1 PCPs and hardness of approximation

The following two questions are supposed to be easy sanity checks, making sure that you understand the basic definitions.

Question 1. *Consider a supposed PCP verifier V_{bad} for CLIQUE. Given a graph G with n vertices and a "PCP proof" π, the verifier V_{bad} reads the first $\lceil 3 \cdot \log(n) \rceil$ bits of π, interprets them as three vertices $i, j, k \in [n]$, and accepts if and only if i, j, k form a triangle in G. Does V_{bad} meet the definition of a PCP verifier for CLIQUE? Prove your answer.*

Question 2. *Without relying on the PCP theorem, prove that* 3SAT *has a PCP verifier satisfying the following. On an input formula Φ over n variables and with $m = O(n)$ clauses, the verifier uses $O(1)$ queries and $O(\log n)$ random bits, and:*

- *If Φ is satisfiable there is a proof π such that $\Pr_r[V^\pi(x,r) = 1] = 1$.*
- *If Φ is not satisfiable then for every proof π we have $\Pr_r[V^\pi(x,r) = 1] \leq 1 - 1/m$.*

Note that the error probability above (i.e., the probability of accidentally accepting proof for an incorrect claim) is quite bad. That is, with probability $1 - 1/m = 1 - o(1)$ we might accidentally accept. The technical challenge in the PCP theorem – whose proof we didn't learn – is to replace the error probability of $1 - 1/m$ with smaller error probability of $1/2$, without increasing the number of queries to be super-constant.[1]

The following question is still a sanity check, but it's more tricky. Recall that a PCP prover gets a proof, and only reads a small part of it. A crucial point, however, is that the proof is fixed in advance, before the verifier chooses which part of it to read.

In the next question you're asked to discover the reason for this. Specifically, you will show that there can never be a "PCP verifier" that first chooses the window $q_1, ..., q_k$ to read and only *afterwards* a proof is chosen. That is, if the proof is determined after the window is chosen, then a prover can cheat.

Question 3. *Consider the following variation on PCPs. The "verifier" gets input x and chooses randomness r, then sends queries $q,...,q_k \in [P]$ to a prover (where $P = P(n) = \text{poly}(n)$ is the "proof length"). The prover sees the queries and responds with $a_1, ..., a_k$. Finally, the verifier applies some fixed predicate $f : \{0,1\}^k \to \{0,1\}$ to the answers, e.g. takes the* OR.

Is it possible that for every $x \notin L$ and every prover *(including a malicious prover), such a verifier rejects with probability at least $1/2$?*

In class we stated that the $(1, 1-\epsilon)$-approximation for 3SAT is NP-hard under Karp reductions.[2] Using this theorem as a black-box, prove the following result:

[1] Recall from the tutorials that $1/2$ can be further reduced to any small constant (by re-running the PCP verifier for $O(1)$ times with independent coins and taking an AND), and the number of queries will still be a constant. In fact, the same is true for 0.99, so the real challenge in the PCP theorem is replacing $1 - 1/m$ with 0.99 without blowing up the number of queries. In contrast, repeating a verifier with error $1 - 1/m$ constantly many times does not significantly decrease the error probability (do you see why?).

[2] That is, for every $L \in \text{NP}$ there is a polytime machine $R = R_L$ (i.e., the Karp reduction) satisfying the following. If $x \in L$ then $R(x)$ is a satisfiable 3CNF formula, and if $x \notin L$ then $R(x)$ is a 3CNF formula such that every assignment to it satisfies at most a $(1-\epsilon)$-fraction of the clauses.

Question 4. *Prove that every $L \in \mathsf{NP}$ has a PCP verifier V that makes* **three queries** *to its PCP proof and for some $\epsilon > 0$ satisfies the following:*

- *If $x \in L$ there is π such that $\Pr_r[V^\pi(x,r) = 1] = 1$.*
- *If $x \notin L$ then for all π we have $\Pr_r[V^\pi(x,r) = 1] \leq 1 - \epsilon$.*

It is known (though not easy to show) that every $L \in \mathsf{NP}$ has a PCP verifier that, on input x and randomness r and access to a proof π, chooses a bit b and three locations $q_1, q_2, q_3 \in [|\pi|]$, and accepts if and only if $\pi_{q_1} \oplus \pi_{q_2} \oplus \pi_{q_3} = b$. The verifier accepts correct proofs with probability at least $1 - \epsilon$ over r, and accepts proofs for incorrect claims with probability at most $1/2 + \epsilon$ over r, where $\epsilon > 0$ is some tiny constant.[3]

Now consider the following problem. Given a system of linear equations, it is easy to decide if the system is satisfiable or not, by Gaussian elimination. However, when Gaussian elimination fails, we still don't know whether there is a solution satisfying 99% of the equations, or if any solution satisfies at most 51% of the equations. The next question asks to show that this approximation task is actually NP-hard.

Question 5. *In the $(1-\epsilon, 1/2+\epsilon)$-LIN problem the input is a system of m linear equations modulo 2 over variables $x_1, ..., x_n$. We need to accept if there is x satisfying at least $(1-\epsilon) \cdot m$ equations, and reject if every x satisfies at most $(1/2+\epsilon) \cdot m$ equations.[4] Using the PCP verifier mentioned above, show that this problem is NP-hard.*

The inapproximability result you showed above is tight, since all non-trivial linear systems modulo 2 have an assignment satisfying at least $1/2$ of the equations.[5]

[3] That is, if $x \in L$ then there is π such that $\Pr_r[V^\pi(x,r) = 1] \geq 1 - \epsilon$, and if $x \notin L$ then for every π we have $\Pr_r[V^\pi(x,r) = 1] \leq 1/2 + \epsilon$.

[4] As usual when studying promise problems, a valid algorithm may output any answers on input systems that are outside the promise (e.g., the best assignment satisfies $2/3$ of the equations).

[5] This can be proved by showing that, when choosing x uniformly at random, the expected number of satisfied equations is $m/2$ (using linearity of expectation). Thus, there exists x satisfying at least half of the equations (otherwise, the expectation you just calculated could not have been at least $m/2$).