Introduction to Quantum Algorithms

This is a collection of slides for lectures based on my book "Introduction to Quantum Algorithms," which has been published as part of the AMS Undergraduate Texts in Pure and Applied Mathematics series, number 64.
You can choose the slides that fit your lecture from this collection.
I am happy to integrate additional slides that you have created yourself. Please send these and any other comments and suggestions about the book to johannes.buchmann@tu-darmstadt.de.
Many thanks. Johannes Buchmann

Version: March 21, 2024

1. Classical Computation

1.1. Deterministic algorithms

Binary expansion of integers

Proposition

Let $a \in \mathbb{N}$. Then there is a uniquely determined sequence $\vec{b} = (b_0, \ldots, b_{n-1}) \in \{0,1\}^*$ with $b_0 = 1$ such that

$$a = \sum_{i=0}^{n-1} b_i 2^{n-i-1}. \tag{1.1}$$

The length of this sequence is $n = \lfloor \log_2 a \rfloor + 1$.

Definition

1. $\vec{b} = (b_0, \ldots, b_{n-1})$: *binary expansion* or *binary representation* of a.
2. $\mathrm{bitLength}(a) = \lfloor \log_2 a \rfloor + 1$: *bit length* or *binary length* of a.
3. $\mathrm{stringToInt}(\vec{b}) = \sum_{i=0}^{n-1} b_i 2^{n-i-1}$.

Deterministic algorithms and pseudocode

Example: the Euclidean algorithm

Input: $a, b \in \mathbb{Z}$
Output: $\gcd(a, b)$

```
1:  gcd(a, b)
2:      a ← |a|
3:      b ← |b|
4:      while b ≠ 0 do
5:          r ← a mod b
6:          a ← b
7:          b ← r
8:      end while
9:      return a
10: end
```

Data types and their sizes

Elementary data types:

Elementary data type	Element	Size
$\{0,1\}$	b	$\text{size}(b) = \mathrm{O}(1)$
$\mathcal{R} = \{\text{a}, \ldots, \text{z}, \text{A}, \ldots, \text{Z}, \text{␣}\}$	x	$\text{size}(x) = \mathrm{O}(1)$
$\mathbb{Z}$	a	$\text{size}(a) = \mathrm{O}(\text{bitLength}(\lvert a \rvert))$

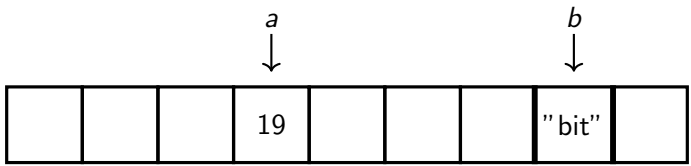
In addition: the sets of vectors and matrices over a data type.

If $\vec{s} = (s_0, \ldots, s_{k-1})$, then $\text{size}(\vec{s}) = \mathrm{O}\left(\sum_{i=0}^{k-1} \text{size}\, s_i\right)$.

If $M = (m_{i,j})$, then $\text{size}(M) = \mathrm{O}\left(\sum \text{size}\, m_{i,j}\right)$.

Variables

Variable: a symbol that serves as a reference to a memory unit with the capability to store an element of a specific data type.



Operations on integers

Operation	Operands	Result	Running time	Space
absolute value	$a \in \mathbb{Z}$	$\lvert a \rvert$	$O(n)$	$O(n)$
add	$a, b \in \mathbb{Z}$	$a + b$	$O(n)$	$O(n)$
subtract	$a, b \in \mathbb{Z}$	$a - b$	$O(n)$	$O(n)$
multiply	$a, b \in \mathbb{Z}$	$a * b$	$O(n^2)$	$O(n)$
divide	$a, b \in \mathbb{Z},\ b \neq 0$	$\lfloor a/b \rfloor$	$O(n^2)$	$O(n)$
remainder	$a, b \in \mathbb{Z},\ b \neq 0$	$a \bmod b$	$O(n^2)$	$O(n)$
floor	$a, b \in \mathbb{Z},\ b \neq 0$	$\lfloor a/b \rfloor$	$O(n^2)$	$O(n)$
ceiling	$a, b \in \mathbb{Z},\ b \neq 0$	$\lceil a/b \rceil$	$O(n^2)$	$O(n)$
square root	$a \in \mathbb{Z}$	$\lfloor \sqrt{a} \rfloor$	$O(n^2)$	$O(n)$
equal	$a, b \in \mathbb{Z}$	$a = b$	$O(n)$	$O(n)$
less than	$a, b \in \mathbb{Z}$	$a < b$	$O(n)$	$O(n)$
less than or equal to	$a, b \in \mathbb{Z}$	$a \leq b$	$O(n)$	$O(n)$

Logic operations

		AND	OR	NOT	NAND	NOR	XOR
a	b	$a \wedge b$	$a \vee b$	$\neg a$	$a \uparrow b$	$a \downarrow b$	$a \oplus b$
0	0	0	0	1	1	1	0
0	1	0	1	1	1	0	1
1	0	0	1	0	1	0	1
1	1	1	1	0	0	0	0

Time and space requirements: $\mathrm{O}(1)$.

Assignments and return statement

$$a \leftarrow b, \quad \textbf{return}(b)$$

Time and space requirements: $\mathrm{O}(\text{size}(b))$.

$$c \leftarrow c + 1, \quad c \leftarrow A(x), \quad \textbf{return}(2 * a + b)$$

Time and space requirements: Time and space to evaluate right-hand side.

For, while, and repeat statements

```
p ← 1
for i = 1 to e do
    p ← 2p
end for
```

```
i ← 1
p ← 1
while i ≤ e do
    p ← 2p
    i ← i + 1
end while
```

```
i ← 0
p ← 1
repeat
    p ← 2p
    i ← i + 1
until i = e
```

Time and space requirement:
Time and space to evaluate the conditions and carry out the loops.

If statements

if $a < 0$ **then**
    $a \leftarrow -a$
end if

if $a \bmod 11 = 0$ **then**
    $s \leftarrow 1$
else
    $s \leftarrow 0$
end if

if $a > 0$ **then**
    $s \leftarrow 1$
else if $a = 0$ **then**
    $s \leftarrow 0$
else if $a < 0$ **then**
    $s \leftarrow -1$
end if

Time and space requirement:
Time and space to evaluate the conditions and carry out the assignments.

Definition of algorithms

An algorithm A has the following components:

1. An **Input** statement. It specifies a finite number of *input variables*, their data types, and the permitted values of these variables. The set of all permitted input value tuples is referred to as $\mathrm{Input}(A)$.
2. An **Output** statement. For every $a \in \mathrm{Input}(A)$ it specifies that may depend on a makes a return value a correct output. The set of all correct outputs for input a is denoted by $\mathrm{Output}(A, a)$.
3. An algorithm name followed by the sequence of input variables, which is used when A is called by other algorithms as a subroutine.
4. A finite sequence of instructions that ends with **end**.

Defining properties of algorithms

1. Each run of the algorithm with a permitted input carries out a **return** instruction. This means that the algorithm terminates on any input $a \in \mathrm{Input}(A)$.
2. When the algorithm performs a **return** instruction, the return value is correct, i.e., it has the property specified in the **Output** statement.
3. Executing the **return** instruction is the only way the algorithm can terminate. This means that after executing a statement that is not a **return** instruction, there is always a next instruction that the algorithm carries out.

