Randomized Methods in Computation*
Lecture 8: Small Bias Sample Spaces

Notes taken by Boaz Barak and Itsik Mantin

Spring 2001

Summary: In the previous lecture we discussed the definition of small bias sample spaces and some properties of such spaces. In this lecture we prove that such sample spaces can be constructed in polynomial time, and demonstrate an application of this construction in proving a tight hardness of approximation result for an $\mathcal{NP}$-complete problem.

1 Introduction

In the previous lecture, we have seen the definition of ϵ-bias sample spaces. Let us recall this definition:

Definition 1 *Let $n \in \mathbb{N}$ and $\epsilon > 0$. A (multi) set $\Omega \subseteq \{0,1\}^n$ is called an* ϵ-bias sample space over $\{0,1\}^n$ *if for any non-empty $I \subseteq \{1,\ldots,n\}$*

$$\left| \Pr_{\mathbf{s}\in\Omega}[\bigoplus_{i\in I} s_i = 0] - \Pr_{\mathbf{s}\in\Omega}[\bigoplus_{i\in I} s_i = 1]\right| < \epsilon$$

where $s = s_1 \ldots s_n$.

An equivalent formulation is that for any non-empty $I \subseteq \{1,\ldots,n\}$ it holds that

$$\Pr_{\mathbf{s}\in\Omega}[\bigoplus_{i\in I} s_i = 0] \in \frac{1}{2} \pm \frac{\epsilon}{2}$$

(where by $a \pm c$ we mean the interval $(a-c, a+c)$)

Last lecture we noted that the uniform distribution has bias zero, and proved a relation between the bias of a distribution and its statistical difference from the uniform distribution.

This lecture we will see (in Section 2) that ϵ-bias sample spaces can be constructed in time polynomial in $\frac{n}{\epsilon}$. In particular this ϵ-bias sample spaces (over $\{0,1\}^n$) has size $poly(\frac{n}{\epsilon})$ (as opposed to the uniform sample space over $\{0,1\}^n$ that is of size 2^n).

In Section 3 we will use this construction to prove that a specific $\mathcal{NP}$-complete problem (Quadratic Equations) that can be approximated with ratio $\frac{1}{2}$ cannot be approximated with any better constant ratio, unless $\mathcal{P} = \mathcal{NP}$. This is one of the few tight hardness of approximation results that can be obtained without using the $\mathcal{PCP}$ Theorem.

* Lecture Notes for a course given by Oded Goldreich at the Weizmann Institute of Science, Israel.

2 Construction of a Small Bias Sample Space

In this section we will show that it is possible to construct an ϵ-bias sample space of size $poly(\frac{n}{\epsilon})$. That is, we prove the following Theorem:

Theorem 2 *There exists an algorithm that, on input n and ϵ outputs a list of n-bit strings that constitutes an ϵ-bias space and does so within time $poly(\frac{n}{\epsilon})$.*

2.1 On the Existence of $poly(\frac{n}{\epsilon})$-size ϵ-bias sample spaces.

Before proving Theorem 2 we note that the mere *existence* of an ϵ-bias sample space of $poly(\frac{n}{\epsilon})$ size can be easily proved using the probabilistic method. That is, we have the following:

Proposition 2.1 *For any $n \in \mathbb{N}$, $\epsilon > 0$, there exists an ϵ-bias sample space over $\{0,1\}^n$ of size $O(\frac{n}{\epsilon^2})$.*

Proof: Let Ω be a (multi) set containing strings in $\{0,1\}^n$. For any non-empty $I \subseteq \{1,\ldots,n\}$ we define $f_I(\Omega)$ to be the number of strings $\mathbf{s} = s_1 \ldots s_n \in \Omega$ such that $\bigoplus_{i\in I} s_i = 0$. By definition, it holds that Ω is an ϵ-bias sample space if (and only if) for any such I, it holds that $f_I(\Omega) \in (\frac{1}{2} \pm \frac{\epsilon}{2}) \cdot |\Omega|$.

Consider a random (multi) set Ω of size $\frac{n}{\epsilon^2}$. That is $\Omega = \{\mathbf{s}^{(1)},\ldots,\mathbf{s}^{(n/\epsilon^2)}\}$ where for each j, $\mathbf{s}^{(j)}$ is chosen uniformly and independently in $\{0,1\}^n$. Let us fix I to be a non-empty subset of $\{1,\ldots,n\}$, and let X_I be the random variable $f_I(\Omega)$. Another way to define X_I is as $\sum_{j=1}^{|\Omega|} Y_j$ where Y_j is a random variable that is defined to be 1 if $\oplus_{i\in I}\mathbf{s}_i^{(j)} = 0$ and 0 otherwise. The fact that the uniform distribution has bias zero implies that $\Pr[Y_j = 1] = \frac{1}{2}$, and so $\mu \stackrel{\text{def}}{=} Ex[X] = \frac{|\Omega|}{2} = \frac{n}{2\epsilon^2}$.

