Quantum Computing

Jim Royer

CIS 428/628: Introduction to Cryptography

September 5, 2021

⋮

References

- *A Physics-Free Introduction to the Quantum Computation Model* by Stephen A. Fenner. https://arxiv.org/abs/cs/0304008
  (. . . more importantly, it is complex analysis free)
- *The Talk* by Scott Aaronson and Zach Weinersmith, http://www.smbc-comics.com/comic/the-talk-3
  (There is tons of misleading hype about quantum computing. This is a good, double-entendre-filled, dehyping.)
- *Quantum Computing Since Democritus* by Scott Aaronson https://www.scottaaronson.com/democritus/
  (This connects quantum computing to the wider intellectual world while being rather goofy.)

Quantum Computing and Cryptography

- Given RSA with key size k,
  it can be broken by a computer with quantum register size $\approx k$.*
- Similarly with discrete-log-based cryptosystems.
- There are latticed-based cryptosystems that quantum computers seemingly cannot do better than classical computers in breaking.
- We will cover enough about quantum computing give you a *glimpse* of what is behind all the fuss.
- This is based on *A Physics-Free Introduction to the Quantum Computation Model* by Stephen A. Fenner. https://arxiv.org/abs/cs/0304008.

**Assuming that you can build a quantum computer of that size.*

Classical Boolean Circuits, I

We view them as naming maps $\{0,1\}^n \to \{0,1\}^n$

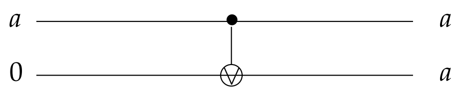
Consider We can describe this by either of:

- $b \leftarrow a \wedge b;\quad a \leftarrow \neg a;\quad b \leftarrow b \vee c$ $\qquad |x,y,z\rangle$ = *state vector*
- $|a,b,c\rangle \;\mapsto\; |a, a\wedge b, c\rangle \;\mapsto\; |\neg a, a\wedge b, c\rangle \;\mapsto\; |\neg a, (a\wedge b)\vee c, c\rangle$

Classical Boolean Circuits, II

Input/Output Conventions

- The first k registers are input $\qquad 0 \le k \le n$
- The first ℓ registers are output $\qquad 0 \le \ell \le n$
- Each non-input register is assigned 0 or 1



$a \mapsto (a, a)$

Uniform Computation

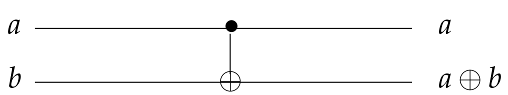
- A *circuit family*, $\mathcal{C}$, is a sequence of circuits $C_0, C_1, C_2, \ldots \ni$
  for each i, C_i has i-inputs and 1-output.
- $L(\mathcal{C}) =_{\text{def}} \{\, w \mid |w| = n \;\&\; C_n(w) = 1 \,\}$, the *language defined by* $\mathcal{C}$.
- A circuit family is *ptime uniform* $\iff$
  $\exists$ a poly-time alg $D \ni$
  for all i,
  $D(\underbrace{1\ldots1}_{i \text{ many}}) =$ a description of C_i.

FACT: $P =$ the languages accepted by ptime uniform circuit families.

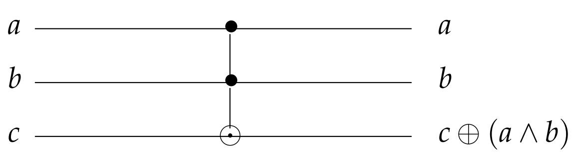
Reversible Circuits, I

Reversible circuits have inverses.

The controlled not gate (CNOT)



Toffoli Gate where $\odot(x, y, z) = z \oplus (x \wedge y)$



Reversible circuits do not collapse states. *(Why?)*

Reversible Circuits, II

CNOT Gate

input		*output*	
0	0	0	0
0	1	0	1
1	$\boxed{0}$	1	$\boxed{1}$
1	$\boxed{1}$	1	$\boxed{0}$

Toffoli Gate

input			*output*		
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	$\boxed{0}$	1	1	$\boxed{1}$
1	1	$\boxed{1}$	1	1	$\boxed{0}$

$\boxed{0}$ and $\boxed{1}$ are the *interesting* bits.

