PCP - Lecture 7: Composition

Note Title 4/8/2008

Thus far, we have seen

$$NP \subseteq PCP_{1,\frac{1}{2}}[O(\lg n), O(1)]_{\{0,1\}^{poly\lg}}$$

Our final goal is to replace by $\{0,1\}$.

<u>Today:</u> We will show a method of reducing the alphabet size.

Here is the simplified idea. Recall that an $(r,q)_\Sigma$-verifier for SAT works as follows:

a) Read input Φ
toss coins r and compute $i_1 \dots i_q$ and a predicate $\varphi: \Sigma^q \to \{0,1\}$

b) Accept iff $\varphi(\Pi_{i_1}, \Pi_{i_2}, \dots, \Pi_{i_q}) = 1$

C: If $\Phi \in SAT$ then $\exists \Pi$ s.t. $Prob_r[Ver^{\Pi}(\phi, r)\ acc] = 1$

S: If $\Phi \notin SAT$ then $\forall \Pi$ $Prob_r[Ver^{\Pi}(\phi, r)\ acc] \leq S$

Suppose we have two verifiers:

1) V_1 is an $(r_1, q_1)_{\Sigma_1}$ - restricted verifier for SAT

2) V_2 is an $(r_2, q_2)_{\Sigma_2}$ - restricted verifier for SAT

the param are functions of the input size, n.

(one may think of $V_1 = V_2 = (o(\lg n), o(1))_{\{0,1\}^{\mathrm{polylg}\, n}}$ - verifier that we've seen.)

Consider their composition: V_{comp} works as follows

(i) Run step a) of V_1 on input Φ, let φ_r be the predicate instead of reading $\pi_{i_1} \ldots \pi_{i_q}$ and computing $\varphi_r(\cdots)$, let us transfer control to V_2 for this job.

(ii) Let $\tilde{\varphi}_r$ be a circuit computing φ_r

Run V_2 on input φ_r (expecting an oracle proof for the satisfiability of φ_r)

expect a proof π_r for each possible run of V_2

Question: V_2 is verifying that φ_r is SAT, but not necessarily by $\pi_{i_1} \ldots \pi_{i_q}$. Is this ok? (later)

Example: Take $V_1 = V_2 = (\lg n, o(1))_{\Sigma = \{0,1\}^{\mathrm{polylg}\, n}}$

then $n' = (o(1) \cdot \mathrm{polylg}\, n)^{o(1)} = \mathrm{polylg}\, n$

$q_{comp} = o(1)$

$\Sigma_{comp} = \Sigma_2 = \{0,1\}^{\mathrm{polylg}(n')} = \{0,1\}^{\mathrm{polylglg}(n)}$

$r_{comp} = o(\lg n) + o(\lg n') = o(\lg n)$

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 Π s.t. V_{comp}^{Π} accepts always
Simply use the honest proof for V_1, to see that each φ_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 $\forall \Pi \; \Pr_r\left[V_1^{\Pi}(\phi, r) \text{ accepts}\right] \leq s$

... hower, for a random r, is φ_r satisfiable or not?
might be SAT for $\underline{\underline{\text{all}}}$ r.
↗ in fact, must be unless $NP \subset DTIME(2^{q \log \Sigma + r})$
since we can go over all r and check exaustively if φ_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 "φ is sat by a given a"
rather than just that "φ is sat".

PCPs of Proximity or Assignment Testers

We need V_1 to be able to verify that $\varphi_r(\Pi_{i_1} \dots \Pi_{i_p}) = 1$ without reading the entire "assignment" $\Pi_1 \dots \Pi_q$.

Given a circuit Φ, and an assignment a for it's vars test (with the possible assistance of a proof) that $\Phi(a) = \text{true}$

Note (Testing is approximate by nature)
If we only read a part of a we cannot expect to test if a is a satis. assign. or not. (Since a random bit flip of a will not be detected)

Def: So we denote $sat(\Phi) = \{ a \mid a \text{ is a satis. assign. for } \phi \}$ and also let $d(a,b) = \Pr(a_i \neq b_i)$. If $d(a,b) \geq \delta$ they are called δ-far. If a is δ-far from all strings $b \in B$ then a is δ far from B.

Def: We say that SAT has a PCP of Proximity with proximity param δ if $\exists$ $(r,q)_\Sigma$-restricted verifier of proximity for SAT that receives two inputs: Circuit ϕ (explicitly) Assignment a (oracle access) and then:

a) Reads ϕ, tosses coins r, computes $i_1 \dots i_q$, $\varphi_r : \Sigma^q \to \{0,1\}$

b) Reads $(a \circ \Pi)_{i_1}$ $(a \circ \Pi)_{i_2}$ $\dots$ $(a \circ \Pi)_{i_q}$ and accepts iff satisfy φ_r.

and such that the following holds:

C: If a satisfies ϕ then $\exists \pi$ s.t. $\text{Prob}_r\left[\text{Ver}^{a,\pi}(\phi,r) \text{ acc}\right]=1$

S: If a is δ-far from $\underbrace{\text{sat}(\phi)}$ then $\forall \pi$ $\text{Prob}_r\left[\text{Ver}^{a,\pi}(\phi,r) \text{ acc}\right] \leq s$

recall: this is the set of assignments satisfying ϕ. If ϕ not satisf. then it is empty and <u>every</u> a is δ-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, for a "pair language") but we don't need it here.