Run of an algorithm I

Example: Beginning of the run of the Euclidean algorithm.

State#	Memory contents			Next instruction
	a	b	r	
1	100	35		$a \leftarrow \lvert a \rvert$
2	100	35		$b \leftarrow \lvert b \rvert$
3	100	35		**while** $b \neq 0$ **do**
4	100	35		$r \leftarrow a \bmod b$
5	100	35	30	$a \leftarrow b$
6	35	35	30	$b \leftarrow r$
7	35	30	30	**end while**
8	35	30	30	**while** $b \neq 0$ **do**

Run of an algorithm II

Example: End of the run of the Euclidean algorithm

State#	Memory contents			Next instruction
	a	b	r	
1	30	5	5	**while** $b \neq 0$ **do**
2	30	5	5	$r \leftarrow a \bmod b$
3	30	5	0	$a \leftarrow b$
4	5	5	0	$b \leftarrow r$
5	5	0	0	**end while**
6	5	0	0	**while** $b \neq 0$ **do**
7	5	0	0	**return** a

Deterministic factoring algorithm

Input: $a \in \mathbb{Z}_{>1}$
Output: A proper divisor b of a if a is composite, or 0 if a is a prime number

```
1: detFactor(a)
2:     for all b = 2, ..., ⌊√a⌋ do
3:         if a mod b = 0 then
4:             return b
5:         end if
6:     end for
7:     return 0
8: end
```

Definition of decision algorithms

1. A decision algorithm decides whether a string $s \in \{0,1\}^*$ belongs to a subset L of which is called a *language*.
2. Input: $\vec{s} \in \{0,1\}^*$.
3. Output: 0 or 1:
   - 1 means that $\vec{s} \in L$.
   - 0 means that $\vec{s} \in \{0,1\}^* \setminus L$.
4. Short: The algorithm *decides the language* L.

Compositeness decision algorithm

Input: $\vec{s} \in \{0,1\}^*$
Output: 1 if $\mathrm{stringToInt}(\vec{s})$ is composite and 0 otherwise

```
1: decideComp(s⃗)
2:     a ← stringToInt(s⃗)
3:     for all b = 2, ..., ⌊√a⌋ do
4:         if a mod b = 0 then
5:             return 1
6:         end if
7:     end for
8:     return 0
9: end
```

Definition of the time complexity of a deterministic algorithm

Let A be a deterministic algorithm

1. The *running time* or *time complexity* of A for a particular input $a \in \mathrm{Input}(A)$ is the sum of the time required for reading the a which is $\mathrm{O}(\mathrm{size}(a))$ and the running times of the instructions executed during the algorithm run with input a.
2. The *worst-case running time* or *worst-case time complexity* of A is the function

$$\mathrm{wTime}_A : \mathbb{N} \to \mathbb{R}_{\geq 0}$$

that sends a positive integer n which is the size of an input of A to the maximum running time of A over all inputs of size n. If n is not the size of an input of a, then we set $\mathrm{wTime}_A(n) = 0$.

Definition of the time complexity of a deterministic algorithm

Let A be a deterministic algorithm

1. The *space complexity* of A for a particular input a is the total amount of memory space that is used in the algorithm run with input a.
2. The *worst-case space complexity* of A is the function

$$\mathrm{wSpace}_A : \mathbb{N} \to \mathbb{N}$$

that sends a positive integer n which is the size of an input of A to the maximum space complexity of A over all inputs of size n. If n is not the size of an input of a, then we set $\mathrm{wSpace}_A(n) = 0$.

Definition of worst-case complexity of a deterministic algorithm

Let $f : \mathbb{N} \to \mathbb{N}$ be a function.

- We say that A has *asymptotic worst-case running time or space complexity* $\mathrm{O}(f)$ if $\mathrm{wTime}_A = \mathrm{O}(f)$ or $\mathrm{wSpace}_A = \mathrm{O}(f)$, respectively.
- The words "asymptotic" and "worst-case" may also be omitted.

Complexity classes

Name	Time or space complexity
constant	$\mathrm{O}(1)$
logarithmic	$\mathrm{O}(\log n)$
linear	$\mathrm{O}(n)$
quasilinear	$\mathrm{O}(n(\log n)^c)$ for some $c \in \mathbb{N}$
quadratic	$\mathrm{O}(n^2)$
cubic	$\mathrm{O}(n^3)$
polynomial	$\mathrm{O}(n^c)$ for some $c \in \mathbb{N}$
subexponential	$\mathrm{O}(2^{n^\epsilon})$ for all $\epsilon \in \mathbb{R}_{>0}$
exponential	$\mathrm{O}(2^{n^c})$ for some $c \in \mathbb{N}$

1.2. Probabilistic algorithms

Defining properties of probabilistic algorithms

The differences between a probabilistic algorithm A and a deterministic algorithm are the following.

1. A may call the subroutine $\mathrm{coinToss}$ which returns 0 or 1, both with probability $\frac{1}{2}$.
2. A may call other probabilistic algorithms as subroutines if given a permitted input, they terminate and return one of finitely many possible outputs according to a probability distribution.
3. A's run of on input of $a \in \mathrm{Input}(A)$ may depend on a and the return values of the probabilistic subroutines called during the run of the algorithm.
4. A may not terminate, since termination may depend on certain return values of some probabilistic subroutine that may never occur.
5. If A terminates on input of $a \in \mathrm{Input}(A)$ with output o, then o may not be uniquely determined by a. Also, we may have $o \in \mathrm{Output}(A, a)$ or $o =$ "Failure".
6. "Failure" must never be a correct output.

Success and error-freeness

If a probabilistic algorithm A returns an output from $\mathrm{Output}(A, a)$ for a specific input $a \in \mathrm{Input}(A)$, we refer to the corresponding algorithm run as a "success". Otherwise, it is considered a "failure".

A probabilistic algorithm that, upon termination, always returns a correct result or "Failure" is called "error-free".

Types of probabilistic algorithms

1. *Monte Carlo algorithms*. They always terminate, but may not always be successful.
2. *Las Vegas algorithms*. They may not terminate, but when they terminate, they are successful.

Monte Carlo algorithm randomString

Selecting a uniformly distributed random bit string of fixed length.

Input: $k \in \mathbb{N}$
Output: $s \in \{0,1\}^k$

```
1: randomString(k)
2:     for i = 0 to k − 1 do
3:         s_i ← coinToss                ▷ Basic probabilistic operation
4:     end for
5:     return s⃗ = (s_0, ..., s_{k−1})
6: end
```

Monte Carlo Algorithm $\mathrm{randomInt}$

Selecting a uniformly distributed random positive integer of bounded bit length.

