Homework 33

Austin Frownfelter Matthew Bialecki

April 11, 2018

1 Problem 61

There are two steps to the proof that NP=L-PCP($\log n$).

First, L-PCP($\log n$) $\subseteq$ NP.

This is trivial. For a language in L-PCP($\log n$), we can construct an algorithm which instead of randomly checking with $\log n$ random bits, it nondeterministically selects the advice which causes it to accept.

Then, NP $\subseteq$ L-PCP($\log n$).

To do this, we pick a complete language in NP and show it is also in L-PCP($\log n$). Let's choose 3SAT. If L $\in$ 3SAT, then if $x \in$ L all clauses are satisfied, and if $x \notin$ L there exists a clause which is not satisfied.

$x \in L \rightarrow \exists y P[M(x, y) accepts] = 1$.

This is trivial. If $x \in L$, then no matter what clauses M selects, there is a satisfying assignment to those variables.

$x \notin L \rightarrow \forall y P[M(x, y) accepts] < 0.5$

Our verifier M takes book advice y, which is a string of variable assignments on x. We can construct the book such that its entry is a set of copies of the variable assignments, since M has only one-way access to y. To limit the tape size of M, y will be n copies of the variables. M can track which "run" it is currently on, then move to an offset which represents the specific clause in the equation, which is selected by $\log n$ random coin flips. In each test, it chooses a "permutation" of these coin flips, which is just a circulation of a single bit each time. Thus, M uses $2 \log n$ space.

In this log space, M is able to check a linear number of clauses. There is a potentially large probability of selecting a clause that would be true even though the formula is not. Over a linear number of checks, the probability M selects only satisfied clauses shrinks exponentially.