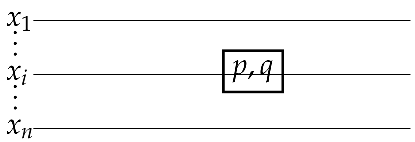
Probabilistic Circuits, I

The Biased Coin-Flip Gate $-\boxed{p,q}-$

input	output	
0	$0{:}p$	$1{:}(1-p)$
1	$0{:}q$	$1{:}(1-q)$

$|\vec{v}\rangle$: 2^n basis vectors

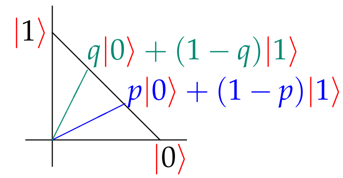
$\mathcal{H}$: a 2^n-dim. real vector space
($\mathcal{H}$ for Hilbert space)



$$
\begin{aligned}
|x_{1..i-1}, 0, x_{i+1..n}\rangle &\mapsto p \cdot |x_{1..i-1}, 0, x_{i+1..n}\rangle + (1-p) \cdot |x_{1..i-1}, 1, x_{i+1..n}\rangle \\
|x_{1..i-1}, 1, x_{i+1..n}\rangle &\mapsto q \cdot |x_{1..i-1}, 0, x_{i+1..n}\rangle + (1-q) \cdot |x_{1..i-1}, 1, x_{i+1..n}\rangle
\end{aligned}
$$

Probabilistic Circuits, II

Consider the subspace spanned by $|0\rangle$ and $|1\rangle$.



The gate $\boxed{p,q}$ always maps the line segment from (1,0) to (0,1) to itself.

We can also represent the $\boxed{p,q}$ gate by the matrix:

$$
\begin{bmatrix} p & q \\ 1-p & 1-q \end{bmatrix}
$$

This is a *stochastic matrix*: all entries ≥ 0, all columns sum to 1.

Probabilistic Circuits: *Gates as Linear Maps*

The irreversible AND gate is:

a	b	a	$a \wedge b$
0	0	0	0
0	1	0	0
1	0	1	0
1	1	1	1

$a\,b$	00	01	10	11
00	1	1	0	0
01	0	0	0	0
10	0	0	1	0
11	0	0	0	1

- All entries are 0–1
- One 1 in each col
- $\therefore$ Stochastic

Reversible gates are permutation matrices! *(Why?)*

Definition

A *probabilistic circuit* is a circuit built from Boolean & $\boxed{p,q}$ gates, where

- The input state is a basis state.
- The output state is of the form: $\sum_{x \in \{0,1\}^n} p_x |x\rangle \quad \ni$
  (i) each $p_x \geq 0$ and (ii) $\sum |p_x| = 1$.

p_x = the probability that the output will be $|x\rangle$.

"Majority Coin Flips" Circuit

$\boxed{\frac{1}{2},\frac{1}{2}}$ = flip of a fair coin

A Complexity-Theoretic Aside

- $\vec{C} = C_0, C_1, C_2, \ldots$: a ptime uniform probablistic circuit family
- (R, A) is an *acceptance criterion* when $R, A \subset [0,1]$ with $R \cap A = \emptyset$.
  (R for reject, A for accept)
- $\vec{C}$ *computes L with acceptance criterion* (R, A) when
  for each n and each $x \in \{\, \mathbf{0}, \mathbf{1} \,\}^n$:

$$\begin{aligned} x \in L &\implies Prob[C_n(x) = 1] \in A \\ x \notin L &\implies Prob[C_n(x) = 1] \in R \end{aligned}$$

Class	Acceptance Criterion	
P	$(\{\,0\,\}, \{\,1\,\})$	
NP	$(\{\,0\,\}, (0,1])$	
RP	$(\{\,0\,\}, (\frac{1}{2}, 1])$	
BPP	$([0,q], [1-q, 1])$	where $0 < q < \frac{1}{2}$
PP	$([0, \frac{1}{2}], (\frac{1}{2}, 1])$	

