(67902) Topics in Theory and Complexity Jan 31, 2007

Lecture 14

Lecturer: Irit Dinur *Scribe: Dvir Falik*

1 Introduction

In this section we will complete the proof of alphabet reduction for constraint graphs.

Theorem 1 *There exists $q > 1$, $0 < \epsilon < 1$ such that for every constraint graph $G = (V, E, \Sigma, C)$ there exists a constant c_Σ such that a constraint system C' of q-ary constraints can be constructed that satisfies:*

- $|C'| \leq c_\Sigma \cdot (|V| + |E|)$
- **Soundness** $unsat(G) = 0 \Rightarrow unsat(C') = 0$
- **Completeness** $unsat(G) > 0 \Rightarrow unsat(C') \geq \epsilon \cdot unsat(G)$

Remark It is easy (and left as exercise) to translate such a constraint system C' to a constraint graph (binary constraints) with the desired parameters. therefore, Theorem **??** constitutes the alphabet-reduction for constraint graphs.

Here is a rough idea of the proof. We can replace each variable v with a set of Boolean variables $[v]$, and expect a proper encoding from an assignment to v to an assignment to $[v]$. Then, given an assignment A to the new variables we would like to be able to test:

- For every $v \in V$, whether the assignment to $[v]$ (roughly) encodes an assignment to v.
- For every edge $(u, v) \in E$, whether the assignment to $[v], [u]$ (roughly) encodes an assignment that satisfies the constraint on (v, u).

It is impossible to perform these tests by a single constraint in C' that reads only $q = O(1)$ Boolean variables. Therefore, the encoding (taking v to $[v]$) has to be *locally testable*, and each such constraint in G will be replaced by a number of new constraints in C'.

2 Local Testing Algorithms

Definition 2 (Local Testing Algorithm) *A property is a subset $P \subseteq \{0,1\}^n$. An algorithm A is called a* (q, ϵ) Local Testing Algorithm (LTA) *for P if for every $w \in \{0,1\}^n$, A randomly computes indices $i_1, \ldots, i_q \in [n]$ and a function $\Phi : \{0,1\}^q \to \{0,1\}$ and outputs $\Phi(w_{i_1} \ldots w_{i_q})$, satisfying:*

- *If $w \in P$ then* $\Pr[A^w = 0] = 0$.
- *If $w \notin P$ then* $\Pr[A^w = 0] \geq \epsilon \cdot dist(w, P)$.

Example Choose P to be the set of words in the Hadamard code,

$$P = \left\{ w \in \{0,1\}^{2^l} \;\middle|\; \exists a \in \{0,1\}^l, w(x) =< x, a > \right\}$$

and consider the following Algorithm A:

- Randomly pick $x, y \in \{0,1\}^l$.
- Output $w(x) \oplus w(y) = w(x \oplus y)$.

We have seen in the previous lecture that A is a $(3, \frac{1}{5})$-LTA for P.

We now state three lemmas (which are strengthenings of each other) which will give encodings to implement the idea of the proof of Theorem**??**.

Lemma 3 *There exists $L_1 \in \mathbb{N}$ and an encoding $H : \Sigma \to \{0,1\}^{L_1}$ such that the property $P = \{ H(a) \mid a \in \Sigma \}$ has an LTA T_1, and the encoding H has relative distance at least $1/3$.*

Lemma **??** has been proven in the previous lecture, taking H to be the Hadamard encoding, and the LTA T_1 is simply the linearity testing algorithm.

Lemma 4 *For every $\Phi : \Sigma \times \Sigma \to \{0,1\}$ there exist $L_2 \in \mathbb{N}$ and an encoding $E_\Phi : \Sigma \times \Sigma \to \{0,1\}^{L_2}$ such that the property $P_\Phi = \{ E_\Phi(a,b) \mid \Phi(a,b) = 1 \}$ has an LTA T_2, and the encodings H, E_Φ have relative distance at least $1/3$.*

We remark that Lemma **??** is stronger than Lemma **??**, in that it allows the property P to depend on an arbitrary predicate Φ. For the case $\Phi \equiv 1$ the claim is already proven in Lemma **??**.

