Simon's Algorithm: Classical Post-Processing

Andreas Klappenecker

Post-Processing: The Problem

In Simon's problem, a function $f: \{0,1\}^n \rightarrow \{0,1\}^n$ is given with the promise that there exists a bit string s in $\{0,1\}^n$ such that

$$f(x) = f(y) \text{ if and only if } x = y \text{ or } x \oplus s = y$$

Each round of the quantum algorithm yields a string y such that

$$s \cdot y = 0$$

The goal is to determine s.

Results of the Quantum Part

Case $s=0$:

- yields each string y with probability $1/2^n$.
- y is any element from F_2^n.

Case $s \neq 0$:

- yields y such that $s \cdot y = 0$ with probability $1/2^{n-1}$.
- y is an element from a $n-1$ subspace of F_2^n.

Classical Post-Processing

Let's run the quantum algorithm n-1 times. We get strings $y_1, \ldots, y_{n-1}$ satisfying the system of linear equations:

$$s \cdot y_1 = 0,\ s \cdot y_2 = 0,\ \ldots,\ s \cdot y_{n-1} = 0$$

If the y's are linearly independent, then there is a unique nonzero string s' solving this system of equations. Test whether f(0)=f(s'). If so, then s = s', else s = 0.

How likely is it that the y's are linearly independent?

Probability of Linear Independence

Let $N(s) = 2^n$ if $s=0$, and $N(s) = 2^{n-1}$ otherwise.

In other words, $N(s)$ denotes the number of possible choices for y.

y_1 is linearly independent iff it is nonzero, so the probability to obtain y_1 linearly independent is $(1-1/N(s))$.

The probability that y_1, y_2 are linearly independent is

$$(1-1/N(s))(1-2/N(s)).$$

Probability of Linear Independence

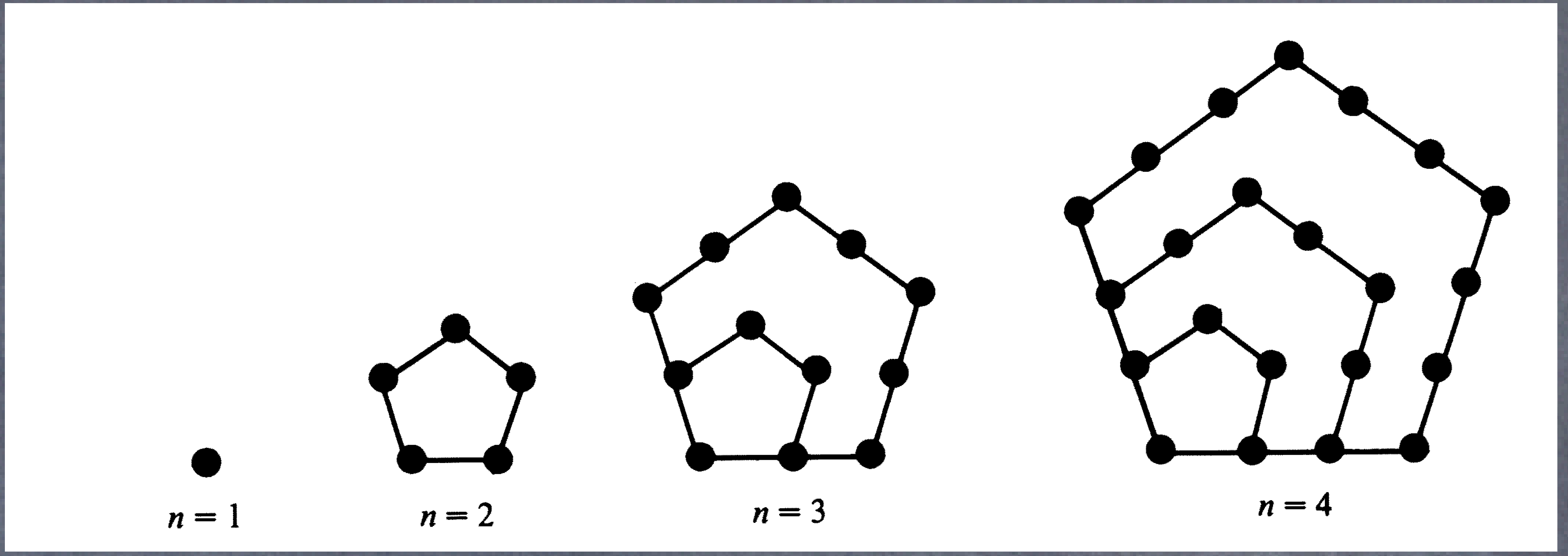
The probability that $\{ y_1, \ldots, y_{n-1} \}$ are linearly independent is given by

$$\Pr[\text{independence}] = (1-1/N(s))(1-2/N(s)) \cdots (1-2^{n-1}/N(s))$$

In other words,

$$\Pr[\text{independence}] \geqq \prod_{n=1}^{\infty} (1 - 1/2^n) \geqq \ ??$$

Pentagonal Numbers



The Pentagonal numbers are given by

1, 1+4, 1+4+7, 1+4+7+10, ...

so $w(n) = \Sigma_{k=0}^{n-1} (3k+1) = 3n(n-1)/2 + n = (3n^2-n)/2$.

Define $w(-n) = (3n^2+n)/2$. Then $w(n)$ and $w(-n)$ are the Pentagonal numbers.

Euler's Pentagonal Number Theorem

$$\prod_{k=1}^{\infty}(1-x^k) = 1 + \sum_{n=1}^{\infty}(-1)^n(x^{w(n)} + x^{w(-n)}).$$

$$= 1 - x - x^2 + x^5 + x^7 - x^{12} - x^{15} + \cdots$$

$$\prod_{k=1}^{\infty}(1-\frac{1}{2^k}) = \underbrace{1 - 1/2 - 1/4}_{=1/4} + 1/2^5 + 1/2^7 - 1/2^{12} - 1/2^{15} \cdots \geq 1/4$$

Repetitions

Suppose that we repeat this method 4m times. The chance that we do not even once end up with a set of linearly independent vectors is given by

$$\left(1 - \frac{1}{4}\right)^{4m} < e^{-m},$$

$$\text{since } (1+x) \leq e^{x}$$

Conclusions

Simon's algorithm finds the unknown string s on length n by repeating the quantum algorithm n-1 times. This yields n-1 vectors $y_1, \dots, y_{n-1}$ that are orthogonal to s.

The probability that these vectors are linearly independent is > 1/4.

Repeating the process m times leads to a probability of failure that is less than e^{-m}.

Further Reading

Read Chapter 6.5 in our textbook for an alternative approach.

Also, a different way to estimate the probabilities is given in Appendix A.3.