Quantum Circuits (á la Fenner), I

- states = vectors in $\mathcal{H}$ gates = matrices
- Now allow nonegative entries in matrices. (But all real numbers)
- Now require: $\|Mv\|_2 = \|v\|_2$ for all v.
- Note: $\|\vec{a}\|_2 =_{\text{def}} \sqrt{a_1^2 + \cdots + a_n^2}$
- This forces the matrices to be *orthonormal*,
  i.e., its columns form an orthogonal basis of $\mathcal{H}$.
- Registers are now called *qubits* (quantum bits) instead of bits.
- The *Hadamard gate*, $-\boxed{H}-$, has the matrix: $\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ See the next slide
  $H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. $H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. **Note:** $H^2 = I$.
- Fact: $\{\, H, \text{Toffoli gates} \,\}$ are a *universal* collection of quantum gates.
- The $\boxed{p, q}$ gates now correspond to *measurements*.

Hadamard Gate Geometrically

aa

1. Transpose around the x-axis: $(x, y) \mapsto (x, -y)$
2. Then do a $+45^o$ rotation.

$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. $H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$.

Quantum Circuits (á la Fenner), II

QCF (Quantum Coin Flip)

This is a variation on Hadamard gate.

$$\text{QCF} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$$

Note that $(\text{QCF})^2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ = the not gate.

So, QCF = $\sqrt{\text{NOT}}$, *the square root of not*.

Quantum I/O

Input: basis states
Output: $\sum_{x \in \{\,0,1\,\}^n} a_x |x\rangle$ **Note:** $\sum a_x^2 = 1$
a_x^2 = the probability associated with $|x\rangle$
a_x = the *probability amplitude* for $|x\rangle$

Another Complexity-Theoretic Aside

If we use quantum circuits, then

Class	Description	Acceptance Criterion
EQP	Exact quantum polynomial time	$(\{0\}, \{1\})$
$C_{\neq}P$	Co-Exact-Counting Polynomial-Time	$(\{0\}, (0,1])$
RQP	One-sided Error Extension of EQP	$(\{0\}, (\frac{1}{2}, 1])$
BQP	Bounded-Error Quantum Polynomial-Time	$([0, \frac{1}{n}), (\frac{n-1}{n}, 1])$
PP	Probabilistic Polynomial-Time	$[0, \frac{1}{2}], (\frac{1}{2}, 1])$

See: https://complexityzoo.uwaterloo.ca/Complexity_Zoo

“Traditional” Quantum Circuits

- In place of vector spaces over $\mathbb{R}$, we use v.s.’s over $\mathbb{C}$.
- In place of orthonormal matrices, we use unitary matrices.
- Etc., etc. See §6 of Fenner for details.

- Past this point, we shall be even sketchier than before.
- …so, we won’t digress into complex linear algebra.

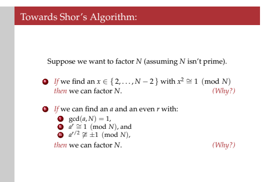
Towards Shor’s Algorithm:

Suppose we want to factor N (assuming N isn’t prime).

(a) *If* we find an $x \in \{2, \ldots, N-2\}$ with $x^2 \cong 1 \pmod{N}$ *then* we can factor N. *(Why?)*

(b) *If* we can find an a and an even r with:
  - (i) $\gcd(a, N) = 1$,
  - (ii) $a^r \cong 1 \pmod{N}$, and
  - (iii) $a^{r/2} \ncong \pm 1 \pmod{N}$,

  then we can factor N. *(Why?)*



(a) Suppose $1 < x < N-1$ and $x^2 \cong 1 \pmod{N}$.
Then $N | (x^2 - 1)$, i.e, $N | (x-1)(x+1)$.
Since $1 < x < N-1$, neither $x - 1 = 0$ nor $x + 1 = n$.
So $\gcd(N, x-1) > 1$ or $\gcd(N, x+1) > 1$.

(b) Use (a).