Lemma **??** allows us to check that an assignment for a set of Boolean variables that encode a pair of Σ-variables is correct. However, this is useless if we cannot check that this assignment is also consistent with an assignment for the variables that encode a singe Σ-variable. This is taken care of by the following lemma, which is generalizes both previous lemmas.

Lemma 5 *For every $\Phi : \Sigma \times \Sigma \to \{0,1\}$ there exists $L_1, L_2 \in \mathbb{N}$ and encodings $H : \Sigma \to \{0,1\}^{L_1}$ and $E_\Phi : \Sigma \times \Sigma \to \{0,1\}^{L_2}$ such that the property*

$$P_\Phi = \{ (H(a), E_\Phi(a,b)) \mid a,b \in \Sigma, \Phi(a,b) = 1 \} \cup \{ (H(b), E_\Phi(a,b)) \mid a,b \in \Sigma, \Phi(a,b) = 1 \}$$

has an LTA T_3, and the encodings H, E_Φ have relative distance at least $1/3$.

3 Proof of Theorem ??

We are now ready to prove Theorem **??** assuming the correctness of the three lemmas above.
Proof We begin by describing the reduction, and then prove its correctness. Given a constraint graph G we construct the variables and constraints of C' as follows.

- **The Variables:** For every $v \in V$ define L_1 variables [v]. For every $e = (v_1, v_2) \in E$ define L_2 variables $[e]$. So
$$V' = \bigcup_{v \in V} [v] \cup \bigcup_{e \in E} [e].$$
Assume for simplicity that $L_1 = L_2$ (otherwise we can use a repetition of the encodings to obtain equality).

- **The Constraints:** Consider the following LTA T: given an assignment $A : V' \to \{0,1\}$
  1. Randomly choose $e = (v_1, v_2) \in E$. Test $A|_{[e]}$ using T_2. If the test fails, output 0. Otherwise,
  2. Randomly choose $v \in \{v_1, v_2\}$ and test $A|_{[v]}$ using T_1. If the test fails, output 0. Otherwise,
  3. Output the test of $A|_{[v] \cup [e]}$ using T_3.

  Each of the steps in T reads a constant number of variables, so T does as well. The number of random bits T uses is $r(T) = \log(|E|) + r(T_2) + 1 + r(T_1) + r(T_3) = \log(|E|) + s$ where s depends only on Σ and not on E.

  For every $\rho \in \{0,1\}^{r(T)}$ let c_ρ be the constraint that T computes when ρ are the random bits T uses. For each such ρ we will define a constraint in C':
$$C' = \left\{ \Phi_\rho \mid \rho \in \{0,1\}^{r(T)} \right\}$$

Clearly, $|V'| \leq max(L_1, L_2) \cdot (|V| + |E|) \leq c_\Sigma \cdot (|V| + |E|)$. Also, the size of C' is $|C'| = 2^{r(T)} = E \cdot 2^s = E \cdot c'_\Sigma$.

The constraints in C' simulate T, and therefore we have for every assignment A to V', $unsat_A(C') = Pr(T^A = 0)$.

We will now show this constraint system to be sound and complete:

- **Completeness** If $unsat(G) = 0$, then there exists an assignment a to V that satisfies all constraints in C. Encode a as A according to lemmas **??**, **??** and **??**, and these lemmas imply that $Pr(T^A = 0) = 0$.

- **Soundness** Assume $unsat(G) = \alpha > 0$. Let $A : V' \to \{0,1\}$ be the best possible assignment for V'. Define $a(v)$ to be the value in Σ whose encoding $H(a(v))$ is the closest to $A|_{[v]}$. By definition, $unsat_a(G) \geq \alpha$.

