MAX-k-SAT

- Missing link: first gap-producing reduction
  - history’s guide
    It should have something to do with SAT
- Definition: MAX-k-SAT with gap $\varepsilon$
  - instance: k-CNF $\phi$
    - YES: some assignment satisfies all clauses
    - NO: no assignment satisfies more than $(1 - \varepsilon)$ fraction of clauses

Proof systems viewpoint

- MAX-k-SAT with gap $\varepsilon$ NP-hard $\Rightarrow$ for any language $L \in$ NP proof system of form:
  - given $x$, compute reduction to MAX-k-SAT: $\phi_x$
    - expected proof is satisfying assignment for $\phi_x$
    - verifier picks random clause (“local test”) and checks that it is satisfied by the assignment
      $x \in L \Rightarrow \Pr[\text{verifier accepts}] = 1$
      $x \notin L \Rightarrow \Pr[\text{verifier accepts}] \leq (1 - \varepsilon)$
    - can repeat $O(1/\varepsilon)$ times for error $< \frac{1}{2}$

PCP

- Probabilistically Checkable Proof (PCP) permits novel way of verifying proof:
  - pick random local test
  - query proof in specified k locations
  - accept iff passes test

- fancy name for a NP-hardness reduction

PCP

- PCP[r(n),q(n)]: set of languages $L$ with p.p.t. verifier $V$ that has $(r, q)$-restricted access to a string “proof”
  - $V$ tosses $O(r(n))$ coins
  - $V$ accesses proof in $O(q(n))$ locations
  - (completeness) $x \in L \Rightarrow \exists$ proof such that
    $\Pr[V(x, \text{proof}) \text{ accepts}] = 1$
  - (soundness) $x \notin L \Rightarrow \forall$ proof*
    $\Pr[V(x, \text{proof}*) \text{ accepts}] \leq \frac{1}{2}$
PCP

- Two observations:
  - $\text{PCP}[1, \text{poly } n] = \text{NP}$ proof?
  - $\text{PCP}[\log n, 1] \subseteq \text{NP}$ proof?

**The PCP Theorem** (AS, ALMSS):

$\text{PCP}[\log n, 1] = \text{NP}$.

---

**Corollary**: MAX-$k$-SAT is NP-hard to approximate to within some constant $\epsilon$.

- using PCP[log $n$, 1] protocol for, say, VC
- enumerate all $2^{O(\log n)} = \text{poly}(n)$ sets of queries
- construct a $k$-CNF $\phi_i$ for verifier’s test on each
  - note: $k$-CNF since function on only $k$ bits
- "YES" VC instance $\Rightarrow$ all clauses satisfiable
- "NO" VC instance $\Rightarrow$ every assignment fails to satisfy at least $\frac{1}{2}$ of the $\phi_i$ $\Rightarrow$ fails to satisfy an $\epsilon = (\frac{1}{2})^{2^k}$ fraction of clauses.

---

**The PCP Theorem**

- Elements of proof:
  - arithmetization of 3-SAT
    - we will do this
  - low-degree test
    - we will state but not prove this
  - self-correction of low-degree polynomials
    - we will state but not prove this
  - proof composition
    - we will describe the idea

---

**Proof Composition (idea)**

1. composition of verifiers:
   - reformulate "outer" so that it uses $O(\log n)$ random bits to make 1 query to each of 3 provers
   - replies $r_1, r_2, r_3$ have length polylog $n$
   - Key: accept/reject decision computable from $r_1, r_2, r_3$ by small circuit $C$

2. composition of verifiers (continued):
   - final proof contains proof that $C(r_1, r_2, r_3) = 1$ for inner verifier’s use
   - use inner verifier to verify that $C(r_1, r_2, r_3) = 1$
   - $O(\log n) \cdot \text{polylog } n$ randomness
   - $O(1)$ queries
   - tricky issue: consistency
Proof Composition (idea)

- \( \text{NP} \subseteq \text{PCP}[^{\log n}, 1] \) comes from
  - repeated composition
  - \( \text{PCP}[^{\log n}, \text{polylog} n] \) with \( \text{PCP}[^{\log n}, \text{polylog} n] \) yields \( \text{PCP}[^{\log n}, \text{polylog} n] \)
  - \( \text{PCP}[^{\log n}, \text{polylog} n] \) with \( \text{PCP}[^{\log n}, 1] \) yields \( \text{PCP}[^{\log n}, 1] \)
- details omitted…

