Shor's Algorithm

Elisa Bäumer, Jan-Grimo Sobez, Stefan Tessarini

May 15, 2015

Integer factorization

- $n = p \cdot q$ (where p, q are prime numbers) is a cryptographic one-way function
- Classical algorithm with best asymptotic behavior: General Number Field Sieve with superpolynomial scaling: $\mathrm{O}\left(\exp\left[c\,(\ln n)^{\frac{1}{3}}(\ln \ln n)^{\frac{2}{3}}\right]\right)$
- Basis for commercially important cryptography

Shor's algorithm

- Factorization algorithm with polynomial complexity
- Runs only partially on quantum computer with complexity
  $$\mathrm{O}\left((\log n)^2(\log\log n)(\log\log\log n)\right)$$
- Pre- and post-processing on a classical computer
- Makes use of reduction of factorization problem to order-finding problem
- Achieves polynomial time with efficiency of Quantum Fourier Transform

Talk outline

1. Classical computer part
    - Sketch of various subroutines
    - Reduction to period-finding problem
    - Full classical algorithm
2. Period-finding on quantum computer
    - Quantum Fourier Transform
    - Period-finding algorithm
3. Example: Factoring 21
4. Summary

Sketch of various subroutines

- greatest common divisor: e.g. Euclidean algorithm

  $$\gcd(a,b) = \begin{cases} b & \text{if } a \mod b = 0 \\ \gcd(b, a \mod b) & \text{else} \end{cases}$$

  with $a > b$, quadratic in number of digits of a, b.

  reminder: $\gcd(a,b) = 1 \rightarrow a, b$ coprime
- Test of primality: e.g. Agrawal-Kayal-Saxena 2002, polynomial
- Prime power test: determine if $n = p^\alpha$, e.g. Bernstein 1997 in $\mathrm{O}(\log n)$
- continued fraction expansion: required to approximate a rational number by an integer fraction, e.g. Hardy and Wright 1979, polynomial

Reduction to period-finding problem, Miller 1976

- Find factor of odd n provided some method to calculate the order r of $x^a \mod n$, $a \in \mathbb{N}$:

1. Choose a random $x < n$.
2. Find order r (somehow) in $x^r \equiv 1 \mod n$.
3. Compute $p, q = \gcd(x^{\frac{r}{2}} \pm 1, n)$ if r even.

- Since $(x^{\frac{r}{2}} - 1)(x^{\frac{r}{2}} + 1) = x^r - 1 \equiv 0 \mod n$.
- Fails if r odd or $x^{\frac{r}{2}} \equiv -1 \mod n$.
- Yields a factor with $p = 1 - 2^{-k+1}$ where k is the number of distinct odd prime factors of n.

Shor's algorithm

1. Determine if n is even, prime or a prime power. If so, exit.
2. Pick a random integer $x < n$ and calculate $\gcd(x, n)$. If this is not 1, then we have obtained a factor of n.
3. Quantum algorithm

   Pick q as the smallest power of 2 with $n^2 \leq q < 2n^2$.

   Find period r of $x^a \mod n$.

   Measurement gives us a variable c which has the property $\frac{c}{q} \approx \frac{d}{r}$ where $d \in \mathbb{N}$.
4. Determine d, r via continued fraction expansion algorithm.
   d, r only determined if $\gcd(d, r) = 1$ (reduced fraction).
5. If r is odd, go back to 2. If $x^{\frac{r}{2}} \equiv -1 \mod n$ go back to 2.
   Otherwise the factors $p, q = \gcd(x^{\frac{r}{2}} \pm 1, n)$.

Quantum Fourier Transform (QFT)

- Define the QFT with respect to an ONB $\{|x\rangle\} = \{|0\rangle, ..., |q-1\rangle\}$

$$QFT : |x\rangle \mapsto \frac{1}{\sqrt{q}} \sum_{y=0}^{q-1} exp\left\{\frac{2\pi i}{q} x \cdot y\right\} |y\rangle = \frac{1}{\sqrt{q}} \sum_{y=0}^{q-1} \omega^{x \cdot y} |y\rangle$$

- Apply QFT to a general state $|\psi\rangle = \sum_x \alpha_x |x\rangle$:

$$QFT(|\psi\rangle) = \frac{1}{\sqrt{q}} \sum_{y=0}^{q-1} \beta_y |y\rangle,$$

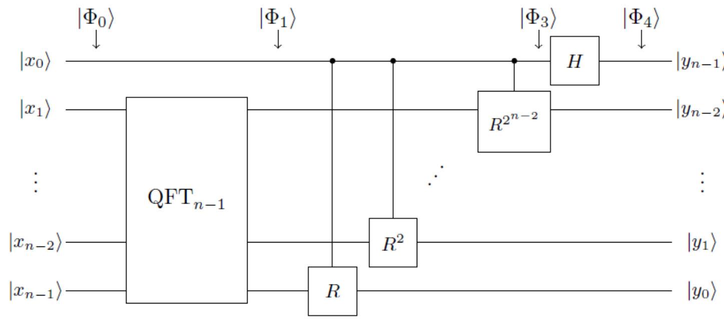
  where the β_y's are the discrete Fourier transform of the amplitudes α_x.

