For example, the 4CNF formula $x \vee y \vee z \vee w$ can be transformed to an equivalent 3CNF formula by adding the variable A : $(A \vee x \vee y) \wedge (\neg A \vee z \vee w)$. There exists $K = K(q) \in \mathbb{N}$, such that every ψ_R can be transformed into K 3CNF clauses.

If an assignment satisfies a constraint, it can be extended to satisfy the set of 3CNF clauses corresponding to that constraint. We conclude that any assignment which satisfies all the constraints gives an assignment which satisfies the new 3CNF. On the other hand, if an assignment did not satisfy a constraint, at least one of the K new clauses corresponding to this constraint cannot be satisfied by an extension of the assignment; thus, if previously $\frac{1}{2}$ fraction of the constraints were left unsatisfied, now at least $\frac{1}{2K}$ of the clauses are unsatisfied.

We have shown a polynomial-time (remember there are only n ψ_R s) reduction from SAT to gap-3SAT. Since SAT is NP-hard, this shows 2.

(2 \Rightarrow 1) It is sufficient to show that $\mathbf{NP} \subset \mathbf{PCP}[\log(n), 1]$, and in particular that SAT has $(\log(n), 1)$ -restricted verifier. From the assumption there is a polynomial-time reduction from SAT to $\text{gap-3SAT}(\alpha, 1)$ which has an input ϕ and an output ψ over variables x_1, \dots, x_m , such that:

- $\phi \in \text{SAT} \Rightarrow \psi$ is satisfiable.
- $\phi \notin \text{SAT} \Rightarrow$ Any assignment satisfies at most α fraction of the clauses of ψ .

The verifier transforms the the input ϕ into ψ using the given reduction; the proof is an assignment to the variables x_1, \dots, x_m . The verifier uses the $\log(n)$ random bits to choose one of the clauses of ψ (there are at most $O(n^k)$ such clauses since the reduction is polynomial), and accepts if and only if the chosen clause is satisfied by the assignment. If $\phi \in \text{SAT}$, there is an assignment that satisfies all clauses of ψ , and therefore the chosen clause will be satisfied. If $\phi \notin \text{SAT}$, any assignment satisfies at most α fraction of the clauses in ψ ; thus for any assignment, the probability that a random clause is satisfied is at most α . We will later show how to decrease the probability of error. ■

2 PCP Reductions

2.1 Introduction to PCP reductions

From lemma 2 we have that in order to prove the PCP Theorem, it is sufficient to show a polynomial reduction from CSP to gap-CSP ; the reduction transforms a CSP $\phi = \{C_1, \dots, C_m\}$ over variables x_1, \dots, x_d into $\phi' = \{C'_1, \dots, C'_R\}$ over variables $y_1, \dots, y_{d'}$.

For any CSP χ , denote by $\mathbf{OPT}(\chi)$ the maximal number of satisfiable constraints in χ . We require the following conditions to hold:

- $\mathbf{OPT}(\phi) = 1 \Rightarrow \mathbf{OPT}(\phi') = 1$
- $\mathbf{OPT}(\phi) < 1 \Rightarrow \mathbf{OPT}(\phi') < \epsilon$

The parameters of the reduction are:

- ϵ - The probability of error.
- R - The number of constraints in ϕ' .
- s - The size of the constraints (e.g. the circuit size) of the constraints in ϕ' .
- q - The number of variables over which each C'_i is defined.
- Σ - The alphabet of the variables \bar{y} .

We wish to point out that, although it is not written explicitly, these parameters are functions of n .

A reduction that satisfies the above conditions and has the above parameters is known as a **PCP Reduction**. We want to find a reduction with $\epsilon < 1$, constant q, s, Σ , as well as $R = \text{poly}(n)$.

2.2 Sequential Repetition and Pseudorandomness Sequential Repetition

In order to understand the significance of the parameters of PCP reductions, we shall study trade-offs between different parameters. We first present increasingly sophisticated methods of decreasing the probability of error of a PCP reduction, at the price of increasing other parameters.

Lemma 4 *Given a PCP Reduction with parameters $\epsilon, R, q, s, \Sigma$, there exists a PCP Reduction with parameters $\epsilon^t, R^t, tq, ts, \Sigma$ for all $t \in \mathbb{N}$.*

Proof For $t = 2$, we build a new CSP ϕ'' with constraints

$$C''_{ij} = C'_i \wedge C'_j$$

for all $i, j \in [R]$. We have:

- $OPT(\phi) = 1 \Rightarrow OPT(\phi') = 1 \Rightarrow OPT(\phi'') = 1$.
- $OPT(\phi) < 1 \Rightarrow OPT(\phi') < \epsilon \Rightarrow OPT(\phi'') < \epsilon^2$.

In order to get the result for a general t , we define ϕ'' with constraints

$$C''_{i_1, \dots, i_t} = C'_{i_1} \wedge \dots \wedge C'_{i_t}$$

for all $i_1, \dots, i_t \in [R]$. ■

In order to maintain a low probability of error while preventing an explosion in the number of constraints in ϕ' , we will need some notions from Pseudorandomness Theory. The following definition will later allow us to decrease the probability of error from $1 - \mu$ to β , for certain values of μ and β .

