PCP - Lecture 7: Composition

Thus far, we have seen

\[ \text{NP} \leq_{\text{PCP}} \left( \Omega(\lg n), O(1) \right) \subset_{\text{poly}} \left( \Omega(1)^{\text{poly}} \right) \]

Our final goal is to replace \( \Omega(1)^{\text{poly}} \) by \( \text{polylog} \).

Today: We will show a method of reducing the alphabet size.

Here is the simplified idea. Recall that an \( (\epsilon, \delta) \)-verifier for SAT works as follows:

1. Read input \( \psi \)
2. Toss coins \( r \) and compute \( l_1, \ldots, l_\epsilon \) and a predicate \( \psi: \Sigma^k \rightarrow \{0,1\} \)
3. Accept if \( \psi(\Pi l_1, \Pi l_2, \ldots, \Pi l_\epsilon) = 1 \)

\[ \text{C: If } \psi \text{ SAT then } \exists r \text{ s.t. } \text{Prob}_r[\text{Ver}(\psi, r) \text{ acc}] = 1 \]
\[ \text{S: If } \psi \not\text{ SAT then } \forall r \text{ } \text{Prob}_r[\text{Ver}(\psi, r) \text{ acc}] < S \]

Suppose we have two verifiers:

1. \( V_1 \) is an \( (\epsilon_1, \delta_1) \)-restricted verifier for SAT
2. \( V_2 \) is an \( (\epsilon_2, \delta_2) \)-restricted verifier for SAT

The parameters are functions of the input size, \( n \).
Consider their composition: $V_{\text{comp}}$ works as follows:

(i) Run step 2 of $V_1$ on input $\overline{\Phi}$, let $\overline{\Psi}$ be the predicate instead of reading $\overline{\Pi}_1 ... \overline{\Pi}_k$ and computing $\overline{\psi}_r(...)$, let us transfer control to $V_2$ for this job.

(ii) Let $\overline{\psi}_r$ be a circuit computing $\overline{\psi}_r$

Run $V_2$ on input $\overline{\psi}_r$ (expecting an oracle proof for the satisfiability of $\overline{\psi}_r$)

expect a proof $\overline{\Pi}_r$ for each possible run of $V_2$

**Question:** $V_2$ is verifying that $\overline{\psi}_r$ is SAT but not necessarily by $\overline{\Pi}_1 ... \overline{\Pi}_k$. Is this OK? (later)

**Example:** Take $V_1 = V_2 = (\overline{\psi}_r, o(i))$, $\Sigma = \overline{\psi}_r \in \text{poly} y_n$

then $n' = (o(i) \cdot \text{poly} y_h) o(i) = \text{poly} y_n$

$q_{\text{comp}} = o(i)$

$\Sigma_{\text{comp}} = \sum_{\overline{\psi}_r \in \text{poly} (n')} = \overline{\psi}_r \in \text{poly} y_h (n)$

$r_{\text{comp}} = o(y_n) + o(y_{n'}) = o(y_n)$

so we achieved alphabet reduction! or have we?
We must check that $V_{comp}$ has completeness and soundness.

- It is easy to see that $V_{comp}$ enjoys completeness: if $\phi \in SAT$ then there is a proof $\Pi$ s.t. $V_{comp}$ accepts always. Simply use the honest proof for $V_1$ to see that each $\Phi_r$ is satisfiable, so there is an honest proof for each run of $V_2$.
  (we used the completeness of $V_1$ and of $V_2$)

- What about soundness?

Suppose $\Phi \notin SAT$. Then $\forall \Pi \forall r \forall \psi \{ V_{\Pi}(\phi, r) \text{ accepts} \} \leq s$.

However, for a random $r$, is $\Phi_r$ satisfiable or not?

It might be SAT for all $r$.

In fact, must be unless $\text{NP} \subseteq \text{DTIME}(2^{O(n\log n)})$.

Since we can go over all $r$ and check exhaustively if $\Phi_r$ is SAT.

Example: 3SAT: it is easy to satisfy each clause, the whole point is to do so via common assignment.

Summary so far: composition has potential of alphabet reduction, but so far does not seem to work!

Missing: A way to test that $\Phi$ is SAT by a given $q$ rather than just that $\Phi$ is SAT.
PCPs of Proximity or Assignment Testers

We need $V_\frac{1}{4}$ to be able to verify that $\Psi_r(\prod_{i=1}^n \prod_{j=1}^{p_i}) = 1$ without reading the entire "assignment" $\prod_{i=1}^n \prod_{j=1}^{p_i}$.

Given a circuit $\Phi$, and an assignment $\alpha$ for its vars test (with the possible assistance of a proof) that $\Phi(\alpha) = \text{true}$.

Note (Testing is approximate by nature)

If we only read a part of $\alpha$ we cannot expect to test if $\alpha$ is a satis. assign. or not. (Since a random bit flip of $\alpha$ will not be detected)

$\text{Def:}$ So we denote $\text{sat}(\Phi) = \{ \alpha | \alpha \text{ is a satis. assign. for } \Phi \}$

and also let $d(\alpha, \beta) = \text{Pr}(\alpha_i \neq \beta_i)$. If $d(\alpha, \beta) \geq \delta$ they are called $\delta$-far. If $\alpha$ is $\delta$-far from all strings $\beta \in B$ then $\alpha$ is $\delta$-far from $B$.

