D.2 Simon's algorithm

Simon's algorithm was first presented in 1994 and can be found in Simon, D. (1997), "On the power of quantum computation," *SIAM Journ. Computing*, 26 (5), pp. 1474–83.[9] For breaking RSA we will see that its useful to know the *period* of a function: that r such that $f(x+r) = f(x)$. Simon's problem is a warmup for this.

Simon's Problem: Suppose we are given an unknown function $f : \mathbf{2}^n \rightarrow \mathbf{2}^n$ and we are told that it is *two-to-one*. This means $f(\mathbf{x}) = f(\mathbf{y})$ iff $\mathbf{x} \oplus \mathbf{y} = \mathbf{r}$ for some fixed $\mathbf{r} \in \mathbf{2}^n$. The vector $\mathbf{r}$ can be considered the *period* of f, since $f(\mathbf{x} \oplus \mathbf{r}) = f(\mathbf{x})$. The problem is to determine the period $\mathbf{r}$ of a given unknown f.

Consider first the classical solution. Since we don't know anything about f, the best we can do is evaluate it on random inputs. If we are ever lucky enough to find $\mathbf{x}$ and $\mathbf{x}'$ such that $f(\mathbf{x}) = f(\mathbf{x}')$, then we have our answer, $\mathbf{r} = \mathbf{x} \oplus \mathbf{x}'$. After testing m values, you will have eliminated about $m(m-1)/2$ possible $\mathbf{r}$ vectors (namely, $\mathbf{x} \oplus \mathbf{x}'$ for every pair of these m vectors). You will be done when $m^2 \approx 2^n$. Therefore, on the average you need to do $2^{n/2}$ function evaluations, which is exponential in the size of the input. For $n = 100$, it would require about $2^{50} \approx 10^{15}$ evaluations. "At 10 million calls per second it would take about three years" (Mermin, 2007, p. 55). We will see that a quantum computer can determine $\mathbf{r}$ with high probability ($> 1 - 10^{-6}$) in about 120 evaluations. At 10 million calls per second, this would take about 12 microseconds!

algorithm Simon:

Input superposition: As before, start by using the Walsh-Hadamard transform to create a superposition of all possible inputs:

$$|\psi_1\rangle \stackrel{\text{def}}{=} H^{\otimes n}|0\rangle^{\otimes n} = \frac{1}{2^{n/2}} \sum_{\mathbf{x} \in \mathbf{2}^n} |\mathbf{x}\rangle.$$

[9] The following presentation follows Mermin's *Quantum Computer Science* (Mermin, 2007, §2.5, pp. 55–8).

Function evaluation: Suppose that U_f is the quantum gate array implementing f and recall $U_f|\mathbf{x}\rangle|\mathbf{y}\rangle = |\mathbf{x}\rangle|\mathbf{y} \oplus f(\mathbf{x})\rangle$. Therefore:

$$|\psi_2\rangle \stackrel{\text{def}}{=} U_f|\psi_1\rangle|0\rangle^{\otimes n} = \frac{1}{2^{n/2}} \sum_{\mathbf{x} \in \mathbf{2}^n} |\mathbf{x}\rangle|f(\mathbf{x})\rangle.$$

Therefore we have an equal superposition of corresponding input-output values.

Output measurement: Measure the output register (in the computational basis) to obtain some $|\mathbf{z}\rangle$. Since the function is two-to-one, the projection will have a superposition of two inputs:

$$\frac{1}{\sqrt{2}}(|\mathbf{x}_0\rangle + |\mathbf{x}_0 \oplus \mathbf{r}\rangle)|\mathbf{z}\rangle,$$

where $f(\mathbf{x}_0) = \mathbf{z} = f(\mathbf{x}_0 \oplus \mathbf{r})$. The information we need is contained in the input register,

$$|\psi_3\rangle \stackrel{\text{def}}{=} \frac{1}{\sqrt{2}}(|\mathbf{x}_0\rangle + |\mathbf{x}_0 \oplus \mathbf{r}\rangle),$$

but it cannot be extracted directly. If we measure it, we will get either $\mathbf{x}_0$ or $\mathbf{x}_0 \oplus \mathbf{r}$, but not both, and we need both to get $\mathbf{r}$. (We cannot make two copies, due to the no-cloning theorem.)