The inner verifier

**Theorem:** \( \text{NP} \subseteq \text{PCP}[n^2, 1] \)

**Proof (first steps):**
1. Quadratic Equations is NP-hard
2. PCP for QE:
   - proof = all quadratic functions of a soln. \( x \)
   - verification = check that a random linear combination of equations is satisfied by \( x \)
     (if prover keeps promise to supply all quadratic fns of \( x \))

Quadratic Equations

**Lemma:** QE is NP-complete.

**Proof:** clearly in NP; reduce from CIRCUIT SAT
- circuit \( C \) an instance of CIRCUIT SAT
- QE variables = variables + gate variables

Quadratic Functions Code

- intended proof:
  - \( F \) the field with 2 elements
  - given \( x \in F_n \), a solution to instance of QE
  - \( f_i : F^n \rightarrow F_2 \) all linear functions of \( x \)
  - \( f_i = \sum_i a_i x_i \)
  - \( g_i : F_n \times F_n \rightarrow F_2 \) includes all quadratic fns of \( x \)
    - KEY: can evaluate any quadratic function of \( x \) with a single evaluation of \( f_i \) and \( g_i \)

PCP for QE

- quadratic equation over \( F_2 \):
  \[ \sum_{i \leq j} a_{ij} x_i x_j + \sum b_i x_i + c = 0 \]
- language QUADRATIC EQUATIONS (QE) = \{ systems of quadratic equations over \( F_2 \) that have a solution (assignment to the X variables) \}
Theorem: NP \subseteq PCP[\log n, \text{polylog } n]

Proof (first steps):
- define: Polynomial Constraint Satisfaction (PCS) problem
- prove: PCS gap problem is NP-hard

To "enforce promise", verifier needs to perform:
- linearity test: verify \( f, g \) are (close to) linear
- self-correction: access the linear \( f', g' \) that are close to \( f, g \)
[so \( f = \text{Had}(u) \) and \( g' = \text{Had}(V) \)]
- consistency check: verify \( V = u \otimes u \)

\begin{align*}
\text{PCP for QE} \\
x \in F^n \; \text{s.t.} \\
f(a) = \sum a x \quad & \text{Had}(x) \\
g(A) = \sum A[i,j]x_i x_j \quad & \text{Had}(x \otimes x)
\end{align*}

\begin{align*}
\text{PCP for QE} \\
x \in F^n \; \text{s.t.} \\
f(a) = \sum a x \quad & \text{Had}(x) \\
g(A) = \sum A[i,j]x_i x_j \quad & \text{Had}(x \otimes x)
\end{align*}

\begin{align*}
\text{PCP for QE} \\
x \in F^n \; \text{s.t.} \\
f(a) = \sum a x \quad & \text{Had}(x) \\
g(A) = \sum A[i,j]x_i x_j \quad & \text{Had}(x \otimes x)
\end{align*}

\begin{align*}
\text{PCP for QE} \\
x \in F^n \; \text{s.t.} \\
f(a) = \sum a x \quad & \text{Had}(x) \\
g(A) = \sum A[i,j]x_i x_j \quad & \text{Had}(x \otimes x)
\end{align*}

\begin{align*}
\text{PCP for QE} \\
x \in F^n \; \text{s.t.} \\
f(a) = \sum a x \quad & \text{Had}(x) \\
g(A) = \sum A[i,j]x_i x_j \quad & \text{Had}(x \otimes x)
\end{align*}

\begin{align*}
\text{PCP for QE} \\
x \in F^n \; \text{s.t.} \\
f(a) = \sum a x \quad & \text{Had}(x) \\
g(A) = \sum A[i,j]x_i x_j \quad & \text{Had}(x \otimes x)
\end{align*}
**NP \subseteq \text{PCP}[^{\log n}, \text{polylog } n]**

- **MAX-\(k\)-SAT**
  - given: \(k\)-CNF \(\phi\)
  - output: max. \# of simultaneously satisfiable clauses
- **generalization: MAX-\(k\)-CSP**
  - given:
    - variables \(x_1, x_2, \ldots, x_n\) taking values from set \(S\)
    - \(k\)-ary constraints \(C_1, C_2, \ldots, C_t\)
  - output: max. \# of simultaneously satisfiable constraints

**NP \subseteq \text{PCP}[^{\log n}, \text{polylog } n]**

- algebraic version: MAX-\(k\)-PCS
  - given:
    - variables \(x_1, x_2, \ldots, x_n\) taking values from field \(F_q\)
    - \(n = q^m\) for some integer \(m\)
    - \(k\)-ary constraints \(C_1, C_2, \ldots, C_t\)
  - assignment viewed as \(f: (F_q)^m \rightarrow F_q\)
  - output: max. \# of constraints simultaneously satisfiable by an assignment that has deg. \(\leq d\)

