Scribe notes
Vazirani class
Lecture #7 (Tuesday, September 16)

1 The problem

We use the finite field Z_n^2. For $x, y \in Z_n^2$, $x + y$ denotes the bitwise addition and $x \cdot y$ denotes the inner product $(\sum x_i y_i mod 2)$.

Input for Simon's algorithm is a reversible circuit C_f computing $f : Z_n^2 \to Z_n^2$ such that

(a) f is one-to-one or

(b) f is two-to-one and there exists u such that $f(x) = f(x + u)$ for all $x \in Z_n^2$.

Simon's algorithm determines whether f satisfies (a) or (b) and, in the second case, finds u.

2 Efficient quantum algorithm

Simon's algorithm uses a following quantum circuit.

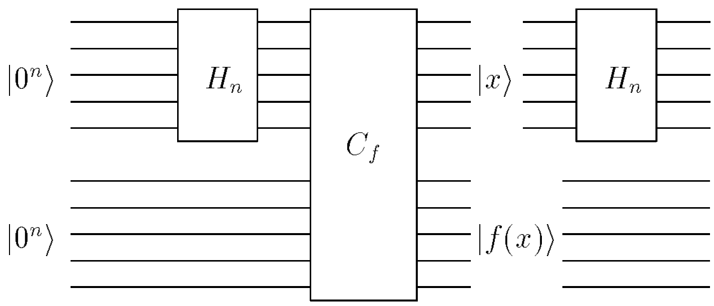

Figure 1: Simon's circuit

Claim. If f is two-to-one and there is u such that $f(x) = f(x+u)$, then the output of the circuit is y such that $y \cdot u = 0$. Moreover, y is uniformly distributed over all such y.

Simon's algorithm runs this circuit $n-1$ times and obtains $n-1$ vectors $y^{(1)}$, $y^{(2)}$, ..., $y^{(n-1)}$. If all $n-1$ vectors are linearly independent, they give a system of linear equations $y^{(i)} \cdot u = 0$ that can be resolved, obtaining u. After that, algorithm checks that $f(x) = f(x+u)$ for some x. If yes, f is two-to-one with this u. If no, f is one-to-one. If some of vectors are linearly dependent, it just runs the circuit once again, until $n-1$ independent vectors are obtained.

Proof of the claim. We start by calculating the quantum state. After Hadamard transform H_n, the state is

$$\frac{1}{2^{n/2}} \sum_x |x\rangle.$$

After the circuit C_f, the state is

$$\frac{1}{2^{n/2}} \sum_x |x\rangle |f(x)\rangle.$$

Bits $|f(x)\rangle$ are not used after that. Hence, we can apply the principle of safe storage:

The principle of safe storage. If some bits in a quantum circuit are not changed after some moment, then the outcome of the circuit is the same as in the case if these bits are measured immediately.

Let $|f(z)\rangle$ be the result of measuring $|f(x)\rangle$. There are exactly two x such that $f(x) = f(z)$: one is z and another is $z+u$. Hence, the quantum state after measuring $|f(x)\rangle$ is

$$\frac{1}{\sqrt{2}}|z\rangle|f(z)\rangle + \frac{1}{\sqrt{2}}|z+u\rangle|f(z)\rangle.$$

If we now measure $|x\rangle$, we get z or $z+u$, but not both. This would not give much information (just a random value $x \in Z_n^2$ and $f(x)$). Instead of measuring $|x\rangle$, we do Hadamard transform. After it, the state is

$$H_n \left(\frac{1}{\sqrt{2}}|z\rangle + \frac{1}{\sqrt{2}}|z+u\rangle \right) =$$

$$\frac{1}{\sqrt{2}} \sum_{y\in\{0,1\}^n} \frac{(-1)^{z\cdot y}}{2^{n/2}}|y\rangle + \frac{1}{\sqrt{2}} \sum_{y\in\{0,1\}^n} \frac{(-1)^{z\cdot(y+u)}}{2^{n/2}}|y\rangle =$$

$$\frac{1}{2^{n/2}\sqrt{2}} \sum_{y\in\{0,1\}^n} (-1)^{z\cdot y} + (-1)^{z\cdot y}(-1)^{u\cdot y}]|y\rangle =$$

$$\frac{1}{2^{n/2}\sqrt{2}} \sum_{y\in\{0,1\}^n} (-1)^{z\cdot y}[1 + (-1)^{u\cdot y}]|y\rangle.$$

Hence, if $u \cdot y = 1$, the amplitude of $|y\rangle$ is 0. All $|y\rangle$ such that $u \cdot y = 0$ have the same amplitude (with different signs). Hence, the output of the measurement is a random y such that $y \cdot u = 0$. $\square$

If f is one-to-one, a similar argument shows that the output of the measurement is just a random y.

Next, we show that sufficiently many such vectors y give us enough information to recover u. If $n-1$ vectors y are linearly independent, the corresponding system of $n-1$ equations has two solutions: 0 and u. It suffices to show that $n-1$ vectors are linearly independent with a constant probability because we can run the algorithm several times, making the probability of success arbitrarily high.