Input: $k \in \mathbb{N}$
Output: $b \in \mathbb{N}_0$ with $\mathrm{bitLength}(b) \leq k$

1: $\mathrm{randomInt}(k)$
2: $\quad \vec{s} \leftarrow \mathrm{randomString}(k)$
3: $\quad b \leftarrow \mathrm{stringToInt}(\vec{s})$
4: $\quad$ **return** b
5: **end**

Monte Carlo factoring algorithm

Input: $a \in \mathbb{N}_{>1}$
Output: A proper divisor $b \in \mathbb{N}$ of a

```
1: mcFactor(a)
2:     b ← randomInt(⌈(bitLength a)/2⌉)
3:     if 1 < b < a ∧ a mod b = 0 then
4:         return b
5:     end if
6:     return "Failure"
7: end
```

Las Vegas factoring algorithm

Input: $a \in \mathbb{N}_{>1}$
Output: A proper divisor $b \in \mathbb{N}$

```
1: lvFactor(a)
2:     repeat
3:         b ← mcFactor(a)
4:     until b ≠ "Failure"
5:     return b
6: end
```

Bernoulli algorithms

Bernoulli algorithm associated with an error-free Monte Carlo algorithm A.

Input: $a \in \mathrm{Input}(A)$
Output: $b \in \mathrm{Output}(A, a)$

```
1: bernoulli_A(a)
2:     b ← "Failure"
3:     while b = "Failure" do
4:         b ← A(a)
5:     end while
6:     return b
7: end
```

Probabilistic decision algorithms

Their input set is $\{0,1\}^*$. They always return 0 or 1. But the answer may be wrong.

Types of probabilistic decision algorithms A:

1. A is called *true-biased* if it never returns *false positives*. So, if on input of $\vec{s} \in \{0,1\}^*$ the algorithm returns 1, then $\vec{s} \in L$.
2. A is called *false-biased* if it never returns *false negatives*. So, if at the input of $\vec{s} \in \{0,1\}^*$ the algorithm returns 0, then $\vec{s} \notin L$.
3. If A is true-biased or false-biased, then it is also called an *algorithm with one-sided error*.
4. A is called an *algorithm with two-sided error* if it can return false positives and false negatives.

True-biased Monte Carlo compositeness decision algorithm

Input: $\vec{s} \in \{0,1\}^*$
Output: 1 if $\mathrm{stringToInt}(\vec{s})$ is composite and 0 otherwise

```
1: mcComposite(s⃗)
2:     a ← stringToInt(s⃗)
3:     b ← mcFactor(a)
4:     if b ∈ ℕ then
5:         return 1
6:     else
7:         return 0
8:     end if
9: end
```

Monte Carlo compositeness decision algorithm with two-sided error

Input: $\vec{s} \in \{0,1\}^*$
Output: 1 if $\mathrm{stringToInt}(\vec{s})$ is composite and 0 otherwise

```
1:  mcComposite2(a)
2:      a ← stringToInt(s⃗)
3:      c ← coinToss
4:      b ← mcFactor(a)
5:      if c = 1 ∨ b ∈ ℕ then
6:          return 1
7:      else
8:          return 0
9:      end if
10: end
```

1.3. Analysis of probabilistic algorithms

Random sequences of probabilistic algorithms

Let A be a probabilistic algorithm and let $a \in \mathrm{Input}(A)$.

1. $\mathrm{Rand}(A, a)$: The set of all random sequences of runs of A with input a.
2. $\mathrm{FRand}(A, a)$: The set of all finite sequences in $\mathrm{Rand}(A, a)$.
3. The run of A with input a corresponding to $\vec{r} \in \mathrm{Rand}(A, a)$ terminates if and only if $\vec{r} \in \mathrm{FRand}(A, a)$. The output of A after this run: $A(a, \vec{r})$.
4. If $\vec{r} = (r_0, \ldots, r_{k-1})$ is a prefix of an element in $\mathrm{Rand}(A, a)$ and p_i is the probability of r_i, then we write $\Pr_{A,a}(\vec{r}) = \prod_{i=0}^{k-1} p_i$.

Example

Let $A = \mathrm{randomString}$ and $a \in \mathrm{Input}(A) = \mathbb{N}$.

1. $\mathrm{Rand}(A) = \mathrm{FRand}(A) = \{0, 1\}^a$.
2. If $\vec{r} = (r_0, \ldots, r_{k-1})$ is a prefix of an element in $\mathrm{Rand}(A, a)$, then $\Pr_{A,a}(\vec{r}) = \frac{1}{2^k}$.

A discrete probability space

Let A be a probabilistic algorithm and let $a \in \mathrm{Input}(A)$.

1. $\sum_{\vec{r} \in \mathrm{FRand}(A,a)} \Pr(\vec{r})$ converges and its limit is in the interval $[0, 1]$.
2. Set $\Pr_{A,a}(\infty) = 1 - \sum_{\vec{r} \in \mathrm{FRand}(A,a)} \Pr_{A,a}(\vec{r})$.
3. $(\mathrm{FRand}(A, a) \cup \{\infty\}, \Pr_{A,a})$ is a discrete probability space.
4. If $\Pr_{A,a}(\infty) = 0$, then $(\mathrm{FRand}(A, a), \Pr_{A,a})$ is a discrete probability space.

Monte Carlo algorithms satisfy $\Pr_{A,a}(\infty) = 0$ for all $a \in \mathrm{Input}(a)$

Proposition

Let A be a Monte Carlo algorithm and let $a \in \mathrm{Input}(A)$. Then the following holds.

1. The running time of A on input of a is bounded by some $k \in \mathbb{N}$ that may depend on a.
2. On input of a, algorithm A returns one of finitely many possible outputs according to a probability distribution.

Success probability of Monte Carlo algorithms

Let A be a Monte Carlo algorithm and let $a \in \mathrm{Input}(A)$.

1. $\mathrm{Rand}_{\mathrm{succ}}(A, a)$: The set of all $\vec{r} \in \mathrm{Rand}(A, a)$ such that $A(a, \vec{r}) \in \mathrm{Output}(A, a)$.
2. *Success probability of A on input of a*: $\mathrm{p}_A(a) = \sum_{\vec{r} \in \mathrm{Rand}_{\mathrm{succ}}(A,a)} \mathrm{Pr}_{A,a}(\vec{r})$.
3. *Failure probability of A on input of a*: $\mathrm{q}_A(a) = 1 - \mathrm{p}_A(a)$.

Amplifying success probabilities

Success probability amplifier for an error-free Monte Carlo algorithm

Input: $a \in \mathrm{Input}(A)$ for an error-free Monte Carlo algorithm A, $k \in \mathbb{N}$
Output: $b \in \mathrm{Output}(A, a)$

```
1: repeat_A(a, k)
2:     for i = 1 to k do
3:         b ← A(a)
4:         if b ≠ "Failure" then
5:             return b
6:         end if
7:     end for
8:     return "Failure"
9: end
```

Success probability of repeat_A

Definition

Let $a \in \text{Input}(A)$. We denote the success probability of $\text{repeat}_A(a, k)$ by $\text{p}_A(a, k)$ and the failure probability of this call by $\text{q}_A(a, k) = 1 - \text{p}_A(a, k)$.

Proposition

Let $k \in \mathbb{N}$, and let $a \in \text{Input}(A)$ with $\text{p}_A(a) < 1$. Then we have

$$e^{-k\text{p}_A(a)/\text{q}_A(a)} \leq \text{q}_A(a, k) \leq e^{-k\text{p}_A(a)}. \tag{1.2}$$

Corollary

Let $a \in \text{Input}(A)$ with $\text{p}_A(a) > 0$ and let $\epsilon \in \mathbb{R}$ with $0 < \epsilon \leq 1$.