The random variables $Y_1,\ldots,Y_{k/\epsilon^2}$ are independent. Therefore, we can apply the Chernoff bound and get that (for every $I \neq \emptyset$)

$$\begin{aligned}
\Pr_{\Omega}[f_I(\Omega) \notin (\frac{1}{2} \pm \frac{\epsilon}{2}) \cdot |\Omega|] &= \Pr_{\Omega}[|X_I - \mu| > \epsilon\mu] \\
&= \Pr_{\Omega}[|\sum_{j=1}^{|\Omega|} Y_j - \mu| > \epsilon\mu] \\
&< 2e^{-2\epsilon^2\mu} \\
&= 2e^{-2\epsilon^2\cdot(n/2\epsilon^2)} \\
&= 2e^{-n}
\end{aligned}$$

For sufficiently large n, this probability is *strictly smaller* than 2^{-n}. As there are at most 2^n subsets of $\{1,\ldots,n\}$, we get that by the union bound there is a *positive probability* that for a random Ω of size $\frac{n}{\epsilon^2}$, for *any* non-empty set $I \subseteq \{1,\ldots,n\}$ the number $f_I(\Omega)$ is in $(\frac{1}{2} \pm \frac{1}{2}\epsilon)\cdot|\Omega|$. In particular there must *exist* at least one (multi) set Ω of size $\frac{n}{\epsilon^2}$ that satisfies the above condition for any non-empty $I \subseteq \{1,\ldots,n\}$. Such a set Ω will be an ϵ-bias sample space.

■

2.2 The Construction

We will prove Theorem 2 by showing a polynomial-time algorithm that outputs an ϵ-bias sample space Ω of size $O((\frac{n}{\epsilon})^2)$. This sample space will be larger than the sample space of Proposition 2.1 (which is $O(\frac{n}{\epsilon^2})$) but it will be explicitly constructible. We will start by showing the construction and then provide its analysis.

2.2.1 LFSR sequences

The primary tool we will use is a *Linear Feedback Shift Register* (LFSR). An LFSR is a sequence of bits defined by two parameters: a *starting sequence* and a *feedback rule*. The name linear feedback shift register comes from a popular hardware implementation of this sequence.

Notation. In this section it will be convenient for us to treat strings as indexed from 0 to $l-1$ (where l is the string's length) instead of being indexed from 1 to l.

Definition 3 *Let $l \in \mathbb{N}$ and let $\mathbf{s} = s_0 \cdots s_{l-1}$ and $\mathbf{f} = f_0 \cdots f_{l-1}$ be two l-bit strings. The* LFSR sequence defined by $\mathbf{s}$ and $\mathbf{f}$ *is the sequence $(r_0, r_1, r_2, \ldots)$ defined as follows:*

$$r_i \stackrel{\text{def}}{=} \begin{cases} s_i & 0 \leq i \leq l-1 \\ \sum_{j=0}^{l-1} f_j \cdot r_{(i-l)+j} & i \geq l \end{cases}$$

We call $\mathbf{s}$ *the* starting sequence *and* $\mathbf{f}$ *the* feedback rule.

Note that the definition of an LFSR sequence is a recursive definition, with each bit r_i (for $i \geq l$) depending only on the previous l bits. Therefore, this sequence will repeat itself after at most 2^l steps, or in other words, it is a periodic sequence with period at most 2^l.

Let us fix $l, n \in \mathbb{N}$ with $l < n$ (actually we will use $l \ll n$). We define the function $\mathrm{seq}_n : \{0,1\}^l \times \{0,1\}^l \to \{0,1\}^n$ as follows: for $\mathbf{s}, \mathbf{f} \in \{0,1\}^l$, $\mathrm{seq}_n(\mathbf{s}, \mathbf{f})$ consists of the first n bits of the LFSR sequence defined by $\mathbf{s}$ and $\mathbf{f}$. That is $\mathrm{seq}_n(\mathbf{s}, \mathbf{f}) = r_0 \cdots r_{n-1}$, where $r_0, \ldots, r_{n-1}$ are defined as in Definition 3.

Polynomials over $GF(2)$. To describe the rest of the construction, we will need to use polynomials with coefficients in $GF(2)$. Recall that one can define multiplication and division for such polynomials. We say that a polynomial is *irreducible* if it has no non-trivial factors (i.e., the only polynomials that divide it are itself and the polynomial $f(t) \equiv 1$). That is, irreducible polynomials in the polynomial ring correspond to prime numbers in the domain of natural numbers.

It will be useful for us to identify the feedback rule $\mathbf{f} = (f_0, \ldots, f_{l-1})$ with the following l-degree polynomial with coefficients in $GF(2)$:

$$f(t) = -t^l + \sum_{j=0}^{l-1} f_j \cdot t^j$$

(Note that $-1 = +1$ over $GF(2)$. The reason that we prefer to write the definition of the polynomial $f(t)$ in the above way, will become clear later on)

We say that $\mathbf{f}$ is a *valid* feedback rule if the corresponding polynomial is *irreducible* over $GF(2)$. We let $IP_l \subseteq \{0,1\}^l$ denote the set of all valid feedback rules (or equivalently, the set of all irreducible polynomials of degree exactly l over $GF(2)$). The following fact is well-known:

Fact 2.2 $|IP_l| = \Theta(\frac{2^l}{l})$.