$\text{Def:}$ We say that SAT has a PCP of Proximity with proximity parameter $\delta$ if $\exists \ (r, q) - \text{restricted verifier of proximity for SAT}$

that receives two inputs: circuit $\Phi$ (explicitly) and a \coroaccess{Assignment $\alpha$} and then:

a) Reads $\Phi$, tosses coins $r$ computes $\Psi_r : \Sigma^q \rightarrow \Omega_{0,1}^r$ b) Reads $(\alpha_0(\text{not }), \ldots, \alpha_{r-1}(\text{not }))$, and accepts $\Psi_r$, iff $\text{satisfy } \Psi_r$. 


and such that the following holds:

\[ C: \text{If } a \text{ satisfies } \phi \text{ then } \exists \, \Pi \text{ s.t. } \Pr_b[\mathsf{Ver}^\phi_{\Pi}(\phi, a) \text{ acc}] = 1 \]

\[ S: \text{If } a \text{ is } \delta \text{-far from } \text{sat}(\phi) \text{ then } \forall \Pi \text{ } \Pr_b[\mathsf{Ver}^\phi_{\Pi}(\phi, a) \text{ acc}] \leq \delta \]

Recall: this is the set of assignments satisfying \( \phi \). If \( \phi \) not satisfy then it is empty and every \( a \) is \( \delta \)-far from it.

Such a verifier is also called an assignment tester.

We call the proof for such a verifier a PCP of Proximity.

The definition is more general (not only for SAT) but we don’t need it here for a “pair language”

At first sight — unclear if stronger/weaker than PCP.

Not weaker: Any \((r, q)\)-verifier of proximity can be made into an \((r, q)\)-verifier (by asking the prover to provide \(a\) as well).

Does this help composition?

We can now ask \( V_2 \) to test whether \( \Pi_{1q} \cdot \Pi_{1q} \) is close to an assignment satisfying \( \Pi_{1q} \).

Would work had \( V_2 \) complied with Robust Soundness:

Suppose the proof for \( V_2 \) is over binary alphabet, (always uncertain to grow)
Let us say that $\text{Ver}_\Psi^\Pi(\phi, r)$ $\delta$-accepts if the string $\Theta_1...\Theta_n$ is $\delta$-close to some string that satisfies $\forall r$.

\text{FRS: If } \phi \text{ SAT then } \forall \Pi \text{ Prob}[\text{Ver}_\Psi^\Pi(\phi, r) \text{ $\delta$-accepts}] \leq S.

\text{S: If } \neg \phi \text{ SAT then } \forall \Pi \text{ Prob}[\text{Ver}_\Psi^\Pi(\phi, r) \text{ accepts}] \leq S$

\textbf{Robustness Lemma: if } \exists \text{ PCP } \epsilon_S \left[ r, g \right] \Sigma \text{ then it has a PCP over binary alphabet with } \delta = 1/3g \quad \text{robust soundness. (c, s, r remain the same, } q = q_1 \Sigma \text{ is unimportant)}$

\textbf{Composition Theorem: Let } V_1 \text{ be an } (r_1, q_1) \text{ -restricted verifier for SAT with } \delta\text{-robust soundness. Let } V_2 \text{ be an } (r_2, q_2) \text{ -restricted verifier of proximity for SAT (with proximity parameter } \delta \text{). Then one can define a verifier } V_{\text{comp}} = V_1 \circ V_2 \text{ such that it is a }$

\left( r_1(n) + r_2(n'), q_2(n') \right) \Sigma_2(n') \text{ - restricted verifier, and}$

\text{ (1) if } V_1, V_2 \text{ have perfect completeness } \Rightarrow \text{ so does } V_{\text{comp}}

\text{ (2) if } s_1, s_2 \text{ are the soundness params of } V_1, V_2 \text{ then } s_{\text{comp}} = s_1 + s_2.$

\text{ Proof:}
Completeness: follows from \( C \) of \( V \).

This completes the description of \( V \).

The proof is over alphabet \( Z \).

Now it runs \( V \), using an auxiliary proof. Suppose \( \phi \), \( \psi \).

The appropriate \( \phi \) proof lifts of \( \phi \) to function as the assignment \( \phi \).

\( \psi \) sets \( \phi \) to be the explicit input and run recursively.

Booleans variables and computes the predicate \( \phi \).

\( \psi \) computes a circuit \( \phi \) which inputs \( \phi \).

Now it runs all \( \phi \), running \( \psi \).

The proof will consist of two parts: \( \phi \), and \( \psi \).

The idea is to run \( \psi \), but invoke \( \phi \) instead.
Soundness: Suppose $\Phi \notin \text{SAT}$. We claim that
\[
\forall \Pi \quad \Pr_{r \in (r_1, r_2)} \left[ V_{\text{comp}}(\phi, r) \text{ accepts} \right] \leq s_1 + s_2 - s_1 s_2
\]
Indeed $\forall \Pi, \forall \Pi_1 \quad \Pr_{r_1} \left[ V_{\text{comp}}(\phi, r_1) \text{ accepts} \right] \leq s_1$. So after selecting $r, V_{\text{comp}}$ may already accept $w$ w. prob $s_1$, or else:
Suppose $r$ is such that $V_2$ would have not $\delta$-accepted $\Pi_1$.
In other words, the corr $a$ is $\delta$-far from any assignment satisfying $\phi$. By the soundness of $V_2$ (recall it is a verifier of proximity w param $\delta$) it will accept with prob $\leq s_2$, no matter what proof $\Pi_1$ it sees.
Altogether $\Pr_{r, r_2} \left[ V_{\text{comp}}(\phi, r) \text{ accepts} \right] \leq$
\[
\Pr_{r_1} \left[ V_{\text{comp}}(\phi, r_1) \text{ accepts} \right] + \Pr_{r_2} \left[ V_2(\phi, r_2) \text{ accepts} \right] \text{ if } a \text{ is } \delta \text{-far from sat}(\phi)
\]
\[
\leq s_1 + (1 - s_1) \cdot s_2 = s_1 + s_2 - s_1 s_2
\]
params - check.

pf of robustization - ex or next week...