1. If $k \geq \log(1/\epsilon)/\text{p}_A(a)$, then $\text{p}_A(a, k) \geq 1 - \epsilon$.
2. If $\text{p}_A(a, k) \geq 1 - \epsilon$ then $k \geq \log(1/\epsilon)\text{q}_A(a)/\text{p}_A(a)$.

Success probability amplifier for a Monte Carlo decision algorithm A

A may make two-sided errors.

Input: $\vec{s} \in \{0,1\}^*$, $k \in \mathbb{N}$

Output: 1 if $\vec{s} \in L$ and 0 if $\vec{s} \in \{0,1\}^* \setminus L$ where L is the language decided by the Monte Carlo decision algorithm A that is used as a subroutine

```
1:  majorityVote_A(s, k)
2:      l = 0
3:      for i = 1 to k do
4:          l ← l + A(s)
5:      end for
6:      if l > k/2 then
7:          return 1
8:      else
9:          return 0
10:     end if
11: end
```

Success probability of majorityVote$_A$

Definition

Assume that a Monte Carlo algorithm A decides a language L, let $\vec{s} \in L$, and let $b \in \{0, 1\}$. Then we write $\Pr(A(\vec{s}) = b)$ for the probability that on input of $\vec{s}$ the algorithm A returns b.

Proposition

Let A be a Monte Carlo algorithm that decides a language L, let $\vec{s} \in L$, and $\epsilon \in \mathbb{R}_{>0}$ such that $\Pr(A(\vec{s}) = 1) \geq \frac{1}{2} + \epsilon$. Then for all $k \in \mathbb{N}$ we have

$$\Pr(\text{majorityVote}_A(\vec{s}, k) = 1) > 1 - e^{-2k\epsilon^2}. \tag{1.3}$$

1.4. Complexity Theory

Computational problems

Definition of computational problems

Definition

A *computational problem* is a triplet $\mathrm{CP} = (I, O, R)$ where I and O are subsets of Cartesian products of finitely many data types. Also, $R \subset I \times O$ such that for all $a \in I$ there is $b \in O$ with $(a, b) \in R$.

Definition

Let $\mathrm{CP} = (I, O, R)$ be a computational problem.

1. The elements of I are called the *instances* of CP.
2. If $(a, b) \in R$, then b is called a *solution* of the problem instance a.

Solving computational problems

Definition

Let CP $= (I, O, R)$ be a computational problem.

1. We say that a deterministic algorithm A *solves CP* if $I \subset \mathrm{Input}(A)$ and on input of a problem instance $a \in I$ the algorithm returns a solution of a.
2. We say that a Monte Carlo algorithm A solves CP if $I \subset \mathrm{Input}(A)$ and on input of $a \in I$ a successful run of A returns a solution of a.
3. We say that a Las Vegas algorithm solves CP if $I \subset \mathrm{Input}(A)$ and on input of $a \in I$ the algorithm either terminates and returns a solution of a or does not terminate.

Complexity of computational problems

Deterministic complexity of computational problems

Definition

Let CP be a computational problem and let $f : \mathbb{N} \to \mathbb{N}$ be a function.

1. We say that *CP can be solved in (deterministic) time* $\mathrm{O}(f)$ or *has time complexity* $\mathrm{O}(f)$ if there is a deterministic algorithm that solves CP and has running time $\mathrm{O}(f)$.
2. We say that CP can be solved in *(deterministic) linear*, *quasilinear*, *quadratic*, *cubic*, *polynomial*, *subexponential*, or *exponential time* or has this time complexity if there is a deterministic algorithm that solves CP and has the respective time complexity.
3. The corresponding space complexities are defined analogously.

Probabilistic complexity of computational problems

Definition

Let CP be a computational problem and let be a function.

1. We say that *CP can be solved in probabilistic time* $\mathrm{O}(f)$ if there is a Monte Carlo algorithm that solves CP, has running time $\mathrm{O}(f)$, and success probability $\geq \frac{2}{3}$.
2. We say that CP can be solved in *probabilistic linear*, *quasilinear*, *quadratic*, *cubic*, *polynomial*, *subexponential*, or *exponential time*, if there is a Monte Carlo algorithm with the respective running time that solves CP and has success probability $\geq \frac{2}{3}$.

Deterministic complexity classes

Definition

Let $f : \mathbb{N} \to \mathbb{R}_{>0}$ be a function.

1. The *complexity class* DTIME(f) is the set of all languages L for which there is a deterministic algorithm that decides L and has time complexity $\mathrm{O}(f)$.
2. The *complexity class* DSPACE(f) is the set of all languages L for which there is a deterministic algorithm that decides L and has space complexity $\mathrm{O}(f)$.

Definition

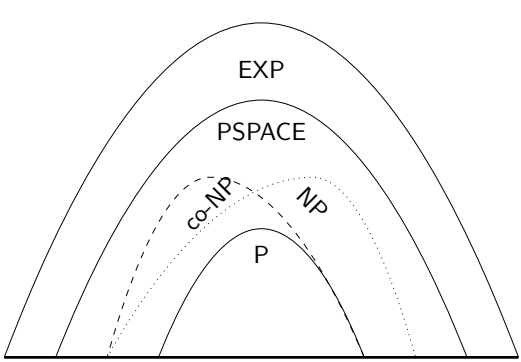
1. The *complexity class P* is the set of all languages L for which there is a deterministic polynomial time algorithm which decides L.
2. The *complexity class PSPACE* is the set of all languages L for which there is a deterministic polynomial space algorithm which decides L.
3. The complexity class *EXPTIME* is the set of all languages L for which there is a deterministic exponential time algorithm which decides L.

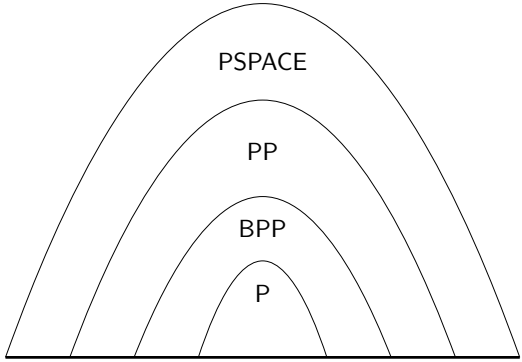
Probabilistic complexity classes

Definition

1. The *complexity class PP* is the set of all languages L for which there is a polynomial time Monte Carlo algorithm A which decides L and satisfies $\Pr(A(s) = 1) > \frac{1}{2}$ for all $s \in L$ and $\Pr(A(s) = 0) > \frac{1}{2}$ for all $s \in \{0,1\}^* \setminus L$.
2. The *complexity class BPP* is the set of all languages L for which there is a polynomial time Monte Carlo algorithm A which decides L and satisfies $\Pr(A(s) = 1) \geq \frac{2}{3}$ for all $s \in L$ and $\Pr(A(s) = 0) \geq \frac{2}{3}$ for all $s \in \{0,1\}^* \setminus L$.

