Lecture 23: Dec 1, 2003

Lecturer: Yuval Rabani *Scribe: Elliot Anshelevich*

1 Interactive Proof Systems

Recall from the last lecture the concepts of 2-prover single-round interactive proofs and of parallel repetition. In that context, we chose k random clauses $C_1, \ldots, C_k$ and the first prover gave an assignment for all variables in these clauses, which we denote by X. We also chose k random variables $x_1 \in C_1, \ldots, x_k \in C_k$ and the second prover returned an assignment for these variables, which we denote by Y. Now the outer verifier accepts if X and Y are consistent and rejects otherwise.

The parallel repetition theorem says that the error of a parallel repetition system decreases exponentially as k grows, so we may assume that for every input, the outer verifier either accepts with probability 1 or rejects with probability arbitrarily close to 0. We can guarantee this by taking a large enough k.

Now list all possibilities for the random inputs into the provers above. The total number of random bits needed is $O(k)$, so the number of possibilities is 2^k. Take X and Y returned for each random input by the above provers. The size of X is $3k$ bits, and the size of Y is k bits. Write both X and Y in the long code. Recall that a long code of a string X is a string of length $2^{2^{|X|}}$, with each position corresponding to a subset S of $\{0,1\}^{|X|}$. We will denote the long code of X by $lc(X)$, and the position corresponding to S in $lc(X)$ is 1 if and only if $X \in S$.

We will now form an inner verifier that will only check 3 bits of $lc(X)$ and $lc(Y)$ and will determine whether the outer verifier should accept on X and Y or not. It will work by generating some subsets S, and then asking questions like "Is $X \in S$?" about these subsets. We first give a simplification of an inner verifier, which will not quite work as we desire it, and later we correct the inadequacies. Specifically, our simplification will access the following bits of $lc(X)$ and $lc(Y)$:

b_1**:** Generate a random set S_2 of assignments to $x_1 \ldots x_k$. Ask prover 2 whether $Y \in S_2$ (equivalently, access the bit b_1 of $lc(Y)$ corresponding to S_2). There is a maximal set S_1 of possible values of X that are consistent with Y being in S_2.

b_2**:** Generate a random set R_1 of assignments to $C_1 \ldots C_k$. Ask prover 1 whether $X \in R_1$, and set that to be b_2.

b_3**:** Ask prover 1 whether $X \in R_1 \oplus S_1$, i.e. whether it is in exactly one of those sets. Set that to be b_3. We need b_2 and the $\oplus$ here so that prover 1 cannot prepare its answer to whether X is in S_1 in advance.

The inner verifier then accepts iff $b_1 \oplus b_2 \oplus b_3 = 0$. If X and Y are consistent, this will be true. There is an obvious problem with this approach, however. What if the provers always return 0? (i.e. always give the all-zero string as a witness) Then this verifier will always accept. We can fix this problem by using *folding*. For every pair of complement sets $S, \overline{S} \in \{0,1\}^{|X|}$, only one of the two bits corresponding to these sets is included in the long code. The other bit is implied. So, now we are dealing with strings of length $2^{2^k - 1}$. We can use some deterministic rule (such as lexicographical order of sets) to determine which bits to keep. Since R_1 is random, we know that about half of it is 1 and half is 0, so this fixes the all-zero-string problem.

There are still other problems with the above approach, however. The inputs for the inner verifier are not X and Y, but strings of length $2^{2^{3k}}$ and 2^{2^k}, which, according to the provers, represent the long codes of X and Y. For simplicity, assume that $X = Y$, with $|X| = k$, and X and Y are always consistent. There are 2^{2^k} possible long codes. But, only 2^k of them are valid, since there are only 2^k strings of length k. If we take $X_1, X_2, \ldots X_m \in \{0,1\}^k$, then strings of the form $lc(X_1) \oplus lc(X_2) \oplus \ldots \oplus lc(X_m)$ will be accepted as well, even though they do not correspond to the long codes of any assignment. In fact, the only strings that will be accepted are of the above form (for m odd).

We would like to accept only long codes of strings. That is too optimistic, however, so we will attempt to accept strings of the above form for only small m, i.e. strings which are XOR's of a small number of long codes. We do this as follows.

Let S be a random set of elements of $\{0,1\}^k$ of size $\epsilon 2^k$. Clearly, $Pr[X \in S] = \epsilon$. If $m > 1/\epsilon$, then $Pr[|\{X_1, \ldots, X_m\} \cap S| \text{ is odd}] \approx 1/2$. We modify the above inner verifier as follows. Pick such a set S. If $X \in S$, then we may not get the correct result, but the probability of this is small. To fix our test, we keep b_1 and b_2 the same, but change b_3 to be:

b_3**:** Generate a random set S as above. Ask prover 1 whether $X \in R_1 \oplus S_1 \oplus S$.

The inner verifier still accepts iff $b_1 \oplus b_2 \oplus b_3 = 0$. By making this change, we have sacrificed completeness, but gained soundness. More specifically, before this change all satisfiable formulas are accepted, but now we have the following claim for completeness:

Claim 1. If our 3CNF formula is satisfiable, then there is a proof that will convince the verifier with probability $\geq 1 - \epsilon$.

Suppose that for each $b_1 \oplus b_2 \oplus b_3$, we generate a 3LIN clause. The above claim actually shows that if a 3CNF formula is satisfiable, then the corresponding 3LIN formula is $\geq 1 - \epsilon$ satisfiable.

We now want to show soundness, i.e. that if a 3CNF formula is $< \delta$-satisfiable, then the verifier accepts any proof with probability $< 1/2 + \epsilon$. We will show that given a proof, if $Pr[\text{inner verifier accepts}] > 1/2 + \epsilon$, then $Pr[\text{outer verifier accepts}] > \delta$. Consider strings A and B which are given by the provers to the verifier and are supposed to represent the long codes of X and Y. The problem is that they might not actually be long codes of any particular string, and we want to accept them only if they are of the form $lc(X_1) \oplus lc(X_2) \oplus \ldots \oplus lc(X_m)$

for small m. We will think of A and B as written in the Fourier basis, which is an orthogonal basis of the 2^{2^k} dimensional vector space containing A and B consisting of elements of the form $lc(X_1) \oplus lc(X_2) \oplus \ldots \oplus lc(X_m)$, for every subset $\{X_1, \ldots, X_m\} \subset \{0,1\}^k$.

In the next lecture, we will show that $Pr[\text{inner verifier accepts } (A, B)] > 1/2 + \epsilon$ implies that only a small number of basis vectors have a non-negligible correlation with A and B. We know because of the addition of S to the inner verifier that each such basis vector is an XOR of a small number of long codes. From this, we get a list decoding of A and B: $X_1 \ldots X_m$ and $Y_1 \ldots Y_m$, where m is constant. One of the pairs X_i, Y_j must be consistent to cause the outer verifier to accept. So we can pick a pair at random. We now accept with probability $1/m$.