2.2.2 The actual sample space

We define the following sample space

$$\Omega_{n,l} \stackrel{\text{def}}{=} \{\text{seq}_n(\mathbf{f},\mathbf{s}) : \mathbf{f} \in \mathcal{IP}_l, \mathbf{s} \in \{0,1\}^l\}$$

We have that $|\Omega_{n,l}| = \Theta(\frac{2^{2l}}{l})$. Moreover, it is possible to enumerate the elements of $\Omega_{n,l}$ in time polynomial in n and 2^l. Indeed, given $\mathbf{s},\mathbf{f} \in \{0,1\}^l$ the function $\text{seq}_n(\mathbf{s},\mathbf{f})$ can be evaluated in $poly(n)$ time, and enumerating all possible choices for $\mathbf{s},\mathbf{f}$ can be done in $2^{O(l)}$ time (note that checking whether a polynomial of degree l over $GF(2)$ is irreducible can be done trivially in time $2^{O(l)}$ by checking all possible factors

2.3 Analysis

We prove Theorem 2 by proving the following proposition:

Proposition 2.3 *For some constant k and for every $l,n \in \mathbb{N}$ with $l < n$, the set $\Omega_{n,l}$ is a $k \cdot \frac{n}{2^l}$-bias sample space over $\{0,1\}^n$.*

Proposition 2.3 immediately implies Theorem 2. This is because if we let $l = \log(k \cdot \frac{n}{\epsilon})$, then by Proposition 2.3, $\Omega_{n,l}$ is an ϵ-bias sample space over $\{0,1\}^n$ of size $O(\frac{2^{2l}}{l}) = O((\frac{n}{\epsilon})^2)$ that is constructible in $poly(\frac{n}{\epsilon})$-time.

We will start by some preliminary observations on the connection between the LFSR sequence and the polynomial that corresponds to the feedback rule. Then, in Section 2.3.2, we shall use these observations to prove Proposition 2.3.

2.3.1 Preliminary Observations

Let us fix $l < n$ to be two natural numbers as in the statement of Proposition 2.3. Let $\mathbf{s}$ and $\mathbf{f}$ be a starting sequence and a feedback rule in $\{0,1\}^l$ and let $(r_0,\ldots,r_{n-1}) = \text{seq}_n(\mathbf{s},\mathbf{f})$. We know that for any $i \geq l$

$$r_i = \sum_{j=0}^{l-1} f_j \cdot r_{(i-l)+j} \tag{1}$$

It is clear that any value r_i is a linear combination of the value of the starting sequence $s_0,\ldots,s_{l-1}$. That is, there exists an $n \times l$ matrix $C = (c_{i,j})$ over $GF(2)$ such that for any i

$$r_i = \sum_{j=0}^{l-1} c_{i,j} \cdot s_j \tag{2}$$

In matrix notation, we can write $C\mathbf{s} = \mathbf{r}$, where $\mathbf{r}$ is the vector $\text{seq}_n(\mathbf{s},\mathbf{f}) = (r_0,\ldots,r_{n-1})$. The entries of the matrix C can be obtained by repeated applications of Equation 1, and hence are determined by the value of the feedback rule $\mathbf{f}$. We will sometimes denote this matrix C by $C(\mathbf{f})$.

Consider the polynomial $f(t)$ of degree l that corresponds to $\mathbf{f}$ (that is, $f(t) = -t^l + \sum_{j=0}^{l-1} f_j \cdot t^j$). We can reduce any polynomial (of degree at least l) modulo this polynomial $f(t)$. In particular we have that

$$t^l = \sum_{j=0}^{l-1} f_j \cdot t^j - f(t) \equiv \sum_{j=0}^{l-1} f_j \cdot t^j \pmod{f(t)}$$

For $i \geq l$ we have that

$$t^i = t^{i-l}t^l \equiv t^{i-l}(\sum_{j=0}^{l-1} f_j t^j) \equiv \sum_{j=0}^{l-1} f_j t^{(i-l)+j} \pmod{f(t)} \tag{3}$$

Note the analogy between Equations 1 and 3. The polynomial t^i is equivalent (modulo $f(t)$) to a polynomial of degree $l-1$, or in other words to a linear combination of $t^0, \ldots, t^{l-1}$. The coefficients of this linear combination can be obtained by repeated application of Equation 3. The analogy between Equations 1 and 3 implies that these will be the *same coefficients* as in Equation 2. That is, the following holds:

$$t^i \equiv \sum_{j=0}^{l-1} c_{i,j} \cdot t^j \pmod{f(t)} \tag{4}$$

where $C = (c_{i,j})$ is the very same $n \times l$ matrix $C(\mathbf{f})$ defined above. In matrix notation we have that

$$C \begin{pmatrix} 1 \\ t \\ \vdots \\ t^{l-1} \end{pmatrix} \equiv \begin{pmatrix} 1 \\ t \\ \vdots \\ \vdots \\ t^{n-1} \end{pmatrix} \pmod{f(t)} \tag{5}$$

