Topics in Theoretical Computer Science March 25, 2014

Lecture 6

Lecturer: Ola Svensson *Scribes: Joachim Hugonot, Fabien Jolidon*

1 Summary

In the previous lecture we saw the PCP Theorem which states that $NP = PCP(log(n), 1)$, and we proved that $PCP(log(n), 1) \subseteq NP$, but the proof of $NP \subseteq PCP(log(n), 1)$ is unfortunately too long and we will only prove a weaker result : $NP \subseteq PCP(poly(n), 1)$. In order to prove this result, we need to devise a probabilistic verifier for an NP-complete problem (namely QUADEQ), that runs in polynomial time, performs $O(poly(n))$ random coin flips, and reads $O(1)$ bits from the proof. In this lecture, we will first introduce the Walsh-Hadamard code (WH), then use its properties to design the required verifier.

2 Walsh-Hadamard code

Definition 1 *: The WH code encodes n bits into 2^n bits. The encoding function WH: $\{0,1\}^n \to \{0,1\}^{2^n}$ maps the input $u \in \{0,1\}^n$ into the truth table of the function $x \odot u = \sum_{i=1}^{n} x_i u_i$ mod 2.*

Let's see an example where we encode all possible bit strings x of length 2 into 4 bits :

- For WH(0,0), $x \mapsto (0,0) \odot x$.
- For WH(0,1), $x \mapsto (0,1) \odot x = x_2$.
- For WH(1,0), $x \mapsto (1,0) \odot x = x_1$.
- For WH(1,1), $x \mapsto (1,1) \odot x = x_1 \oplus x_2$.

x	WH(0,0)	WH(0,1)	WH(1,0)	WH(1,1)
(0,0)	0	0	0	0
(0,1)	0	1	0	1
(1,0)	0	0	1	1
(1,1)	1	1	1	0

We can see from this example that the codewords of Walsh-Hadamard code are the truth table of all linear functions over $\mathbb{F}_2^n$. We will see in the following subsection that the WH code has two interesting properties, namely its *local testability* and *local decodability.*

2.1 Local testability of the Walsh-Hadamard

Given a function $f : \{0,1\}^n \mapsto \{0,1\}$, we would like to whether there exist $u \in \{0,1\}^n$ such that $f = WH(u)$. In other word words, we want to know if the function f is linear. f is linear if and only if $f(x+y) = f(x) + f(y)$ $\forall x, y \in \{0,1\}^n$ mod 2. In order to check if f is linear, we do the following test.

Linearity test

1. Select $x, y \in \{0,1\}^n$ independently uniformly at random.
2. Check if $f(x+y) = f(x) + f(y)$ mod 2.

We see that all linear functions are always accepted.

Definition 2 *Two functions f, g are $(1-\epsilon)$-close if they agree on all, but ϵ-fraction of the inputs.*

Theorem 3 *:* **The Blum, Luby, Rubinfeld (BLR) linearity test**
Let $f : \{0,1\}^n \mapsto \{0,1\}$ if $Pr(f(x+y) = f(x) + f(y)) \geq \rho$ for $\rho \frac{1}{2}$ then f is ρ close to some linear function.

Corollary 4 *:*
For any $\delta \in (0, \frac{1}{2})$, there exists a linearity test that reads $O(\frac{1}{\delta})$ bits such that:

- *(Completeness:) if f is linear, the test accepts with probability 1.*
- *(Soundness:) if f is not $(1-\delta)$-close to a linear function, then the test accepts with probability $p \leq \frac{1}{2}$.*

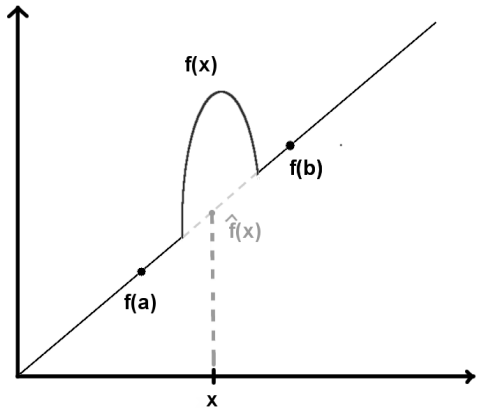
Proof We repeat the test of the theorem $\frac{1}{\delta}$ many times independently.