Relation between complexity classes





1.5. The circuit model

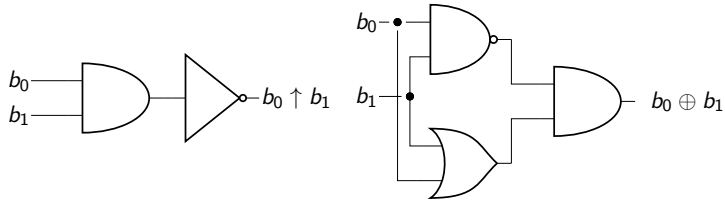
Logic gates I

Name	Logic operator	Circuit symbol
AND	$\wedge$	
OR	$\vee$	
NOT	$\neg$	
NAND	$\uparrow$	
NOR	$\downarrow$	
XOR	$\oplus$	

Logic gates II

a	b	$a \wedge b$	$a \vee b$	$\neg a$	$a \uparrow b$	$a \downarrow b$	$a \oplus b$
0	0	0	0	1	1	1	0
0	1	0	1	1	1	0	1
1	0	0	1	0	1	0	1
1	1	1	1	0	0	0	0

Boolean circuits



Universality

Definition

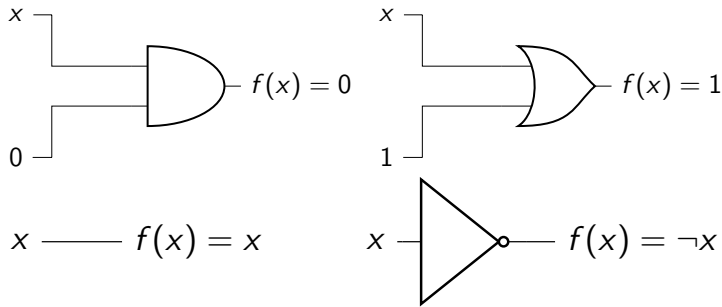
A set G of logic gates is called *universal for classical computation* if for all $m, n \in N$ and every function $f : \{0,1\}^n \to \{0,1\}^m$ there is a boolean circuit that only uses gates from G and computes f.

Theorem

The set $\{\text{NOT}, \text{AND}, \text{OR}\}$ is universal for classical computation.

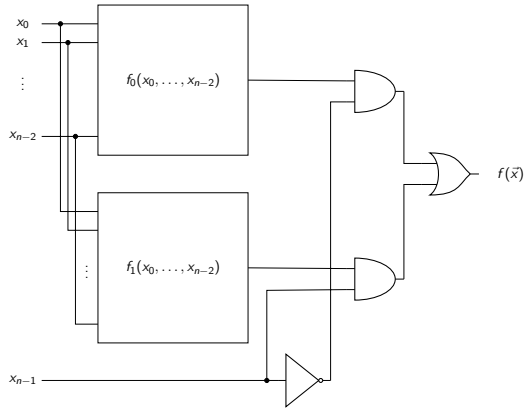
Base case of the induction proof

$$f : \{0,1\} \to \{0,1\}$$



Inductive step

$$f : \{0,1\}^n \to \{0,1\}^n$$



Universality II

Theorem

The gate set $\{\text{NAND}\}$ is universal for classical computing.

1.6. Circuit families and circuit complexity

Circuit families

Definition

A *family of circuits* or *circuit family* is a sequence $(C_n)_{n\in\mathbb{N}}$ of circuits such that the circuit C_n has n input nodes for all $n \in \mathbb{N}$.

Definition

Let $f : \{0,1\}^* \to \{0,1\}^*$ be a function that satisfies

$$|\vec{s}| = |\vec{s}'| \Rightarrow |f(\vec{s})| = |f(\vec{s}')| \text{ for all } \vec{s}, \vec{s}' \in \{0,1\}^*. \tag{1.4}$$

and let $C = (C_n)_{n\in\mathbb{N}}$ be a circuit family. We say that C computes f if for all $n \in \mathbb{N}$, the circuit C_n computes the function $f_n : \{0,1\}^n \to \{0,1\}^*$, $\vec{s} \mapsto f(\vec{s})$.

Solving computational problems and deciding languages

Definition

Let $\mathrm{CP} = (I, O, R)$ be a computational problem and let $C = (C_n)_{n\in\mathbb{N}}$ be a circuit family. We say that C solves CP if for all $n \in \mathbb{N}$ on input of $a \in \{0,1\}^n \cap I$ the circuit C_n computes a solution b of a .

Definition

Let L be a language, and let $C = (C_n)_{n\in\mathbb{N}}$ be a circuit family. We say that C decides L if for all $n \in \mathbb{N}$ on input of $\vec{s} \in \{0,1\}^n$ the circuit C_n returns 1 if $\vec{s} \in L$ and 0 otherwise.

Theorem

For all functions $f : \{0,1\}^* \to \{0,1\}^*$, computational problems CP, and languages L there is a circuit family that computes f, solves CP, or decides L.

Encoding of circuits

We assume that we have fixed an encoding of circuits by bit strings that satisfies the following.

1. The encoding is *sensible*: every circuit is encoded by at least one bit string, and every bit string encodes at most one quantum circuit.
2. The encoding is *efficient*: there is $c \in \mathbb{N}$ such that every circuit C has an encoding of length at least size C and at most $(\text{size } C)^c$.
3. Information about the structure of a circuit is computable in polynomial time from an encoding of the circuit.

"Structure information" means, for example, information about what the input nodes, the gates, and the output nodes are and how these nodes are connected.

Uniform circuit families

Definition

A circuit family $C = (C_n)$ is called *uniform* if there is a deterministic algorithm which on input of 1^n, $n \in \mathbb{N}$, outputs the encoding of C_n.

Definition

A circuit family $C = (C_n)$ is called *P-uniform* if there is a deterministic polynomial time algorithm which on input of 1^n, $n \in \mathbb{N}$, outputs the encoding of C_n.

Circuit complexity

Definition

Let $C = (C_n)_{n \in \mathbb{N}}$ be a circuit family and let $f : \mathbb{N} \to \mathbb{N}$ be a function.

1. The *size-complexity* of C is the function $\mathbb{N} \to \mathbb{N}$, $n \mapsto |C_n|$.
2. The complexity class $\mathrm{SIZE}(f)$ is the set of all languages that can be decided by a P-uniform circuit family with size-complexity $\mathrm{O}(f)$.

Theorem

Let $f : \mathbb{N} \to \mathbb{N}$ then $\mathrm{DTIME}(f) \subset \mathrm{SIZE}(f \log f)$.

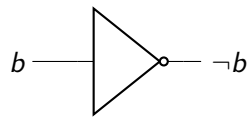
Proof: [AB09]

1.7. Reversible circuits

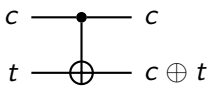
Reversible gates and circuits

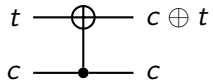
Definition

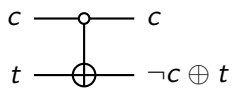
A reversible gate or circuit is a logic gate or boolean circuit that implements an invertible function $f : \{0,1\}^n \to \{0,1\}^n$ for some $n \in \mathbb{N}$, respectively.