By applying the transpose operation to Equation 5, we obtain the following equation that will be useful for us later on:

$$(1 \cdots t^{l-1})C^T \equiv (1 \cdots t^{n-1}) \pmod{f(t)} \tag{6}$$

2.3.2 Proving Proposition 2.3

In order to prove that $\Omega_{n,l}$ is a $O(\frac{n}{2^l})$-bias sample space we need to show that the following holds for any non-empty $I \subseteq \{0, \ldots, n-1\}$:

$$\Pr_{\mathbf{r} \in \Omega_{n,l}} \left[\bigoplus_{i \in I} r_i = 0 \right] \in \frac{1}{2} \pm O(\frac{n}{2^l})$$

Equivalently, we need to prove that for any non-zero vector $\mathbf{v} \in \{0,1\}^n$

$$\Pr_{\mathbf{r} \in \Omega_{n,l}} [\langle \mathbf{v}, \mathbf{r} \rangle = 0] \in \frac{1}{2} \pm O(\frac{n}{2^l}) \tag{7}$$

where $\langle \cdot , \cdot \rangle$ denotes the standard dot-product (i.e., $\langle \mathbf{v}, \mathbf{r} \rangle \stackrel{\text{def}}{=} \mathbf{v}^T \mathbf{r} = \sum_{i=0}^{n-1} u_i \cdot r_i \pmod 2$). In this notation, the fact that the uniform distribution has bias 0 can be phrased as follows:

Fact 2.4 *Let* $\mathbf{v} \neq 0$ *be a vector in* $\{0,1\}^n$*, then*

$$\Pr_{\mathbf{r} \in \{0,1\}^n} [\langle \mathbf{v}, \mathbf{r} \rangle = 0] = \frac{1}{2}$$

Let us fix a non-zero vector $\mathbf{v} \in \{0,1\}^n$ and let p denote the probability defined in Equation 7. By the definition of the set $\Omega_{n,l}$ we know that

$$\begin{aligned} p &= \Pr_{\mathbf{s}\in\{0,1\}^l, \mathbf{f}\in IP_l} [\langle \mathbf{v}\,,\, \mathrm{seq}_n(\mathbf{s},\mathbf{f})\rangle = 0] \\ &= \Pr_{\mathbf{s}\in\{0,1\}^l, \mathbf{f}\in IP_l} [\langle \mathbf{v}\,,\, C(\mathbf{f})\mathbf{s}\rangle = 0] \\ &= \Pr_{\mathbf{s}\in\{0,1\}^l, \mathbf{f}\in IP_l} [\langle C(\mathbf{f})^T\mathbf{v}\,,\, \mathbf{s}\rangle = 0] \end{aligned}$$

where the last equality holds by a well-known property of the dot product. Yet, by Fact 2.4 above, we know that, conditioned on $C(\mathbf{f})^T\mathbf{v}$ being a non-zero vector, the above probability is exactly half. We also know that if $C(\mathbf{f})^T\mathbf{v}$ is the zero vector, then we will have that $\langle C(\mathbf{f})^T\mathbf{v}\,,\,\mathbf{s}\rangle = 0$ with probability 1. It follows that if we let A denote the event that $C(\mathbf{f})^T\mathbf{v}$ is the zero vector, and let ρ denote the probability (over $\mathbf{f} \in IP_l$) that A happens, then

$$p = (1-\rho)\cdot\frac{1}{2} + \rho\cdot 1 = \frac{1}{2} + \frac{1}{2}\cdot\rho$$

Therefore, to prove Proposition 2.3, all we need to do is to prove that $\rho = O(\frac{n}{2^l})$. Consider the case that the event A happens. In this case we have that

$$C(\mathbf{f})^T\mathbf{v} = \mathbf{0}$$

and so (for $C = C(\mathbf{f})$) it holds that for any l-bit vector $\mathbf{x}$

$$\mathbf{x}C^T\mathbf{v} = 0$$

(note that in the first equation $\mathbf{0}$ denotes the zero l-bit vector, while in the second equation we use the scalar 0).

In particular if we plug $\mathbf{x} = (1\cdots t^{l-1})$ we will get that the polynomial $(1\cdots t^{l-1})C^T\mathbf{v}$ is the zero polynomial (modulo $f(t)$). Combining this with Equation 6 we get that

$$(1\cdots t^{n-1})\mathbf{v} \equiv 0 \pmod{f(t)}$$

which means that the polynomial $f(t)$ divides the non-zero polynomial $g_{\mathbf{v}}(t) \stackrel{\mathrm{def}}{=} (1\cdots t^{n-1})\mathbf{v} = \sum_{i=0}^{n-1} u_i \cdot t^i$. Note that the polynomial $g_{\mathbf{v}}(t)$ depends only on $\mathbf{v}$ and has degree at most n. We have that $\rho \leq \Pr_{\mathbf{f}\in IP_l}[f(t) \text{ divides } g_{\mathbf{v}}(t)]$. Yet any polynomial of degree at most n has at most $\frac{n}{l}$ *irreducible*[1] factors of degree l, and so the probability that $f(t)$ happens to be one of these factors is at most $\frac{n/l}{|IP_l|}$, which by Fact 2.2 is equal to $\frac{n/l}{\Theta(2^l/l)} = O(\frac{n}{2^l})$.
■