- The QFT is unitary, i.e.

$$QFT^\dagger QFT |x\rangle = |x\rangle$$

Quantum Fourier Transform (QFT)

- Implement QFT on n qubits



- With the matrix

$$R = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{2\pi i/N} \end{pmatrix}$$

Period Finding Algorithm

- Given a periodic function f: $\{0, ..., q-1\} \rightarrow \{0, ..., q-1\}$, where $q = 2^l$, the periodicity conditions are

$$f(a) = f(a+r) \; r \neq 0$$
$$f(a) \neq f(a+s) \; \forall s < r.$$

- Initialize the q.c. with the state $|\Phi_I\rangle = |0\rangle^{\otimes 2l}$
- Then apply Hadamard gates on the first l qubits and the identity to the others:

$$|\Phi_0\rangle = H^{\otimes l} \otimes \mathbb{1}^{\otimes l} |0\rangle^{\otimes 2l} = \left(\frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) \right)^{\otimes l} \otimes |0\rangle^{\otimes l} = \frac{1}{\sqrt{q}} \sum_{a=0}^{q-1} |a\rangle \, |0\rangle^{\otimes l}$$

- Apply the unitary that implements the function f (here it is $f = x^a \mod n$)

$$|\Phi_1\rangle = U_f \, |\Phi_0\rangle = \frac{1}{\sqrt{q}} \sum_{a=0}^{q-1} |a\rangle \, |f(a)\rangle$$

Period Finding Algorithm

- Imagine one performs a measurement on $f(a)$, then the post measurement state of the first l qubits is

$$\left|\Phi_1\right\rangle_z = \sqrt{\frac{r}{q}} \sum_{a:f(a)=z} |a\rangle \, .$$

- Remember that f is periodic and choose $a_0 = min\{a|f(a) = z\}$. Now one can rewrite

$$\left|\Phi_1\right\rangle_z = \sqrt{\frac{r}{q}} \sum_{t=0}^{q/r-1} |a_0 + t \cdot r\rangle$$

when assuming that $r|q$ (i.e. r divides q).

Period Finding Algorithm

- Perform the QFT

$$\left|\tilde{\Phi}\right\rangle_z = QFT^{-1}(\left|\Phi_1\right\rangle_z) = \sqrt{\frac{r}{q}} \sum_{t=0}^{q/r-1} \frac{1}{\sqrt{q}} \sum_{c=0}^{q-1} \exp\left\{\frac{-2\pi i}{q}(a_0 + rt)c\right\} \left|c\right\rangle$$

$$= \sqrt{\frac{r}{q^2}} \sum_{c=0}^{q-1} \exp\left\{-\frac{2\pi i}{q} a_0 c\right\} \underbrace{\sum_{t=0}^{q/r-1} \exp\left\{-\frac{2\pi i}{q} trc\right\}}_{\alpha_c} \left|c\right\rangle .$$

- Remark: if $rc = kq$ for some $k \in \mathbb{N}$ then

$$\alpha_c = \frac{q}{r}.$$

- The probability for measuring a specific $c' = kq/r$:

$$P[c'] = \left|\left\langle c' \middle| \tilde{\Phi} \right\rangle\right|^2 = \frac{r}{q^2} \left|\alpha_{c'}\right|^2 = \frac{r}{q^2} \frac{q^2}{r^2} = \frac{1}{r}$$

Period Finding Algorithm

- Overall probability to measure a c of the form $\frac{kq}{r}$ is then

$$\sum_{c=kq/r} \left| \left\langle c' \middle| \tilde{\Phi} \right\rangle \right|^2 = r\frac{1}{r} = 1$$

- The algorithm output is a natural number that is of the form $\frac{kq}{r}$, with $k \in \mathbb{N}$.

Example: Factoring n=21

1. Choose x
2. Determine q
3. Initialize first register (r_1)
4. Initialize second register (r_2)
5. QFT on first register
6. Measurement
7. Continued Fraction Expansion $\rightarrow$ determine r
8. Check r $\rightarrow$ determine factors

1. Choose a random integer x, $1 < x < n$

- if it is not coprime with n, e.g. $x = 6$:
  $\rightarrow \gcd(x, n) = \gcd(6, 21) = 3 \rightarrow 21/3 = 7 \rightarrow$ done!
- if it is coprime with n, e.g. $x = 11$:
  $\rightarrow \gcd(11, 21) = 1 \rightarrow$ continue!