Claim. With a probability at least $\frac{1}{4}$, $n-1$ random vectors such that $y \cdot u = 0$ are linearly independent.

Proof. Let $y^{(1)}, y^{(2)}, \ldots, y^{(n-1)}$ be the vectors. There are at most 2^{i-1} vectors that are linear combinations of $y^{(1)}, y^{(2)}, \ldots, y^{(i-1)}$. Hence, the probability that $y^{(i)}$ is linearly independent from $y^{(1)}, y^{(2)}, \ldots, y^{(i-1)}$ is

$$\frac{2^{n-1} - 2^{i-1}}{2^{n-1}} = 1 - \frac{1}{2^{n-i}}.$$

The probability that $y^{(1)}, y^{(2)}, \ldots, y^{(n-1)}$ all are linearly independent is just the product of these probabilities:

$$(1 - \frac{1}{2^{n-1}})(1 - \frac{1}{2^{n-2}}) \ldots (1 - \frac{1}{2}).$$

We evaluate the probability that some of $y^{(1)}, y^{(2)}, \ldots, y^{(i-2)}$ are linearly dependent. It is at most

$$\frac{1}{2^{n-1}} + \frac{1}{2^{n-2}} + \ldots + \frac{1}{4} = \frac{1}{2}.$$

Hence, y^1, $y^{(2)}$, ..., $y^{(i-2)}$ are linearly independent with probability at least $\frac{1}{2}$ and $y^{(i-1)}$ is linearly indepedent from them with probability $\frac{1}{2}$. This implies that the probability that all vectors are linearly independent is at least $\frac{1}{4}$. $\square$

By running the algorithm several times, the probability that we do not find u can be made polynomially small.

If f is one-to-one, we will get $n-1$ independent random vectors y but y will be uniformly distributed over all Z_n^2. Similarly to previous case, we will get $n-1$ independent y with high probability. Again, the system of linear equations will have two solutions: 0 and some u. However, there will be no such x that $f(x) \neq f(x+u)$. By checking $f(x) \neq f(x+u)$ we discover that u is wrong. This implies that f is one-to-one because all other possible u are ruled out by conditions $y \cdot u = 0$.

3 Lower bound for probabilistic algorithms

We prove that any probabilistic algorithm needs an exponential time to solve this problem. Further, we assume that f is two-to-one and prove a lower bound on the time needed for finding u. This proof can be easily modified, proving a lower bound for the original problem.

We first give an informal argument and then make it precise. To find u, we need to guess y and $y+u$, to compute $f(y)$ and $f(y+u)$ and to check that they are equal. The search space consisting of possible y and u is very large and, hence, this search requires lots of time.

More formally, we apply Yao's lemma:

Yao's lemma. Assume there is a probability distribution D on all possible inputs such that no deterministic algorithm running in time T gives a correct answer with probability at least p when the input is drawn from D. Then, there is no probabilistic algorithm running in time T with a probability of correct answer at least p.

The hard probability distribution is very natural. u is chosen uniformly at random from all nonzero elements of Z_n^2. This divides elements into pairs $(x, x+u)$. Pairs are mapped randomly to elements of Z_n^2 so that no two pairs are mapped to the same element.

After m steps, a deterministic algorithm has computed at most m values of f. Let these values be $f(x^{(1)})$, $f(x^{(2)})$, ..., $f(x^{(m)})$. These values give algorithm two types of information:

1. If $f(x^{(i)}) = f(x^{(j)})$ and $x^{(i)} \neq x^{(j)}$, this implies $u = x^{(i)} + x^{(j)}$.

2. If $f(x^{(i)}) \neq f(x^{(j)})$ and $x^{(i)} \neq x^{(j)}$, this implies $u \neq x^{(i)} + x^{(j)}$.

Assume that $f(x^{(1)})$, $f(x^{(2)})$, ..., $f(x^{(k)})$ are all different. Then u is none of $\binom{k}{2}$ values $f(x^{(i)}) + f(x^{(j)})$. It can be proved that all other values are equally likely.

The probability that $f(x^{(i)}) + f(x^{(k+1)}) = u$ for some $i \in \{1, \ldots, k\}$ is at most

$$\frac{k}{2^n - 1 - \binom{k}{2}}$$

because there are at least $2^n - 1 - \binom{k}{2}$ possible values of u. Taking the sum over all $k \in \{1, \ldots, m\}$, we get

$$\sum_{k=1}^{m} \frac{k}{2^n - 1 - \binom{k}{2}} \leq \sum_{k=1}^{m} \frac{k}{2^n - k^2} \leq \frac{m^2}{2^n - m^2}.$$

If $m = 2^{(1/2-\epsilon)n}$, then

$$\frac{m^2}{2^n - m^2} = \frac{2^{(1-2\epsilon)n}}{2^n - o(2^n)} = 2^{-2\epsilon n} - o(2^{-2\epsilon n}).$$

Hence, under this distribution, any deterministic algorithm running in exponential time $m = 2^{(1/2-\epsilon)n}$ has exponentially low probability of correct answer. By Yao's lemma, this implies the same bound for probabilistic algorithms.