3 Using Small Size Sample Spaces for proving Hardness of MaxQE

In this section we present the problem of Quadratic Equations satisfiablity (QE). We state that the QE problem is $\mathcal{NP}$-Complete (and sketch the proof) and discuss the approximability of the

[1] Note that because different *reducible* polynomials may have shared factors, it may be the case that a polynomial of degree n has much more than n/l factors of degree l that are *not* irreducible. Therefore it is crucial that we limit ourselves only to valid feedback rules (i.e., feedback rules that correspond to irreducible polynomials).

optimization variant, denoted as MaxQE. We state that this problem can be approximated with a ratio of one half, and we use a construction of small bias sample spaces (e.g., from Theorem 2), to prove that this bound is tight and no polynomial algorithm can approximate MxaQE with a better constant ratio (unless $\mathcal{P} = \mathcal{NP}$).

3.1 The problem of Quadratic Equations

A quadratic expression is a multi-variate polynomial with maximal degree of 2. A quadratic equation is an equation whose left side is a quadratic expression, and whose right side is a scalar. Given a set of quadratic equations sharing the same variables, a question that arise is whether there exists an assignment for this variables, which satisfies all equations, or formally:

Definition 4 *The set* Quadratic Equations *denoted QE, consists of all mutually satisfiable sequences of quadratic equations over $GF(2)$.*

Since in $GF(2)$ it holds that $x = x^2$ for any number x, we can assume without loss of generality that all the summands are either of degree 2 or are constants. The formulation of QE as a decisional problem is given by:

Input A sequence of n quadratic equations with t variables over $GF(2)$

Question Is there an assignment to the variables that satisfies *all* the equations.

Theorem 5 $QE \in \mathcal{NPC}$

Proof Sketch: Clearly QE is in $\mathcal{NP}$ as a satisfying assignment is an easily verifiable witness. To see that it is $\mathcal{NP}$*-hard* we will see a reduction from $3SAT$.

Consider an instance of $3SAT$, denoted $\varphi = \bigwedge_{i=1}^{m} C_i$, where each clause C_i is the disjunction of three literals l_{i1}, l_{i2} and l_{i3}, each of them being either one of the variables $p_1, \ldots, p_n$ or its negation.

The reduction consists of two stages, the first one translating the formula into a sequence of m cubic equations, while the second translates this sequence into a sequence of quadratic equations.

The first translation is natural, as $3CNF$ formulas have immediate correspondence to cubic equations. For example, a Boolean equation of the form $p_1 \vee p_2 \vee \neg p_3 \stackrel{?}{=} TRUE$ can be translated by De-Morgan's rules into $\neg p_1 \wedge \neg p_2 \wedge p_3 \stackrel{?}{=} FALSE$, which corresponds to the cubic equation $(1 - x_1)(1 - x_2)x_3 \stackrel{?}{=} 0$.

It is clear that the 3SAT input φ is satisfiable if and only if the sequence of cubic equations is satisfiable. Next, we describe the translation of these equations into quadratic equations. For each pair of variables x_i and x_j in the cubic system, we introduce a new variable z_{ij} and a new equation $z_{ij} = x_i x_j$. Then, we go through all the equations and whenever we find a term of degree 3 (e.g., $x_i x_j x_k$) we replace it by a term of degree 2 (e.g., $z_{ij} x_k$) by replacing the multiplication of two variables $x_i x_j$ with the new variable z_{ij}. Our new equation system is composed of two parts, the first consists of the new added equations $\{z_{ij} = x_i x_j\}_{i,j=1}^{n}$ (and thus is quadratic in the size of the original input), and the second part is the transformed sequence of equations all of degree 2 (linear in the size of the input).

Consider for example, the 3SAT input

$$(x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_2 \vee x_3 \vee x_4) \wedge (x_1 \vee x_4 \vee x_5) \stackrel{?}{=} TRUE$$

By applying De-Morgan's rules we get an equivalent formula

$$(\neg x_1 \wedge \neg x_2 \wedge x_3) \vee (x_2 \wedge \neg x_3 \wedge \neg x_4) \vee (\neg x_1 \wedge \neg x_4 \wedge \neg x_5) \stackrel{?}{=} FALSE$$

which is (notice that for the formula to be satisfied, each of the clauses must be FALSE)

$$\begin{pmatrix} \neg x_1 \wedge \neg x_2 \wedge x_3 \\ x_2 \wedge \neg x_3 \wedge \neg x_4 \\ \neg x_1 \wedge \neg x_4 \wedge \neg x_5 \end{pmatrix} \stackrel{?}{=} \begin{pmatrix} FALSE \\ FALSE \\ FALSE \end{pmatrix}$$

