1 2-query PCPs

Recall the definition of the complexity class $\mathbf{PCP}(r(n), q(n))$ from class.
Definition 1.
We say a language L is in the complexity class $\mathbf{PCP}(r(n), q(n))$ if there is a poly-time randomized verifier V such that for any $x \in \{0,1\}^*$, if we let n denote $|x|$ then

1. On input $\langle x, \pi\rangle$, V reads x, tosses $r(n)$ coins, reads $q(n)$ bits of π, then accepts or rejects.
2. Completeness: if $x \in L$, then there exists $\pi \in \{0,1\}^{\text{poly}(n)}$ such that $\alpha \stackrel{\text{def}}{=} \Pr(V(x,\pi) = 1) = 1$.
3. Soundness: if $x \notin L$, then for all $\pi \in \{0,1\}^{\text{poly}(n)}$, we have $\rho \stackrel{\text{def}}{=} \Pr(V(x,\pi) = 1) \le 1/2$.

In class we stated that $\mathbf{NP} = \mathbf{PCP}(O(\log n), q)$ for some universal constant $q > 0$.

Recall in class that we stated Håstad gave a 3-query PCP for SAT with completeness $\alpha = 1 - \varepsilon$ and soundness $\rho = 1/2 + \delta$ for any $\varepsilon, \delta \in (0,1)$. In his PCP, the alphabet was *binary*, i.e. the proof was a string $\pi \in \{0,1\}^{poly(n)}$. What if we sought 2-query PCPs with perfect completeness?

Exercise. *Show that* $\mathbf{P} = \mathbf{PCP}(O(\log n), 2)$.

Solution.
We show both $\mathbf{P} \subseteq \mathbf{PCP}(O(\log n), 2)$ and $\mathbf{PCP}(O(\log n), 2) \subseteq \mathbf{P}$.

$\mathbf{P} \subseteq \mathbf{PCP}(O(\log n), 2)$: Supposing $L \in P$, we give a desired proof system for L. The proof is simply the empty string. The verifier V flips 0 random bits and doesn't look at the proof, and simply decides whether $x \in L$ in polynomial time. The soundess is 0 and the completeness is 1.

$\mathbf{PCP}(O(\log n), 2) \subseteq \mathbf{P}$: Suppose $L \in \mathbf{PCP}(O(\log n, 2))$, with verifier V. The proof in this case is similar to Theorem 22.6 and Theorem 22.8 from Lecture Notes 22. We reiterate the important details here. First, perfect completeness and soundness $\rho \le 1/2$ implies that we can decide $x \in L$ via a ρ-gap2CSP instance. To remind the reader, whether V accepts or not is based on two queries to a supposed proof $\pi \in \{0,1\}^N$ for some $N \le poly(n)$. Thus for each random string $r \in \{0,1\}^R$ for $R = O(\log n)$, there is a function $V_{x,r} : \{0,1\}^N \to \{0,1\}$ such that $V_{x,r}(\pi) = 1$ iff V on input x and random coin flips r would accept the proof π. Note that $V_{x,r}$ only depends on 2 bits in π. Thus $V_{x,r}$ can be written as a 2-CNF formula $\varphi_{x,r}$ as per Theorem 22.8 of the lecture notes (and similarly to the proof that 3-SAT is $\mathbf{NP}$-hard). Then, we can create a 2-CNF formula

$$\varphi_x = \bigwedge_{r \in \{0,1\}^R} \varphi_{x,r}.$$

Note φ_x has polynomial size since $R = O(\log n)$. Then because of the completeness and soundness conditions, φ_x is satisfiable iff $x \in L$ (note we thus only need soundness $\rho < 1$ for this proof to work, not soundness $1/2$!). But deciding whether φ_x is satisfiable can be done in polynomial time, since $2\mathsf{SAT} \in \mathbf{P}$.

Despite the above exercise, we *can* get 2-query PCPs as long as we are willing to change the alphabet size. That is, rather than work with proofs $\pi \in \{0,1\}^{poly(n)}$, we work with proofs $\pi \in \Sigma^{poly(n)}$ for some $|\Sigma| > 2$. Then the verifier is only allowed to read q *symbols* in the proof π, as opposed to q bits.

Let us alter our **PCP** notation to include more information. We let $\mathbf{PCP}^{\Sigma}_{\alpha,\rho}(r(n), q(n))$ denote the class as defined above, but where the alphabet for π is Σ, the completeness is α, and the soundness is ρ.