- Completeness: f is linear then we always accepts
- Soundness: if f is not $(1-\delta)$-close to a linear function then a single test accepts with probability $p \leq 1-\delta$. Since we do $\frac{1}{\delta}$ tests independently, the probability that they all succeed is $\leq (1-\delta)^{\frac{1}{\delta}} \leq \frac{1}{e}$.

■

2.2 Local decodability of the Walsh-Hadamard code

Given a function f that is $(1-\delta)$-close to some linear function $\hat{f}$, how to find $\hat{f}(x)$?



Since the function $\hat{f}(x)$ is linear we can simply compute $f(x) = \frac{1}{2} \cdot (f(a) + f(b))$

1. Select x'$\in \{0,1\}^n$ uniformly at random.
2. Output $f(x+x') + f(x')$.

Claim 5 *The algorithm outputs $\hat{f}(x)$ with probability $\geq 1 - 2 \cdot \delta$*

Proof Since f is $(1-\delta)$-close to $\hat{f}$, we have that

$$\Pr\left[f(x+x') = \hat{f}(x+x')\right] \geq 1-\delta$$
$$\Pr\left[f(x') = \hat{f}(x')\right] \geq 1-\delta$$

Hence by the union bound we get:

$$\Pr\left[f(x+x') = \hat{f}(x+x') \wedge f(x') = \hat{f}(x')\right] \geq 1-2\delta$$

In that case we get:

$$f(x+x') + f(x') = \hat{f}(x') = \hat{f}(x+x') + \hat{f}(x') = \hat{f}(x) + \hat{f}(x') = \hat{f}(x)$$

■

We will prove that NP ⊆ PCP(poly(n),1) by showing that QUADEQ, an NP-complete problem, has a probabilistic verifier that uses $O(poly(n))$ random bits and queries $O(1)$ bits from proof.

3 The QUADEQ Problem

In order to prove that $NP \subset PCP(poly(n), 1)$, we will prove that an NP-complete problem has a proof-verifier that uses $O(Poly(n))$ random bits and queries $O(1)$ bits from the proof.

Definition 6 *(QUADEQ) Given a system of m quadratic equations over n variables in $\mathbb{F}_2$, decide whether there exists an assignment of the variables such that all equations are satisfied.*

Example 1 *Given the following system:*

$$u_1u_2 + u_3u_1 + u_2u_3 = 1$$
$$u_2u_2 + u_1 = 0$$
$$u_3u_2 + u_2u_1 + u_1u_1 = 1$$

The system is satisfied when $u_1 = u_2 = u_3 = 1$.

Since we are working in $\mathbb{F}_2$, we have that $x = x^2$, hence by replacing any linear term by its square value, we will only have quadratic terms in our equations. For n variables, there are n^2 different possible quadratic terms (There are actually $\frac{n(n+1)}{2}$ different quadratic terms, if we consider that $u_iu_j = u_ju_i$, but it won't make a difference for what we are going to do). Let U be a vector containing the n^2 possible quadratic terms:

$$U = \begin{bmatrix} u_1u_1 \\ u_1u_2 \\ u_1u_3 \\ \dots \\ \\ u_nu_n \end{bmatrix}$$

We can then represent the problem with a linear system of n^2 variables $\{U_{1,1}, U_{1,2}, ..., U_{n,n}\}$ with the added constraint that $U_{i,j} = u_iu_j$.

We can now express our problem with matrices.

Let $A \in \mathbb{F}_2^{m \times n^2}$ and $b \in \mathbb{F}_2^m$. We want to find $U \in \mathbb{F}_2^{n^2}$ such that $AU = b$.

Example 2 *We can rewrite our previous example as:*

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} U_{1,1} \\ U_{1,2} \\ U_{1,3} \\ U_{2,1} \\ U_{2,2} \\ U_{2,3} \\ U_{3,1} \\ U_{3,2} \\ U_{3,3} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

4 Exercises

Exercise 1 *What is the rate and distance of the Welsch-Hadamard code?*

The rate of a code is defined as $rate = \frac{log(\#\ of\ code\ words)}{\#\ of\ bits\ of\ a\ code\ word}$ Here we trivially have that $rate = \frac{log(2^n)}{2^n} = \frac{n}{2^n}$

If we take 2 different code words, they correspond to 2 linear functions $f(x) = x \oplus u$ and $g(x) = x \oplus v$ with $u \neq v$. We have then $f(x) \neq g(x)$ if and only if $(x \odot u) \oplus (x \odot v) = (u \oplus v) \odot x = 1$. Because of the fact that $u \neq v$, $(u \oplus v)$ will never be null, and thus, the equation will be satisfied for half of the possible x's which means the relative distance is $1/2$.