This triplet of Boolean equations is translated into the cubic equations

$$\begin{pmatrix} (1-x_1)(1-x_2)x_3 \\ x_2(1-x_3)(1-x_4) \\ (1-x_1)(1-x_4)(1-x_5) \end{pmatrix} \stackrel{?}{=} \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

which is

$$\begin{pmatrix} x_1x_2x_3 - x_1x_3 - x_2x_3 + x_3 \\ x_2x_3x_4 - x_2x_3 - x_2x_4 + x_2 \\ -x_1x_4x_5 + x_4x_5 + x_1x_4 + x_1x_5 - x_4 - x_5 - x_1 + 1 \end{pmatrix} \stackrel{?}{=} \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

Notice that all the cubic terms include either the term x_2x_3 or the term x_1x_4, and thus it suffices to replace only these quadratic products by new variables, i.e. z_{23} and z_{24} to reduce the maximal degree into 2

$$\begin{pmatrix} x_1x_4 - z_{14} \\ x_2x_3 - z_{23} \\ x_3 + x_1z_{23} - x_1x_3 - z_{23}) \\ x_2 + z_{23}x_4 - z_{23} - x_2x_4 \\ -z_{14}x_5 + x_4x_5 + x_1x_4 + x_1x_5 - x_4 - x_5 - x_1 + 1 \end{pmatrix} \stackrel{?}{=} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

The last technical step is to go over the equation sequence and apply the following three technical transforms: a) Replace subtractions by additions (its the same in $GF(2)$); b) Replace every linear term by its square; c) Move the scalar '1' to the right side. Finally, we get an input for the QE problem

$$\begin{pmatrix} x_2x_3 + z_{23}^2 \\ x_1x_4 + z_{14}^2 \\ x_3^2 + x_1z_{23} + x_1x_3 + z_{23}^2 \\ x_2^2 + z_{23}x_4 + z_{23}^2 + x_2x_4 \\ z_{14}x_5 + x_4x_5 + x_1x_4 + x_1x_5 + x_4^2 + x_5^2 + x_1^2 \end{pmatrix} \stackrel{?}{=} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

which is satisfiable if and only if the original 3SAT input φ was satisfiable. Since the entire procedure of reducing a $3CNF$ to a sequence of quadratic equations can be done in polynomial time we have reduced $3SAT$ to QE. ■

3.2 The Optimization Problem MaxQE

Next, we define the optimization variant of QE, denoted MaxQE:

Input A sequence of n quadratic equations with t variables over $GF(2)$

Problem Find an assignment that satisfies *as many* of the equations *as possible*.

We define the function $MaxQE(I) \in [0, 1]$ to be the maximal fraction of equations in I, which can be mutually satisfied by a single assignment.

Example: The following is an instance for both QE and MaxQE, with the parameters $n = 5$ and $t = 6$.

$$\begin{cases} x_2x_3 + z_{23}^2 = 0 \\ x_1x_4 + z_{14}^2 = 0 \\ x_3^2 + x_1z_{23} + x_1x_3 + z_{23}^2 = 0 \\ x_2^2 + z_{23}x_4 + z_{23}^2 + x_2x_4 = 0 \\ z_{14}x_5 + x_4x_5 + x_1x_4 + x_1x_5 + x_4^2 + x_5^2 + x_1^2 = 1 \end{cases}$$

Notice that the assignment

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ z_{23} \\ z_{14} \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ * \\ 1 \\ 0 \end{pmatrix}$$

satisfies all the equations (there are several satisfying assignments for this QE input), and consequently the the answer to the decisional problem is "YES", whereas the answer for the MaxQE problem is one of the satisfying assignments (and $MaxQE(I) = 1$).

The best known poly-time approximation algorithm for MaxQE gives an approximation ratio of $\frac{1}{2}$. The algorithm that achieves this ratio (and its analysis) are beyond the scope of this course and are omitted.

3.3 MaxQE is hard to $(\frac{1}{2} + \epsilon)$-approximate

Theorem 6 *For any $\epsilon > 0$ MaxQE is $\mathcal{NP}$-Hard to approximate to within factor $\frac{1}{2} + \epsilon$.*

Before getting to the proof itself, we present new terminology which we use in the proof. We extend the decisional QE problem into a family of problems, denoted $GapQE_{\alpha,\beta}$, parameterized by $\alpha, \beta \in [0, 1]$. For this purpose let us define a promise problem. Loosely speaking, a promise problem is a problem for which the input-output relation is only partially defined, i.e. for some inputs the output is strictly defined (as in typical problems), while for other inputs the output is insignificant.

Next we define two classes of QE inputs

$\forall\alpha$ $YesQE_\alpha$:
A sequence of n quadratic equations with t variables over $GF(2)$ such that there is an assignment to the variables that satisfies at least a fraction α of the equations.

$\forall\beta$ $NoQE_\beta$:
A sequence of n quadratic equations with t variables over $GF(2)$ such that all the assignments to the variables satisfy strictly less than a fraction β of the equations.

Example: $YesQE_1$ is the set of all satisfiable QE inputs, while $NoQE_1$ is the set of all other QE inputs (those which are not satisfiable).