Exercise. *For any constant q, show that $\mathbf{PCP}^{\Sigma}_{\alpha,1-\varepsilon}(r(n), q) \subseteq \mathbf{PCP}^{\Sigma^q}_{\alpha,1-\varepsilon/q}(r(n) + \log q, 2)$.*

Solution.
Suppose $L \in \mathbf{PCP}^{\Sigma}_{\alpha,1-\varepsilon}(r(n), q)$. Then there is some verifier V which flips $R = r(n)$ coins and does polynomial computation on x, then accepts iff some predicate $V_{x,r} : \Sigma^N \to \{0,1\}$ for some $N \leq poly(n)$ gives $V_{x,r}(\pi) = 1$, where $V_{x,r}$ depends on only q symbols of π.

We now construct a verifier V' to show $L \in \mathbf{PCP}^{\Sigma^q}_{\alpha,1-\varepsilon/q}(r(n) + \log q, 2)$. V' flips $r(n)$ bits as before, as well as an additional $\log_2 q$ bits to pick a random index $j \in \{1, \ldots, q\}$ (if q is not a power of 2 then round it up to a power of 2, then ignore the symbols read during the additional queries). V' then expects a proof of the form (π, π'), where $\pi \in \Sigma^N$ and $\pi' \in (\Sigma^q)^{N^q}$. π is expected to be a proof exactly as in the last paragraph, and π' is expected to be of the form $\pi'_{(i_1,\ldots,i_q)} = (\pi_{i_1}, \ldots, \pi_{i_q})$. V' then uses its random bitstring t of length $r(n)$ to pick $i_1, \ldots, i_q$ just as V would, then reads the symbol $\pi'_{(i_1,\ldots,i_q)} = (\sigma_1, \ldots, \sigma_q)$ (that's one query). It then also queries π_{i_j} (that's the second query). V' then accepts iff $V_{x,t}(\sigma_1, \ldots, \sigma_q) = 1$ *and* $\pi_{i_j} = \sigma_j$.

If $x \in L$, then a proof does exist to make V' accepts with probability α: namely, let π be the same proof that worked for V, and let π' be the proof with $\pi'_{(i_1,\ldots,i_q)} = (\pi_{i_1}, \ldots, \pi_{i_q})$.

If $x \notin L$, then consider any proof (π, π'). We know by assumption that for any π, V would reject π with probability at least ε (i.e. over its random choices of $i_1, \ldots, i_q$, V would reject $(\pi_{i_1}, \ldots, \pi_{i_q})$ with probability at least ε). When V' performs its query, its indices $i_1, \ldots, i_q$ are chosen according to the same probability distribution, and thus with probability at least ε, this choice would lead to proof probes which V would reject. Then there are two scenarios: (1) either $\pi'_{(i_1,\ldots,i_q)} = (\pi_{i_1}, \ldots, \pi_{i_q})$, or (2) they are not equal. In the first case, V' would reject. In the second case, it would reject with probability at least $1/q$, since we check consistency with π_{i_j} for a random j. Thus V' rejects with probability at least ε/q, as desired.

Note that the previous exercise, together with the PCP theorem, implies that for some constant q,

$$\mathbf{NP} \subseteq \mathbf{PCP}_{1,1/2}(O(\log n), q) \subseteq \mathbf{PCP}^{\{0,1\}^q}_{1,1-1/(2q)}(O(\log n), 2).$$

Raz's parallel repetition theorem allows us to decrease the soundness exponentially in t, by asking t questions in parallel. Specifically, Raz's parallel repetition theorem implies

$$\forall \rho \in (0,1),\ \exists c_\rho \in (0, \rho),\ \mathbf{PCP}^{\Sigma}_{1,\rho}(r(n), 2) \subseteq \mathbf{PCP}^{\Sigma^t}_{1,c_\rho^t}(t \cdot r(n), 2).$$

Thus by taking $t = O(\log(1/\varepsilon))$ we have altogether

$$\forall \varepsilon > 0, \exists \Sigma\ (|\Sigma| \leq poly(1/\varepsilon)) \text{ s.t. } \mathbf{NP} \subseteq \mathbf{PCP}^{\Sigma}_{1,\varepsilon}(O(\log n \cdot \log(1/\varepsilon)), 2).$$