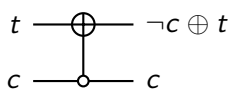


CNOT gates

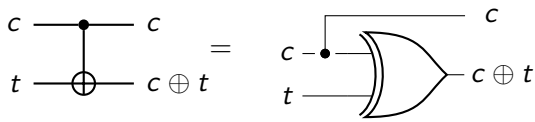




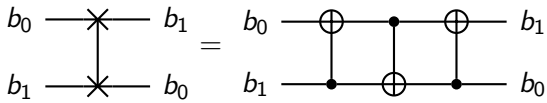




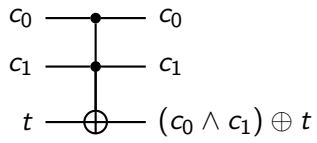
Implementation of CNOT



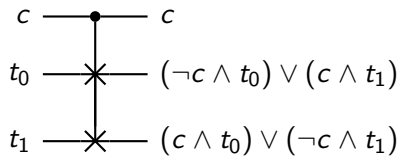
SWAP gate



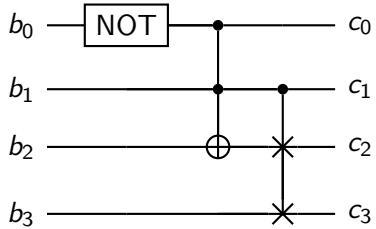
Toffoli or CCNOT gate



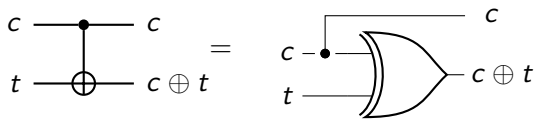
Fredkin gate



Reversible boolean circuit



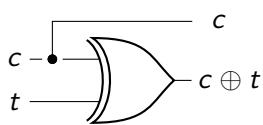
Fanout gates



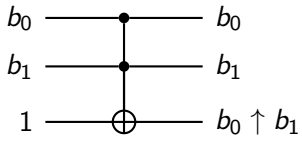
Size of Boolean Functions

Definition

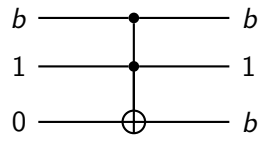
1. For a boolean circuit C we denote by $|C|_F$ the number of gates that it uses, including FANOUT gates.
2. For a boolean function f denote by $|f|_F$ the minimum value of $|C|_F$ over all boolean circuits C that implement f and use only NAND and FANOUT gates.



Implementation of NAND and FANOUT using CCNOT



NAND



FANOUT

Reversible Implementation of Boolean Functions

Theorem

For all Boolean functions $f : \{0,1\}^n \to \{0,1\}^m$, $m, n \in \mathbb{N}$, there is $p \in \mathbb{N}_0$, $p \leq 2|f|_F$, a reversible circuit C_r of size $|f|_F$ that uses only Toffoli gates, $\vec{a} \in \{0,1\}^p$, and a function $g : \{0,1\}^n \to \{0,1\}^{n+p-m}$ such that C_r implements a function

$$h : \{0,1\}^n \times \{0,1\}^p \to \{0,1\}^m \times \{0,1\}^{n+p-m} \tag{1.5}$$

with

$$h(\vec{x}, \vec{a}) = (f(\vec{x}), g(\vec{x})) \tag{1.6}$$

for all $\vec{x} \in \{0,1\}^n$. The bits in $\vec{a}$ are called *ancilla bits*. The functional value $g(\vec{x})$ is called *garbage*.

Reversible Implementation of Boolean Functions I

Theorem

For all Boolean functions $f : \{0,1\}^n \to \{0,1\}^m$, $m, n \in \mathbb{N}$, there is $p \in \mathbb{N}_0$, $p \leq 2|f|_F$, a reversible circuit D_r with $|D_r| = \mathrm{O}(|f|_F)$ that uses only Toffoli, NOT, and CNOT gates such that D_r implements a function

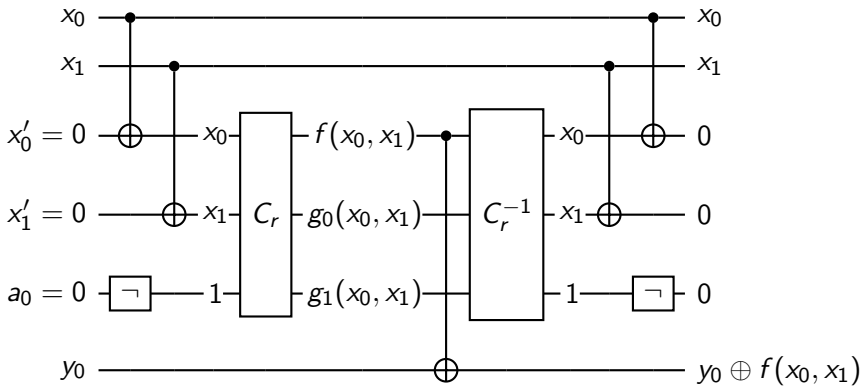
$$h : \{0,1\}^n \times \{0,1\}^{n+p} \times \{0,1\}^m \to \{0,1\}^n \times \{0,1\}^{n+p} \times \{0,1\}^m \tag{1.7}$$

with

$$h(\vec{x}, \vec{0}, \vec{y}) = (\vec{x}, \vec{0}, \vec{y} \oplus f(\vec{x})) \tag{1.8}$$

for all $\vec{x} \in \{0,1\}^n$ and $\vec{y} \in \{0,1\}^m$.

Construction of D_r



2. Hilbert Spaces

2.1. Kets and State Spaces

State Spaces

$\mathbb{H}$: k-dimensional complex vector space, $n \in \mathbb{N}$, $N = 2^n$.

Elements of $\mathbb{H}$: $|\varphi\rangle$: "ket-φ".

$\mathbb{Z}_n = \{0, \ldots, n-1\}$.

$B_n = \left(\left|\vec{b}\right\rangle\right)_{\vec{b} \in \{0,1\}^n} \mathrel{\widehat{=}} (b)_{b \in \mathbb{Z}_N}$, e.g., $B_2 = (|00\rangle, |01\rangle, |10\rangle, |11\rangle) \mathrel{\widehat{=}} (0, 1, 2, 3)$.

$\mathbb{H}_n = \sum_{\vec{b} \in \{0,1\}^n} \mathbb{C} \left|\vec{b}\right\rangle$, e.g., $\mathbb{H}_1 = \{\alpha |0\rangle + \beta |1\rangle : \alpha, \beta \in \mathbb{C}\}$.

B_n *computational basis* of $\mathbb{H}_n$.

2.2. Inner Products

Definition

B basis of $\mathbb{H}$, $|\varphi\rangle = \sum_{b\in B} \alpha_b |b\rangle$, $|\psi\rangle = \sum_{b\in B} \beta_b |b\rangle$, $\langle\varphi|\psi\rangle = \sum_{b\in B} \overline{\alpha_b}\beta_b$.

$b, c \in \mathbb{Z}_N$, $\langle b|c\rangle = ?$

Linear in the second argument: $\langle\xi|\left(|\varphi\rangle + |\psi\rangle\right) = \langle\xi|\varphi\rangle + \langle\xi|\psi\rangle$, $\langle\varphi|\left(\alpha\,|\psi\rangle\right) = \alpha\langle\varphi|\psi\rangle$.