The problem $GapQE_{\alpha,\beta}$ (for $\alpha \geq \beta \in [0, 1]$), is the promise problem of distinguishing $YesQE_\alpha$ inputs from $NoQE_\beta$. However, the fact that $NoQE_\beta$ includes inputs whose best assignments satisfy *strictly less* than a fraction of β, implies that $GapQE_{\alpha,\alpha}$ is defined. Notice that in this case the

output is defined for inputs which are α-satisfiable (and thus in $YesQE_\alpha$) and for inputs which are not α-satisfiable (and thus in $NoQE_\alpha$), i.e. for all inputs. Consequently $GapQE_{\alpha,\alpha}$ is actually a standard decisional problem.
Example: $GapQE_{1,1} \equiv QE$

Lemma 3.1 *Suppose that there exists a polynomial time β-approximation algorithm for QE. Then $GapQE_{1,\beta}$ is decidable in polynomial time.*

Proof: For QE inputs from $YesQE_1$, a β-approximation algorithm is promised to provide an assignment satisfying at least β of the equations. For inputs of $NoQE_\beta$ no algorithm can find such an assignment, since no assignment satisfies β of the equations. Thus, applying such an algorithm to some QE input, one can compare the number of satisfied equations to β, and decide about the output for $GapQE_{1,\beta}$ accordingly. ■

The proof of Theorem 6 is constructed in the following way: we prove that for every $\epsilon > 0$, $GapQE_{1,\frac{1}{2}+\epsilon}$ is $\mathcal{NP}$-Hard, and use Lemma 3.1 to conclude that $MaxQE$ is $\mathcal{NP}$-Hard to $\frac{1}{2}+\epsilon$-approximate.

Lemma 3.2 *For every $\epsilon > 0$, there exists a polynomial reduction from QE to $GapQE_{1,\frac{1}{2}+\epsilon}$*

Proof: Let I be an input for QE. We show a polynomial time algorithm that constructs from I an input I' for $Gap_{1,\frac{1}{2}+\epsilon}$, where $I' \in YesQE_1 \cup NoQE_{\frac{1}{2}+\epsilon}$ (i.e. the output I' satisfies the promise of $GapQE$), and furthermore $I' \in YesQE_1$ if and only if $I \in QE$. (It follows that $I' \in NoQE_{\frac{1}{2}+\epsilon}$ if and only if $I \notin QE$.)

Let us first introduce some notations we use in the proof. Whenever we deal with a sequence I of n quadratic equations with the Boolean variables $x_1, \ldots x_t$, we denote every equation as e_k ($1 \leq k \leq n$), the coefficients of the quadratic terms on e_k as $c_{i,j}^{(k)}$ and the free term of e_k as $b^{(k)}$. That is, such a sequence is denoted by

$$\left\{ e_k : \sum_{i=1}^{t} \sum_{j=1}^{t} c_{i,j}^{(k)} x_i x_j = b^{(k)} \right\}_{k=1}^{n}$$

Whenever we deal with a sample space $\Omega \subseteq \{0,1\}^n$, we denote its size by m (and use $m = poly(n)$), its l^{th} element as $s^{(l)}$ and the k^{th} bit of $s^{(l)}$ as $s_k^{(l)}$. Thus, $\Omega = \{s^{(1)}, \ldots, s^{(m)}\}$) and $s^{(l)} = (s_1^{(l)}, s_2^{(l)}, \ldots, s_n^{(l)})$.

Let I be an input for QE. The transformation algorithm is the following:

1. Construct an ϵ-bias sample space Ω over $\{0,1\}^n$ of size $m = poly(n)$.

2. Construct I', a QE input with the same variables $x_1, \ldots, x_t$ as I, but with linear combinations of $e_1, \ldots, e_n$ as the quadratic equations. The linear combinations are those corresponding to the elements $s_1, \ldots, s_m$ of Ω. That is, for every l for which $1 \leq l \leq m$, we define f_l to be the quadratic equation

$$f_l : \sum_{k=1}^{n} s_k^{(l)} e_i$$

which is

$$\sum_{k=1}^{n} s_k^{(l)} \underbrace{\sum_{i=1}^{t} \sum_{j=1}^{t} c_{i,j}^{(k)} x_i x_j}_{LEFT(e_i)} = \sum_{k=1}^{n} s_k^{(l)} \cdot \underbrace{b^{(k)}}_{RIGHT(e_i)}$$

in other words the new l^{th} equation is

$$\sum_{i=1}^{t}\sum_{j=1}^{t}\underbrace{\sum_{k=1}^{n} s_k^{(l)}\cdot c_{i,j}^{(k)}}_{\text{coefficient of } f_l}\cdot x_i x_j=\underbrace{\sum_{k=1}^{n} s_k^{(l)}\cdot b^{(k)}}_{\text{free term of } f_l}$$

We now prove that the this transformation is indeed a polynomial reduction.