2. Determine q

- $n^2 = 244 \overset{!}{\leq} q = 2^l < 2n^2 = 882$
  $\rightarrow q = 512 = 2^9$
- Initial state consisting of two registers of length l:
  $|\Phi_i\rangle = |0\rangle_{r_1} |0\rangle_{r_2} = |0\rangle^{\otimes 2^l}$

3. Initialize r_1

- initialize first register with superposition of all states $a(\mod q)$:

$$|\Phi_0\rangle = \frac{1}{\sqrt{512}} \sum_{a=0}^{511} |a\rangle\,|0\rangle$$

- this corresponds to $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ on all bits

4. Initialize r_2

- ▷ initialize second register with superposition of all states $x^a(\mod n)$:

$$
\begin{aligned}
|\Phi_1\rangle &= \frac{1}{\sqrt{512}} \sum_{a=0}^{511} |a\rangle \, |11^a(\mod 21)\rangle \\
&= \frac{1}{\sqrt{512}} (|0\rangle \, |1\rangle + |1\rangle \, |11\rangle + |2\rangle \, |16\rangle + |3\rangle \, |8\rangle + ...)
\end{aligned}
$$

a	0	1	2	3	4	5	6	7	8	9	10	...
$11^a(mod 21)$	1	11	16	8	4	2	1	11	16	8	4	...

- ▷ $r = 6$, but not yet observable

5. Quantum Fourier Transform

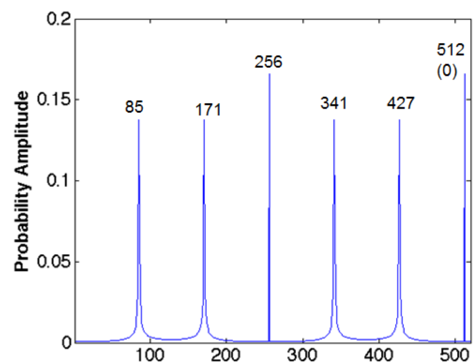
- apply the QFT on the first register:

$$|\tilde{\Phi}\rangle = \frac{1}{512} \sum_{a=0}^{511} \sum_{c=0}^{511} e^{2\pi i a c/512} |c\rangle \, |11^a (mod 21)\rangle$$

6. Measurement!

- ▷ probability for state $|c, x^k(\mod n)\rangle$, e.g. $k = 2 \to |c, 16\rangle$ to occur:

$$p(c) = \left| \frac{1}{512} \sum_{a:11^a \mod 21=16}^{511} e^{2\pi i a c/512} \right|^2 = \left| \frac{1}{512} \sum_b e^{2\pi i (6b+2)c/512} \right|^2$$



- ▷ peaks for $c = \frac{512}{6} \cdot d$, $d \in \mathbb{Z}$:

7. Determine the period r

- Assume we get 427: $\left|\frac{c}{q} - \frac{d}{r}\right| = \left|\frac{427}{512} - \frac{d}{r}\right| \overset{!}{\leq} \frac{1}{1024}$
- Continued fraction expansion:

$$\frac{c}{q} = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{\cdots}}}, \quad d_0 = a_0, \quad d_1 = 1 + a_0 a_1, \quad d_n = a_n d_{n-1} + d_{n-2}$$

$$r_0 = 1, \qquad r_1 = a_1, \qquad r_n = a_n r_{n-1} + r_{n-2}$$

$$\frac{427}{512} = 0 + \frac{1}{1 + \frac{1}{5 + \frac{1}{42 + \frac{1}{2}}}}, \quad d_0 = 0, \quad d_1 = 1, \quad d_2 = 5, \quad d_3 = 427$$

$$r_0 = 1, \quad r_1 = 1, \quad r_2 = 6, \quad r_3 = 512$$

- as $\frac{d_0}{r_0} = 0$ and $\frac{d_1}{r_1} = 1$ obviously don't work, try $\frac{d_2}{r_2} = \frac{5}{6} \rightarrow r = 6$
  $\rightarrow$ it works! $=$)

- for $\frac{c}{q} = \frac{171}{512}$ we would get $\frac{d}{r} = \frac{1}{3}$, so using $r = 3$ this would not work.
  $\rightarrow$ it only works if d and r are coprime!
  $\rightarrow$ if it doesn't work, try again!

8. Check r

- check if r is even ✓
- check if $x^{r/2} \mod n \neq -1$ ✓
- as both holds, we can determine the factors:

$$x^{r/2} \mod n - 1 = 11^3 \mod 21 - 1 = 7$$
$$x^{r/2} \mod n + 1 = 11^3 \mod 21 + 1 = 9$$

$\rightarrow$ the two factors are $\gcd(7, 21) = 7$ and $\gcd(9, 21) = 3$

Conclusion

- Shor's algorithm is very important for cryptography, as it can factor large numbers much faster than classical algorithms (polynomial instead of exponential)
- powerful motivator for quantum computers
- no practical use yet, as it is not possible yet to design quantum computers that are large enough to factor big numbers

References

- ▷ Shor, Peter W. "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer." *SIAM journal on computing* 26.5 (1997): 1484-1509.
- ▷ Agrawal, Manindra, Neeraj Kayal, and Nitin Saxena. "PRIMES is in P." *Annals of mathematics* (2004): 781-793.
- ▷ Bernstein, Daniel. "Detecting perfect powers in essentially linear time." *Mathematics of Computation of the American Mathematical Society* 67.223 (1998): 1253-1283.
- ▷ Hardy, Godfrey Harold, et al. *An introduction to the theory of numbers*. Vol. 4. Oxford: Clarendon press, 1979.

▷ Miller, Gary L. "Riemann's hypothesis and tests for primality." *Journal of computer and system sciences* 13.3 (1976): 300-317.