Definition 5 (Hitting Set) *A set $\mathcal{F} \subset [N]^k$ is (β, μ) -**hitting** if $k = O(\frac{\log(\frac{1}{\beta})}{\mu})$, and every subset $S \subset [N]$ of size at least μN intersects with at least $(1 - \beta)$ fraction of the members of \mathcal{F} .*

Fact 6 [2] *There exists a (β, μ) -hitting set of size $N \text{poly}(\frac{1}{\beta})$, which can be constructed in polynomial time (it can be obtained via a random walk on an expander).*

Corollary 7 *Given a PCP Reduction with parameters $\epsilon = O(1)$, $R = \text{poly}(n)$, $q = O(1)$, $s = O(1)$, Σ , there exists a PCP Reduction with parameters $\epsilon' = \frac{1}{\text{poly}(n)}$, $R' = \text{poly}(n)$, $q' = O(\log(n))$, $s' = O(\log(n))$, Σ .*

Proof Idea Let $\mathcal{F} \subset [R]^{\log(n)}$ be a hitting set for $[R]$ with $\mu = 1 - \epsilon$, $\beta = n^{-c(1-\epsilon)}$ (for a constant $c \in \mathbb{R}$), of size $R \text{poly}(n^{c(1-\epsilon)}) = \text{poly}(n)$. We build a new CSP ϕ'' with constraints

$$C''_F = C'_{f_1} \wedge \dots \wedge C'_{f_{\log(n)}}$$

for all $F = \{f_1, \dots, f_{\log(n)}\} \in \mathcal{F}$. Clearly we have $R' = |\mathcal{F}| = \text{poly}(n)$, $q' = O(\log(n))$, $s' = O(\log(n))$. Assume $OPT(\phi) < 1$, fix an assignment to the variables of ϕ' , and let S be the set of unsatisfied constraints in ϕ' under the assignment; we are guaranteed that $|S| \geq (1 - \epsilon)R$. At most

$$\beta = n^{-c(1-\epsilon)} = \frac{1}{\text{poly}(n)}$$

fraction of the C''_f do not contain clauses from S , and therefore only this fraction of the clauses is satisfied by the assignment. ■

2.3 Constraints Over a Larger Alphabet

(Irit: I rewrote some parts below)

PCP reductions can also be defined over variables whose values are taken from a larger (non-binary) alphabet Σ . A constraint that reads q such variables, potentially sees $|\Sigma|^q$ different inputs, or $q \cdot \log |\Sigma|$ bits of information.

It is important to observe that constraints reading $q \log |\Sigma|$ binary variables are (at least seemingly) more expressive than constraints that read q Σ -ary variables. In particular, the later can easily be simulated by the former (replace each variable with $\log |\Sigma|$ variables that encode its binary representation); but not necessarily vice versa.

Indeed, corollary 7 yields a PCP reduction with parameters $q = O(\log n)$ and $\Sigma = \{0, 1\}$ and polynomially small error-probability. If we set q to be constant and $\log |\Sigma| = O(\log n)$ while maintaining the same (polynomially small) error probability – *it is an open question whether there exists a PCP reduction with such parameters* . In both settings of q, Σ the constraints read the same number of $O(\log n)$ bits.

In fact, this open question is the extreme end of a sliding scale of parameters. First let us ask how small can the error-probability possibly be:

Proposition 8 *Under the assumption that $\mathbf{P} \neq \mathbf{NP}$, in any PCP reduction the probability of error is at least $\frac{1}{|\Sigma|^q}$.*

Proof As mentioned before, a constraint can be regarded as a function $C : \Sigma^q \rightarrow \{0, 1\}$. We can assume without loss of generality that the reduction does not generate any unsatisfiable constraints, because in this case it can simply terminate instead of generating additional constraints¹. Thus, every constraint has at least one assignment which satisfies it, out of the $|\Sigma|^q$ possible assignments. If we choose a value for each variable i.i.d from the uniform distribution over Σ , the probability that C is satisfied is at least $\frac{1}{|\Sigma|^q}$. By the linearity of expectation, the expected number of constraints which are satisfied by this random assignment is at least $\frac{R}{|\Sigma|^q}$. We conclude by noting that there exists an assignment which achieves the expectation, and therefore for any ϕ' , $OPT(\phi') \geq \frac{1}{|\Sigma|^q}$. ■

The following “**Sliding Scale Conjecture**” was proposed in [1] and basically states that the error probability meets the lower bound² of Proposition 8: There exists a constant c , such that for all $1 < |\Sigma| < n$ there exists a PCP reduction with $q = O(1)$ and $\epsilon = \frac{1}{|\Sigma|^c}$.

We mention that this conjecture has been proven for a partial range of the parameters, namely the conjecture is true for $1 < |\Sigma| < 2^{(\log(n))^{1-\delta}}$ for all $\delta > 0$.

References

- [1] Bellare, M., Goldwasser, S., Lund, C., and Russel A. *Efficient Probabilistically Checkable Proofs and Applications to Approximation*. Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing (STOC), pages 294-304, 1993.

¹The assumption that $\mathbf{P} \neq \mathbf{NP}$ guarantees that the (polynomial) PCP reduction doesn't always terminate when applied to unsatisfiable CSPs: otherwise we would get a contradiction to the \mathbf{NP} -completeness of CSP.

²It is stated only for the (most interesting) case of $q = O(1)$.

- [2] Goldreich, O. *A Sample of Samplers - a Computational Perspective on Sampling (survey)*. Electronic Colloquium on Computational Complexity (ECCC), 4(020), 1997.
- [3] Schaeffer, T.J. *The Complexity of Satisfiability Problems*. Proceedings of the Tenth Annual ACM Symposium on Theory of Computing (STOC), pages 216-226, 1978.