Claim 3.3 *The transformation can be applied in polynomial time*

Proof: Constructing a polynomial sized ϵ-bias sample space (see Section 2) can be done in polynomial time (i.e., polynomial in n and $\frac{1}{\epsilon}$, where here ϵ is a fixed constant). The time required for the second stage (generating the new equations) is linear in the size of the space, i.e. polynomial. ■

Claim 3.4 *I is satisfiable $\Longrightarrow I' \in YesQE_1$*

Proof: Suppose that there is an assignment satisfying all the equations of I, and let $\mathcal{T} = (\tau_1, \ldots, \tau_t)$ be this assignment. Notice that an assignment that satisfies some equations, satisfies also every linear combination of them. Consequently $\mathcal{T}$ satisfies all the equations of I', i.e. $I' \in YesQE_1$. ■

Claim 3.5 *I is not satisfiable $\Longrightarrow I' \in NoQE_{\frac{1}{2}+\epsilon}$*

Proof: Let I be an unsatisfiable sequence of quadratic equations. We need to prove that no assignment of $x_1, \ldots, x_t$, satisfies a fraction of $\frac{1}{2}+\epsilon$ of the equations of I'. Fixing any assignment $\mathcal{T}$ to the variables of I, we show that $\mathcal{T}$ satisfies less than $\frac{1}{2}+\epsilon$ of the equations in I'. Our argument relies on the following fact:

Fact 3.6 *Let $\mathcal{T}$ be an assignment to $x_1, \ldots, x_t$, and let $K \subseteq \{1, \ldots, n\}$ be the set of indices of equations in I that the assignment $\mathcal{T}$* does not *satisfy. Let f be a linear combination of the quadratic equations in I, that is $f = \sum_{k=1}^{n} s_k e_k$ where $s_1, \ldots, s_n \in \{0, 1\}$. Then $\mathcal{T}$ satisfies f if and only if*

$$\bigoplus_{k \in K} s_k = 0$$

that is, the number of k's such that $k \in K$ and $s_k = 1$ is even.

Proof: Every s_k that is assigned 1, represents an equation that is included within f. Each of these k's that is in K, represents an equation that is not satisfied by $\mathcal{T}$, and contributes 1 to the difference between $LETF(f)$ and $RIGHT(f)$.

Recall that f is an equation (mod2), and thus a pair of contributions of 1 cancel each other. Thus, whenever the number of these contributions is even, their total difference is summed up to 0, and otherwise a difference of 1 remains, which means that in this case the equation f is not satisfied by the assignment $\mathcal{T}$. ■

Motivating Discussion: Before explicitly proving the upper bound on the number of satisfied equations in I', which has a somewhat technical proof, we first give some intuition for the main ideas in constructing it, via proving such an upper bound for slightly different construction of I'.

Consider the construction of I', which instead of sampling m elements of $\{0,1\}^n$, takes the whole $\{0,1\}^n$ as Ω (note that such an instance I' has exponential size and cannot be constructed in polynomial time from I), which is a perfect sample space. Let $\mathcal{T}$ be an assignment, and recall that I is not satisfiable, and thus the set K of equations that are not satisfied by $\mathcal{T}$ is non-empty.

Recall that the uniform distribution is an 0-bias sample space (as described in Section 2), which means that for every non-empty K, the fraction of n-bit vectors, whose $\bigoplus$ over K equals 0, is exactly one half. Thus, we can apply Fact 3.6 and conclude that the fraction of s's that induce equations unsatisfied by $\mathcal{T}$, is exactly one half.

We return now to the original construction of I'. Although our sample space has only polynomial size ($m = poly(n)$), recall that the main property of an ϵ-bias sample space is that the parity bit of *every* non-empty subset, has an almost uniform distribution in Ω (it has at most a bias of $\frac{\epsilon}{2}$ towards each of the values in $\{0,1\}$). Consequently, the fraction of linear combinations where the number of unsatisfied e_k's is even, is bounded from above by $\frac{1}{2}+\frac{\epsilon}{2}$ (and from below by $\frac{1}{2}-\frac{\epsilon}{2}$) (details follow).

Let $\mathcal{T}$ be an assignment, and suppose that $\mathcal{T}$ does not satisfy *some* of the equations $e_1,\ldots,e_n$ in I (as must be the case since $I \notin QE$). Let $K \subseteq \{1,\ldots,n\}$ be the set of indices of equations in I that are not satisfied by $\mathcal{T}$. We know that K is not empty as I is not satisfiable. An equation f_l in I' is the linear combination of the equations $\{e_k\}_{k=1}^n$, with coefficients corresponding to the element $s^{(l)}$ in Ω. Again we use the observation of Fact 3.6 that $\mathcal{T}$ satisfies f_l if and only if $\bigoplus_{k\in K} s_k^{(l)} = 0$. However, the fraction of these $s^{(l)}$'s in Ω is bounded from above by $\frac{1}{2}+\frac{\epsilon}{2}$. This fact stems from the definition of ϵ-bias sample spaces.

Thus, the fraction of f_l's that are satisfied by $\mathcal{T}$ is at most $\frac{1}{2}+\frac{\epsilon}{2}$, which is strictly smaller than $\frac{1}{2}+\epsilon$. ∎

Thus, the reduction is polynomial and valid and the lemma follows. ∎