  Consider an edge $e = (v_1, v_2) \in E$, whose constraint Φ is not satisfied by a, and denote $w = A|_{[e]}$. We analyze the probability of failure of T conditioned on having chosen e in the first step.

  If $\Pr[T_2^w = 0] \geq \epsilon/6$ then T fails with probability at least $\epsilon/6$ which is a constant and we are done.

  If not, then $\Pr[T_2^w = 0] < \epsilon/6$. So there exist $\sigma_1, \sigma_2 \in \Sigma$ such that $\Phi(\sigma_1, \sigma_2) = 1$ and

  $$dist(w, P_\Phi) \leq \frac{\Pr[T_2^w = 0]}{\epsilon} \leq 1/6$$

  where ϵ is the soundness parameter of T_2. Since $\Phi(a(v_1), a(v_2)) = 0$ either $\sigma_1 \neq a(v_1)$ or $\sigma_2 \neq a(v_2)$. Assume wlog that $\sigma_1 \neq a(v_1)$, and note that in step 2 of T we chose $v = v_1$ with probability $1/2$.

  We claim that in this case the string $w' = A|_{[v]\cup[e]}$ has relative distance at least $\frac{1}{12}$ from the property tested by T_3. Here we use the assumption that $|[v]| = L_1 = L_2 = |[e]|$. Since $a(v_1) \neq \sigma_1$ it means that w must be changed in at least $L_1/6$ bits (since H has relative distance $1/3$) in order to make it an encoding of σ_1. But the length of w is half that of w' so we get a relative distance of $1/12$. Clearly this implies, $\Pr[T_3^{w'} = 0] \geq \epsilon \cdot \frac{1}{12}$.

  We have seen therefore that there is some constant $\epsilon' > 0$ such that $Pr(T^A = 0) \geq \epsilon' \cdot \alpha$.

■

4 Proof of Lemmas ?? and ??.

4.1 Lemma ??, Self Correction for the Hadamard Code, Quadratic Functions Encoding

(Throughout the proof, we assume w.l.o.g. that every element in $\Sigma \times \Sigma$ has a binary representation in l bits).

The proof of Lemma **??** relies on the fact that the Hadamard code is not only *locally testable*, but also *locally decodable*

Claim 6 *There exists a random algorithm $SelfCorr$ such that for every word $w \in \{0,1\}^{2^l}$ whose relative distance from the Hadamard code is at most $\epsilon > 0$, $SelfCorr$ reads two (randomly chosen) bits from w and computes $H(a)[x]$ such that*

$$\Pr[Selfcorr^w(x) = H(a)[x]] \geq 1 - 2 \cdot \epsilon.$$

Proof Here is a description of the algorithm: randomly choose $y \in \{0,1\}^l$ and return $w(y) \oplus w(x \oplus y)$. The probability that both bits were identical to their respective bits in $H(a)$ is greater than $(1 - 2 \cdot \epsilon)$. ■

How is this helpful? Suppose first that Φ were a linear function rather than a general predicate. In other words there is some $b \in \{0,1\}^l$ such that $\Phi(a) = \langle b, a \rangle$. Then, one can test the property

$$P_\Phi = \{\, H(a) \mid \Phi(a) = 1\}$$

by doing the following:

1. Run the LTA (from Lemma **??**) that tests whether w is a legal Hadamard codeword (linearity testing). If it fails, then output 'fail'. Otherwise,
2. Run $Self Corr^w(b)$ and accept iff it outputs 1.

If $w \in \{0,1\}^{2^l}$ is far from a Hadamard codeword, then step 1 will fail with constant probability. Otherwise, if w is close to a codeword $H(a)$ for which $\langle b, a \rangle \neq 1$ then step 2 will fail with constant probability.

In order to prove lemma **??** we need to generalize this idea to every Boolean function Φ (not only linear functions).

Proof of Lemma ?? We begin by describing the encoding E_Φ which is done in two steps.