Conjugate symmetric: $\langle\psi|\varphi\rangle = \overline{\langle\varphi|\psi\rangle}$

Positive-definite: $\langle\varphi|\varphi\rangle \geq 0$ and $\langle\varphi|\varphi\rangle = 0$ if and only if $|\varphi\rangle = 0$.

$(\mathbb{H}_n, \langle\cdot|\cdot\rangle)$ Hilbert space.

Norm

$\|\varphi\| = \sqrt{\langle\varphi|\varphi\rangle}$.

Triangle inequality: $\| \, |\varphi\rangle + |\psi\rangle \, \| \leq \|\varphi\| + \|\psi\|$.

Absolute homogeneity: $\|\alpha \, |\varphi\rangle \, \| = |\alpha| \|\varphi\|$.

Cauchy-Schwarz inequality: $|\langle\varphi|\psi\rangle| \leq \|\varphi\| \|\psi\|$.

Bras

Dual of $\mathbb{H}$: $\mathbb{H}^* = \mathrm{Hom}_{\mathbb{C}}(\mathbb{H}, \mathbb{C})$.

Elements of $\mathbb{H}^*$: $\langle\varphi|$: "bra-φ".

$|\varphi\rangle \in \mathbb{H}$: $\langle\varphi| : \mathbb{H} \to \mathbb{C}$, $|\psi\rangle \mapsto \langle\varphi|\psi\rangle$ is in $\mathbb{H}^*$ and called the *dual* of $|\varphi\rangle$.

$\mathbb{H} \mapsto \mathbb{H}^*$, $|\varphi\rangle \mapsto \langle\varphi|$ is a conjugate linear bijection: $|\xi\rangle = |\varphi\rangle + |\psi\rangle \Rightarrow \langle\xi| = \langle\varphi| + \langle\psi|$, $|\xi\rangle = \alpha\,|\varphi\rangle \Rightarrow \langle\xi| = \overline{\alpha}\,\langle\varphi|$.

Orthogonality

$|\varphi\rangle$ and $|\psi\rangle$ orthogonal to each other $\Leftrightarrow \langle\varphi|\psi\rangle = 0$.

Orthogonal sequence/basis: any two different elements are orthogonal to each other.

Orthonormal sequence/basis: orthogonal and all elements have length 1

Every orthogonal/orthonormal sequence in $\mathbb{H}$ is linear can be appended to an orthogonal/orthonormal basis of $\mathbb{H}$.

Gram-Schmidt Procedure

Theorem

Let $C = (|c_0\rangle, \ldots, |c_{k-1}\rangle)$ be a basis of $\mathbb{H}$. Set

$$|b_0\rangle = |c_0\rangle \tag{2.1}$$

and for $1 \leq j < k$ let

$$|b_j\rangle = |c_j\rangle - \sum_{i=0}^{j-1} \frac{\langle b_i|c_j\rangle}{\langle b_i|b_i\rangle} |b_i\rangle . \tag{2.2}$$

Then $(|b_0\rangle, \ldots, |b_{k-1}\rangle)$ is an orthogonal basis of $\mathbb{H}$ and for $0 \leq j < k$ we have

$$\mathrm{Span}\{|b_0\rangle, \ldots, |b_j\rangle\} = \mathrm{Span}\{|c_0\rangle, \ldots, |c_j\rangle\}. \tag{2.3}$$

Constructing Orthogonal and Orthonormal Bases

Theorem

Every orthogonal or orthonormal sequence in $\mathbb{H}$ is linearly independent and can be extended to an orthogonal or orthonormal basis of $\mathbb{H}$, respectively.

Corollary

Every finite-dimensional Hilbert space has an orthonormal basis.

Proposition

Let $B = (|b_0\rangle, \ldots, |b_{k-1}\rangle)$ be an orthonormal basis of $\mathbb{H}$ and let $\langle\varphi| \in \mathbb{H}^*$. Then we have

$$|\varphi\rangle = \sum_{i=0}^{k-1} \langle\varphi|b_i\rangle \, |b_i\rangle \, . \tag{2.4}$$

2.3. Linear maps

Linear Maps and Matrices

$B = (|b_0\rangle, \ldots, |b_{k-1}\rangle)$ basis of $\mathbb{H}$

$\mathbb{H} \to \mathbb{C}^k$, $|\varphi\rangle \mapsto |\varphi\rangle_B = (\alpha_0, \ldots, \alpha_{n-1})$ where $|\varphi\rangle = \sum_{i=0}^{k-1} \alpha_i |b_i\rangle$ is a $\mathbb{C}$-vector space isomorphism.

$\frac{|0\rangle+|1\rangle}{\sqrt{2}} \mapsto ?$

$\mathrm{End}(\mathbb{H}) \to \mathbb{C}^{(k,k)}$, $f \mapsto \mathrm{Mat}_B(f) = ((f|b_0\rangle)_B, \ldots, (f|b_{k-1}\rangle)_B)$ is an isomorphism of $\mathbb{C}$-algebras.

$X : \mathbb{H}_1 \to \mathbb{H}_1$, $X|0\rangle = |1\rangle$, $X|1\rangle = |0\rangle$, $\mathrm{Mat}_{B_1}(X) = ?$

Inverse: $M \mapsto f_{M,B}$, $f_{M,B}|\varphi\rangle = BM|\varphi\rangle_B$.

Important single-qubit operators

$\mathbb{H} = \mathbb{H}_1$, $B = (|0\rangle, |1\rangle)$

Pauli operators

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{2.5}$$

Hadamard operator

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{2.6}$$

Matrix representation of endomorphisms

Proposition

Let $B = (|b_0\rangle, \ldots, |b_{k-1}\rangle)$ and $C = (|c_0\rangle, \ldots, |c_{l-1}\rangle)$ be orthonormal bases of $\mathbb{H}$ and $\mathbb{H}'$, respectively. Then the matrix representation of a linear map $T \in \mathrm{Hom}(\mathbb{H}, \mathbb{H}')$ with respect to these bases is

$$\mathrm{Mat}_{B,C}(T) = (\langle c_i | T | b_j \rangle)_{i \in \mathbb{Z}_l, j \in \mathbb{Z}_k} \in \mathbb{C}^{(l,k)}. \tag{2.7}$$

Adjoints

Adjoint of $M \in \mathbb{C}^{(k,k)}$: $M^* = \overline{M^{\mathrm{T}}}$.

$$M = \begin{pmatrix} 1 & 1+i \\ 1-i & i \end{pmatrix},\ M^* = ?$$

Proposition

Let $A \in \mathbb{C}^{(k,l)}$, $B \in \mathbb{C}^{(m,n)}$, and $\alpha \in \mathbb{C}$. Then we have

$$\begin{aligned}
&(A^*)^* = A, \\
&(A+B)^* = A^* + B^*, \text{ if } m = k \text{ and } n = l \\
&(\alpha A)^* = \overline{\alpha} A^*, \\
&\operatorname{rank}(A) = \operatorname{rank}(A^*), \\
&(AB)^* = B^* A^*, \text{ if } l = m.
\end{aligned}$$

Adjoints of Endomorphisms

Proposition

If $A \in \mathrm{Hom}(\mathbb{H}', \mathbb{H})$, then there is a uniquely determined operator $A^* \in \mathrm{Hom}(\mathbb{H}, \mathbb{H}')$ such that

$$\left\langle \varphi \middle| A\,|\psi\rangle \right\rangle = \left\langle A^*\,|\varphi\rangle \middle| \psi \right\rangle \tag{2.8}$$

for all $|\varphi\rangle \in \mathbb{H}$ and $|\psi\rangle \in \mathbb{H}'$. The operator A^* is called the *adjoint* of A.