Towards Shor's Algorithm:

Heuristic Procedure for Factoring

Input N.
Pick $a \overset{\text{ran}}{\in} \{2,\ldots,N-2\}$.
If $\gcd(a,N) > 1$, return $\gcd(a,N)$. (* It is a (nontrivial) factor *)
(* So, $\gcd(a,N) = 1$ *)
Find the *smallest* $r > 0$ with $a^r \cong 1 \pmod{N}$. (* Expensive classically *)
If r is odd or $a^{r/2} \cong -1 \pmod{N}$,
then: return *FAILURE*
else: use the trick of the previous page to compute a factor of N
return this factor.

- *FACT:* If $N = p_1^{k_1} \ldots p_s^{k_s}$ where $p_1, \ldots, p_s$ are distinct primes and $s > 1$, then $Prob[\text{the procedure succeeds on } N] \geq 1 - \frac{1}{2^{s-1}} \geq \frac{1}{2}$.
- So repeating the procedure on N not too many times will find us a factor (with high probability).
- *BUT* the best know *classical* methods for finding r are exponential time.

Peter Shor's Clever Idea (One of Many)

Heuristic Procedure for Factoring

Input N.
Pick $a \overset{\text{ran}}{\in} \{2,\ldots,N-2\}$.
If $\gcd(a,N) > 1$, return $\gcd(a,N)$.
Find the *smallest* $r > 0$ with $a^r \cong 1 \pmod{N}$. (* PROBLEM *)
If r is odd or $a^{r/2} \cong -1 \pmod{N}$,
then: return *FAILURE*
else: compute a factor of N and return it

Use QC to find r. That is:

- Consider $1, a^1, a^2, a^3, \ldots \pmod{n}$.
- If $a^r \equiv 1 \pmod{n}$, then the sequence repeats every r times.

$\therefore$ Finding the period of the sequence, finds r.

- In signal processing, *Fourier transforms* are used to find periods.

Quantum Fourier Transform

$$\text{QFT}(|x\rangle) =_{\text{def}} \frac{1}{\sqrt{2^m}} \sum_{c \in \{0,1\}^m} e^{\frac{2\pi i x c}{2^m}} |c\rangle$$

- This can be realized as a quantum circuit.
- We'll come back to the properties of this thing shortly.

Shor's Factoring Algorithm, I

$|0\ldots0,0\ldots0\rangle$ $m+n$ long

$\downarrow$

$\frac{1}{\sqrt{2}}(|00\ldots0,0\ldots0\rangle + |10\ldots0,0\ldots0\rangle)$

$\downarrow$

$\vdots$

$\downarrow$

$\frac{1}{\sqrt{2^m}} \sum_{c \in \{0,1\}^m} |c, \vec{0}\rangle$ superimposition of 2^m states

$\downarrow$

$\frac{1}{\sqrt{2^m}} \sum_{c \in \{0,1\}^m} |c, a^c \bmod n \ldots\rangle$

$\downarrow$

QFT(—) Now what???

Shor's Factoring Algorithm, II

- When you measure $\sum_i a_i |x_i\rangle$
  you get state $|x_i\rangle$ with probability a_i^2.
- Thanks to QFT,
  states near the period have pretty high probability.
- $\therefore$ Measure, test, and refine.
  See: *Shor's Quantum Factoring Algorithm* by Samuel J. Lomonaco,
  https://arxiv.org/abs/quant-ph/0010034
- A similar trick (using QFT) can compute discrete logs.

Quantum Algorithms Beyond Shor's

Grover's Algorithm

- Suppose that $C : \{0,1\}^n \to \{0,1\}$ is such that $C(s) = 1$ for only one $s \in \{0,1\}^n$.
- Classically, finding this s takes $\Theta(2^m)$ time.
- Using QFT trickery, one can do this in $\Theta(\sqrt{2^m})$ time.
- This is the best known quantum algorithm besides Shor's.

- For other quantum algorithms, see:
  https://en.wikipedia.org/wiki/Quantum_algorithm
- The take away is that quantum computers *are* magic bullets,
  but only for some fairly special problems.
- As factoring and discrete-log are among these special problems,
  Cryptography must respond, *e.g., lattice-based cryptosystems*.