Suppose we apply the Walsh-Hadamard transform to this superposition:

$$\begin{aligned} H^{\otimes n}|\psi_3\rangle &= H^{\otimes n}\frac{1}{\sqrt{2}}(|\mathbf{x}_0\rangle + |\mathbf{x}_0 \oplus \mathbf{r}\rangle) \\ &= \frac{1}{\sqrt{2}}(H^{\otimes n}|\mathbf{x}_0\rangle + H^{\otimes n}|\mathbf{x}_0 \oplus \mathbf{r}\rangle). \end{aligned}$$

Now, recall (D.1.b, p. 129) that

$$H^{\otimes n}|\mathbf{x}\rangle = \frac{1}{2^{n/2}} \sum_{\mathbf{y} \in \mathbf{2}^n} (-1)^{\mathbf{x}\cdot\mathbf{y}}|\mathbf{y}\rangle.$$

(This is the general expression for the Walsh transform of a bit string. The phase depends on the number of common 1-bits.) Therefore,

$$\begin{aligned} H^{\otimes n}|\psi_3\rangle &= \frac{1}{\sqrt{2}}\left[\frac{1}{2^{n/2}} \sum_{\mathbf{y} \in \mathbf{2}^n} (-1)^{\mathbf{x}_0\cdot\mathbf{y}}|\mathbf{y}\rangle + \frac{1}{2^{n/2}} \sum_{\mathbf{y} \in \mathbf{2}^n} (-1)^{(\mathbf{x}_0+\mathbf{r})\cdot\mathbf{y}}|\mathbf{y}\rangle\right] \\ &= \frac{1}{2^{(n+1)/2}} \sum_{\mathbf{y} \in \mathbf{2}^n} \left[(-1)^{\mathbf{x}_0\cdot\mathbf{y}} + (-1)^{(\mathbf{x}_0+\mathbf{r})\cdot\mathbf{y}}\right]|\mathbf{y}\rangle. \end{aligned}$$

Note that
$$(-1)^{(\mathbf{x}_0+\mathbf{r})\cdot\mathbf{y}} = (-1)^{\mathbf{x}_0\cdot\mathbf{y}}(-1)^{\mathbf{r}\cdot\mathbf{y}}.$$
Therefore, if $\mathbf{r}\cdot\mathbf{y} = 1$, then the bracketed expression is 0 (since the terms have opposite sign and cancel). However, if $\mathbf{r}\cdot\mathbf{y} = 0$, then the bracketed expression is $2(-1)^{\mathbf{x}_0\cdot\mathbf{y}}$ (since they don't cancel). Hence the result of the Walsh-Hadamard transform is
$$|\psi_4\rangle = H^{\otimes n}|\psi_3\rangle = \frac{1}{2^{(n-1)/2}} \sum_{\mathbf{y} \text{ s.t. } \mathbf{r}\cdot\mathbf{y}=0} (-1)^{\mathbf{x}_0\cdot\mathbf{y}}|\mathbf{y}\rangle.$$

Measurement: Measuring the input register (in the computational basis) will collapse it with equal probability into a state $|\mathbf{y}^{(1)}\rangle$ such that $\mathbf{r}\cdot\mathbf{y}^{(1)} = 0$.

First equation: Since we know $\mathbf{y}^{(1)}$, this gives us some information about $\mathbf{r}$, expressed in the equation:
$$y_1^{(1)}r_1 + y_2^{(1)}r_2 + \cdots + y_n^{(1)}r_n = 0 \text{ (mod 2)}.$$

Iteration: The quantum computation can be repeated, producing a series of bit strings $\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \ldots$ such that $\mathbf{y}^{(k)}\cdot\mathbf{r} = 0$. From them we can build up a system of n linearly-independent equations and solve for $\mathbf{r}$. (If you get a linearly-dependent equation, you have to try again.) Note that each quantum step (involving one evaluation of f) produces an equation (except in the unlikely case $\mathbf{y}^{(k)} = 0$ or that it's linearly dependent), and therefore determines one of the bits in terms of the other bits. That is, each iteration reduced the candidates for $\mathbf{r}$ by approximately one-half.
□

A mathematical analysis (Mermin, 2007, App. G) shows that with $n+m$ iterations the probability of having enough information to determine $\mathbf{r}$ is $> 1 - \frac{1}{2^{m+1}}$. "Thus the odds are more than a million to one that with $n + 20$ invocations of $\mathbf{U}_f$ we will learn [$\mathbf{r}$], no matter how large n may be."

(Mermin, 2007, p. 57) Note that the "extra" evaluations are independent of n. Therefore Simon's problem can be solved in *linear* time on a quantum computer, but requires *exponential* time on a classical computer.