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 $\underline{a}$ as well)

Does this help COMPOSITION ?

We can now ask V_2 to test whether $\pi_{i_1} \ldots \pi_{i_q}$ is <u>close to</u> an assignment satisfying φ_r.

Would work had $\underline{V_1}$ complied with Robust Soundness :

Suppose the proof for V_1 is over binary alphabet, (always true if allow q to grow)

Let us say that $Ver^{\pi}(\phi, r)$ δ-accepts if the string $\pi_{i_1} \ldots \pi_{i_q}$ is δ-close to some string that satisfies φ_r.

δ-RS: If $\phi \notin SAT$ then $\forall \pi$ $Prob_r[Ver^{\pi}(\phi, r)$ δ-accepts$] \leq s$.

compare with

S: If $\phi \notin SAT$ then $\forall \pi$ $Prob_r[Ver^{\pi}(\phi, r)$ accepts$] \leq s$

Robustness Lemma: if $L \in PCP_{c,s}[r, q]_{\Sigma}$ then it has a PCP over binary alphabet with $\delta = 1/3q$ – robust soundness. (c, s, r remain the same, $q' = q \lg \Sigma$ is unimportant)

Composition Theorem: Let V_1 be an (r_1, q_1) – rest. verifier for SAT with δ- robust soundness.
Let V_2 be an $(r_2, q_2)_{\Sigma_2}$ – restricted verifier of proximity for SAT (with proximity parameter δ). Then one can define a verifier $V_{comp} = V_1 \circ V_2$ such that it is a

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

c) if V_1, V_2 have perfect completeness $\Longrightarrow$ so does V_{comp}

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

Proof:

The idea is to run V_1, but transfer control to V_2 before actually accessing the proof.

The proof will consist of two parts: Π_1 and $\underbrace{\Pi_2 = \bigcup_{r_1} \pi_{r_1}}$

consisting of a proof per random string of V_1

Vcomp runs step a) of V_1 and has $r, i_1 \ldots i_{q_1}$ and $\varphi = \varphi_r : \Sigma_1^{q_1} \to \{0,1\}$

Now it wishes to run V_2.

Vcomp computes a circuit C which inputs q_1 Boolean variables and computes the predicate φ_r.

Vcomp sets C to be the explicit input and redirects the appropriate q_1 proof bits of Π_1 to function as the assignment a.

Now it runs V_2, using an auxiliary proof Π_r supposedly a PCPP for the fact that φ_r is satisfied by a.

(this proof is over alphabet Σ_2)

This completes the description of Vcomp.

We must now analyze completeness, soundness and params.

Completeness — follows from C of V_1 V_2.

<u>Soundness</u>: Suppose $\Phi \notin SAT$. We claim that

$$\forall \Pi \quad \operatorname*{Prob}_{r=(r_1,r_2)} \left[V_{comp}^{\Pi}(\phi, r) \text{ accepts} \right] \le s_1 + s_2 - s_1 s_2$$

Indeed $\forall \Pi_1$ $\operatorname*{Prob}_{r_1}\left(V_1^{\Pi_1}(\phi, r_1) \text{ acc} \right] \le s_1$. So after selecting

r_1 V_{comp} may already accept w. prob s_1, or else:
suppose r is such that V_1 would have not δ-accepted Π_1.
In other words, the corr a is δ-far from any assignment
satisfying φ_r. By the soundness of V_2 (recall it is a verifier
of proximity w param δ) it will accept with prob $\le s_2$.
no matter what proof Π_r it sees.

Altogether $\operatorname*{Prob}_{r=r_1 r_2}\left[V_{comp}^{\Pi}[\phi, r] \text{ accepts} \right] \le$

$$\operatorname*{Prob}_{r_1}\left[V_1^{\Pi_1}[\phi, r_1]\ \delta\text{accepts} \right] +$$

$$\Pr\left[V_1 \text{ does not } \delta\text{-accept} \right] \cdot \Pr_{r_2}\left(V_2^{a, \Pi_r}\left[\varphi_r, r_2 \right] \text{ accepts} \;\middle|\; a\ \delta\text{-far from } sat(\varphi_r) \right]$$

$$\le s_1 + (1 - s_1) \cdot s_2 = s_1 + s_2 - s_1 s_2.$$

params - check.

pf of robustization $\leftarrow$ ex or next week ...