Exercise 2 *Prove that QUADEQ is NP-complete. For this you will use a reduction from circuit-SAT. Write the AND, OR and NOT gates using quadratic equations in $\mathbb{F}_2$.*

For example for AND, we have 2 inputs x_1 and x_2, and one output y. Find a quadratic system so that $y = x_1 \wedge x_2$

NOT: $x + y = 1$
AND: $x_1 x_2 + y = 0$
OR: $x_1 + x_2 + x_1 x_2 + y = 0$

5 Random Subsum property

Using the result of the first exercise, we can prove the following affirmation:

Lemma 7 *For any $v, u \in \mathbb{F}_2^n$ such that $v \neq u$, we have that $v \odot x \neq u \odot x$ for half of the possible $x \in \mathbb{F}_2^n$.*

6 PCP(Poly(n),1) verifier for QUADEQ

Given $A \in \mathbb{F}_2^{m \times n^2}$, $b \in \mathbb{F}_2^m$
decide whether there exists a pair (u, U) such that

1. $AU = b$
2. $U = u \otimes u$

The verifier expects as a proof the Walsh-Hadamard encoding of U and u, $WH(U), WH(u)$.

Our verifier will treat the proof as two functions:

$$f : \mathbb{F}_2^{n^2} \to \mathbb{F}_2 \ (WH(U))$$

$$g : \mathbb{F}_2^{n} \to \mathbb{F}_2 \ (WH(u))$$

Step 1

Do a linearity test that only accepts with probability 0.01 if one of the functions is not 99.99% close to linear.

Step 2

Check that $g = WH(U)$ and $f = WH(u)$ when $U = u \otimes u$.

Claim 8 *If $U = u \otimes u$ then $g(r \otimes r') = f(r)f(r')$*

Proof

$$g(r \otimes r') = \sum_{ij} r_i r'_j U_{ij}$$

$$f(r)f(r') = \sum_i u_i r_i \sum_j u_j r'_j = \sum_{ij} r_i r'_j u_i u_j$$

■

Claim 9 *If $U \neq u \otimes u$ then $Pr[g(r \otimes r') = f(r)f(r')] \leq 3/4$*

Proof

$$g(r \otimes r') = \sum_{ij} r_i r'_j U_{ij} = r^t U r'$$

$$f(r)f(r') = \sum_i u_i r_i \sum_j u_j r_j = \sum_{ij} r_i r'_j u_i u_j = r^t (uu^t) r'$$

Now since $U \neq uu^t$, there exist at least one column c_1 in U, and one column c_2 in uu^t such that $c_1 \neq c_2$, and they the same position in U and uu^t respectively. From the random subsum principle, we know that $r^t c_1 \neq r^t c_2$ for exactly half the possibility of r. Fixing an r such that $r^t U \neq r^t (uu^t)$, we get that for half the possibilities of r', we have that $r^t U r' \neq r^t (uu^t) r'$, hence we get that for at least one fourth of the possible values of (r, r'), $(r \otimes r') \neq f(r)f(r')$. ■

Step 3

1. Select a subset $S \subseteq [m]$ of the rows by including each row with probability 1/2 independently.
2. let $\hat{a} = \sum_{i \in S} A_i$ and $\hat{b} = \sum_{i \in S} b_i$
3. check if $g(\hat{a}) = \hat{b}$

Completeness

$$g(\hat{a}) = \hat{a}U = \sum_{i \in S} A_i U = \sum_{i \in S} b_i = \hat{b}$$

If $f = WH(u)$ and $g = WH(U)$, we will pass the verifier with probability 1.

Soundness

If either one of f and g is not 99%- close to a linear function, we will reject probability at least $\frac{1}{2}$ in step 1. If both of them are 99%- close to linear functions, then given that we are choosing a subset of rows instead of a single one, step 2 will reject will probability at least $\frac{1}{2}$.

On the other hand, we have used $O(poly(n))$ random bits to choose the subset of rows, and queried constants number of bits from the proof to check if $g(\hat{a}) = \hat{b}$.