**Lemma:** for every constant \(1 > \varepsilon > 0\), the MAX-\(k\)-PCS gap problem with
\[
t = \text{poly}(n) \quad \text{k-ary constraints with } k = \text{polylog}(n)
\]
field size \(q = \text{polylog}(n)\)
\[
n = q^m \quad \text{variables with } m = O(\log n / \log \log n)
\]
degree of assignments \(d = \text{polylog}(n)\)

- check: headed in right direction
  - \(O(\log n)\) random bits to pick a constraint
  - query assignment in \(O(\text{polylog}(n))\) locations to determine if it is satisfied
  - completeness \(1\); soundness \(1-\varepsilon\)
    (if prover keeps promise to supply degree \(d\) polynomial)

**NP \subseteq \text{PCP}[^{\log n}, \text{polylog } n]**

- **MAX-\(k\)-PCS gap problem:**
  - given:
    - variables \(x_1, x_2, \ldots, x_n\) taking values from field \(F_q\)
    - \(n = q^m\) for some integer \(m\)
    - \(k\)-ary constraints \(C_1, C_2, \ldots, C_t\)
  - assignment viewed as \(f: (F_q)^n \rightarrow F_q\)
  - **YES:** some degree \(d\) assignment satisfies all constraints
  - **NO:** no degree \(d\) assignment satisfies more than \((1-\varepsilon)\) fraction of constraints

**Proof of Lemma**
- reduce from 3-SAT
- 3-CNF \(\phi(x_1, x_2, \ldots, x_n)\)
- can encode as \(\psi: [n] \times [n] \times (0,1)^3 \rightarrow (0,1)\)
- \(\psi((i_1, b_1), (i_2, b_2), (i_3, b_3)) = 1\) iff \(\phi\) contains clause \((x_{i_1}^{b_1} \lor x_{i_2}^{b_2} \lor x_{i_3}^{b_3})\)
- e.g. \((x_5 \lor \neg x_5 \lor x_3) \Rightarrow \psi(3, 5, 2, 1, 0, 1) = 1\)
NP ⊆ PCP[log n, polylog n]
- pick \( H \subseteq F_q \) with \( \{0,1\} \subseteq H, |H| = \text{polylog } n \)
- pick \( m = O(\text{log nloglog } n) \) so \( |H|^m = n \)
- identify \( n \) with \( H_m \)
- \( \psi: H^m \times H^m \times H^m \times H^3 \to \{0,1\} \) encodes \( \varphi \)
- a satisfies \( \varphi \) iff \( \forall i_1, i_2, i_3, b_1, b_2, b_3 \)
  \[ \psi(i_1, i_2, i_3, b_1, b_2, b_3) = 0 \] or \( a(i_1) = b_1 \) or \( a(i_2) = b_2 \) or \( a(i_3) = b_3 \)
- extend \( \psi \) to a function \( \psi': (F_q)^{3m+3} \to F_q \) with degree at most \( |H| \) in each variable
- can extend any assignment \( a: H^m \to \{0,1\} \) to \( a': (F_q)^m \to F_q \) with degree \( |H| \) in each variable

\[ \psi': (F_q)^{3m+3} \to F_q \] encodes \( \varphi \)
\( a': (F_q)^m \to F_q \) s.a. iff \( \forall (i_1, i_2, i_3, b_1, b_2, b_3) \in H^{3m+3} \)
\[ \psi'(i_1, i_2, i_3, b_1, b_2, b_3) = 0 \] or \( a'(i_1) = b_1 \) or \( a'(i_2) = b_2 \) or \( a'(i_3) = b_3 \)
- define: \( p_a: (F_q)^{3m+3} \to F_q \) from \( a' \) as follows
  \[ p_a(i_1, i_2, i_3, b_1, b_2, b_3) = \psi'(i_1, i_2, i_3, b_1, b_2, b_3)(a'(i_1) - b_1)(a'(i_2) - b_2)(a'(i_3) - b_3) \] or \( \psi'(i_1, i_2, i_3, b_1, b_2, b_3) = 0 \)
- \( a' \) s.a. iff \( \forall (i_1, i_2, i_3, b_1, b_2, b_3) \in H^{3m+3} \)
  \[ p_a(i_1, i_2, i_3, b_1, b_2, b_3) = 0 \]