1. **Step 1 – a circuit for Φ** We will consider C_Φ, the canonical boolean circuit that computes Φ. The size of this circuit depends only on Σ. Since Σ is constant, C_Φ is of constant size.

   Suppose X are the input variables of the circuit C_Φ. We will also add a variable for each of the internal edges in C_Φ, and let these variables by Y. Each gate in C_Φ can now be described by a quadratic equation in its input and output variables:

   If z_1, z_2 are input variables to some gate and z_3 is the output variable of the gate, then

   - A NOT gate can be described by the equation $z_1 + z_3 - 1 = 0$.
   - An AND gate can be described by the equation $z_3 - z_1 z_2 = 0$.
   - An OR gate can be described by the equation $z_3 + z_1 z_2 - z_2 - z_1 = 0$.

   Denote these equations by $\{f_i = 0\}_{i \in [m]}$ where m is the number of gates in C_Φ. Note that these are all *quadratic equations*. When the variables are assigned Boolean values, the f_i formulas are also Boolean. Given an assignment $a : X \to \{0,1\}$ there is a unique assignment $a' : X \cup Y \to \{0,1\}$ that agrees with a on X and such that all equations $\{f_i\}$ are satisfied.

2. **Step 2 – encoding using quadratic functions** Let $Z = X \cup Y$ and suppose $a : Z \to \{0,1\}$ is an assignment for Z. We will encode a as $E_\Phi(a) = H(a \otimes a)$.

All in all the encoding of an assignment $a : X \to \{0,1\}$ consists of extending a to an assignment $a' : X \cup Y \to \{0,1\}$ according to the computation of the gate (i.e. a' is the unique assignment that satisfies all equations f_i). Then, a' is encoded by $H(a' \otimes a')$. The encoding E_Φ takes $a \to a' \to H(a' \otimes a')$.

Notice that the set of linear functions on $z \otimes z$ is the set of quadratic functions on z (the linear part is also represented as $z_i^2 = z_i$).

Claim 7 *There are constants $q > 1$ and $\epsilon > 0$ and a (q, ϵ)-LTA for the property $P = \left\{ H(z \otimes z) \mid z \in \{0,1\}^k \right\}$.*

Proof Given w, denote $w_{diag} = \{\, w(x) \mid x_{ij} \leq \delta(i,j)\}$ to be the part of w that encodes only the linear part of $b \otimes b$.

- Randomly choose $a_1, a_2 \in \{0,1\}^k$.
- Check that $SelfCorr^{w_{diag}}(a_1) \cdot SelfCorr^{w_{diag}}(a_2) = SelfCorr^{w}(a_1 \otimes a_2)$

It is easy to see that when $w \in P$ the test succeeds. If w is δ far from P, then the third $SelfCorr$ application will give the wrong answer with probability $\Theta(\delta)$, causing the algorithm to fail with probability $\Theta(\delta)$.

The LTA T_2 will consist of two steps: Given an assignment w

- Check that w is a legal codeword in $H(z \otimes z)$.
- Randomly choose $\alpha \in \{0,1\}^n$ and denote $f_\alpha(z) = \sum_i \alpha_i f_i(z)$. If $SelfCorr^{w}(f_\alpha)$ is 1 return 0 (f_α is a quadratic function on z and therefore a linear function on $z \otimes z$).
- The value of Φ is one of the bits in z (the output of the last gate). Decode and return it using $SelfCorr$.

■

Notice that if z is not an encoding of a feasible computation in C_Φ, then $\{f_i(z)\}$ is a non-zero Boolean vector, and then it is known that $\Pr_\alpha(f_\alpha(z) = 1) = \frac{1}{2}$. Apart from that, the algorithm is a composition of LTA's, and therefore Local Testability is preserved. ■

4.2 Lemma ??

Proof of Lemma ?? We had $E_\Phi(a,b) = H(z \otimes z)$, when z includes the binary representation of (a,b). All that is needed is to locally test, using T_1, the part in w that encodes just (a,b). ■