The Hilbert-Schmidt Inner Product

Proposition

The map

$$\langle \cdot | \cdot \rangle : \mathbb{C}^{(I,k)} \times \mathbb{C}^{(I,k)} \to \mathbb{C}, \quad (A, B) \mapsto \langle A | B \rangle = \mathrm{tr}(A^* B) \tag{2.9}$$

is an inner product on $\mathbb{C}^{(I,k)}$. It is called the *Hilbert-Schmidt inner product* on $\mathbb{C}^{(I,k)}$.

Corollary

The map

$$\langle \cdot | \cdot \rangle : \mathrm{Hom}(\mathbb{H}, \mathbb{H}') \times \mathrm{Hom}(\mathbb{H}, \mathbb{H}'), \quad (A, B) \mapsto \langle A | B \rangle = \mathrm{tr}(A^* B) \tag{2.10}$$

is an inner product on $\mathrm{Hom}(\mathbb{H}, \mathbb{H}')$. It is called the *Hilbert-Schmidt* inner product on $\mathrm{End}(\mathbb{H})$.

2.4. Endomorphisms

Properties of Endomorphisms

Proposition

Let $A \in \mathbb{C}^{(k,k)}$ or $A \in \mathrm{End}(\mathbb{H})$. Let Λ be the set of eigenvalues of A. For each $\lambda \in \Lambda$ let m_λ be its algebraic multiplicity. Then we have

$$p_A(x) = \prod_{\lambda \in \Lambda} (x - \lambda)^{m_\lambda},$$
$$\mathrm{tr}(A) = \sum_{\lambda \in \Lambda} m_\lambda \lambda,$$
$$\det(A) = \prod_{\lambda \in \Lambda} \lambda^{m_\lambda}.$$

Theorem

If all eigenvalues of $A \in \mathbb{C}^{(k,k)}$ have algebraic multiplicity 1, then A is diagonalizable.

Hermitian Operators

Definition

$A \in \mathbb{C}^{(k,k)}$ or $A \in \mathrm{End}(\mathbb{H})$ is called *Hermitian* or *self-adjoint* if $A = A^*$.

Proposition

A, B Hermitian matrices or operators. Then

1. Diagonal elements, determinant, trace, and eigenvalues of A are real numbers.
2. A invertible $\Rightarrow$ A^{-1} Hermitian.
3. $A + B$ Hermitian.
4. AB Hermitian $\Leftrightarrow$ $AB = BA$.
5. ABA Hermitian.

Are the Pauli operators and the Hadamard operator Hermitian?

Unitary operators

Definition

$U \in \mathbb{C}^{(k,k)}$ or $U \in \mathrm{End}(\mathbb{H})$ *unitary* if $U^*U = UU^* = I_k / I_{\mathbb{H}}$,

Theorem

Let $U \in \mathbb{C}^{(k,k)}$. Then the following statements are equivalent.

1. U is unitary.
2. U is invertible and $U^{-1} = U^*$.
3. The columns of U form an ONB of $\mathbb{C}^k$.
4. The rows of U form an ONB of $\mathbb{C}^k$.
5. $\langle U\vec{v}, U\vec{w}\rangle = \langle \vec{v}, \vec{w}\rangle$ for all $\vec{v}, \vec{w} \in \mathbb{C}^k$
6. $\langle U\vec{v}, U\vec{v}\rangle = \langle \vec{v}, \vec{v}\rangle$ for all $\vec{v} \in \mathbb{C}^k$.

The Unitary Group

Theorem

1. The set of all unitary matrices in $\mathbb{C}^{(k,k)}$ is a subgroup of $\mathrm{GL}(k, \mathbb{C})$. It is denoted by $\mathrm{U}(k)$ and called the *unitary group* of rank k.
2. The set of all unitary matrices of determinant 1 is a subgroup of $\mathrm{U}(k)$. It is called the *special unitary group* of rank k and is denoted by $\mathrm{SU}(k)$.

Outer Products

Definition

Let $|\varphi\rangle, |\psi\rangle \in \mathbb{H}$. Then the *outer product* of $|\varphi\rangle$ and $|\psi\rangle$ is the endomorphism

$$|\varphi\rangle \langle\psi| : \mathbb{H} \to \mathbb{H}, \quad |\xi\rangle \mapsto |\varphi\rangle \langle\psi|\xi\rangle$$

of $\mathbb{H}$.

Orthogonal Projections

$S \subset \mathbb{H}$: $S^\perp = \{|\varphi\rangle \in \mathbb{H} : \langle\psi, \varphi\rangle = 0 \,\forall\, |\psi\rangle \in S\}$. Orthogonal complement of S.

$|0\rangle^\perp = ?$

S subspace of $\mathbb{H}$:

All $|\varphi\rangle$ have unique decomposition $|\varphi\rangle = |\varphi_S\rangle + |\varphi_{S^\perp}\rangle$.

Orthogonal projection of $\mathbb{H}$ onto S: $|\varphi\rangle \mapsto |\varphi_S\rangle$.

Properties of Orthogonal Projections

Proposition

Let $\mathbb{H}(0), \mathbb{H}(1)$ be linear subspaces of $\mathbb{H}$. Then the following holds.

1. $(\mathbb{H}(0)^{\perp})^{\perp} = \mathbb{H}(0)$.
2. $\mathbb{H}$ is the direct sum of $\mathbb{H}(0)$ and $\mathbb{H}(0)^{\perp}$ and $\dim \mathbb{H}(0) + \dim \mathbb{H}(0)^{\perp} = \dim \mathbb{H}$.
3. If B_0 is an orthonormal basis of $\mathbb{H}(0)$ and B_1 is an orthonormal of $\mathbb{H}^{\perp}$, then $B_0 \parallel B_1$ is an orthonormal basis of $\mathbb{H}$.
4. If $\mathbb{H} = \mathbb{H}(0) + \mathbb{H}(1)$ and $\mathbb{H}(0)$ and $\mathbb{H}(1)$ are orthogonal to each other, then $\mathbb{H}(1) = \mathbb{H}(0)^{\perp}$.

Decomposition into Orthonormal Subspaces

Proposition

Let $l \in \mathbb{N}$ and let $\mathbb{H}(0), \ldots, \mathbb{H}(l-1)$ be subspaces of $\mathbb{H}$. Then the following holds.

1. If $\mathbb{H}(0), \ldots, \mathbb{H}(l-1)$ are pairwise orthogonal to each other, then their sum is direct.
2. The subspaces $\mathbb{H}(0), \ldots, \mathbb{H}(l-1)$ are pairwise orthogonal to each other if and only if there are orthonormal bases $B_0, \ldots, B_{l-1}$ of $\mathbb{H}(0), \ldots, \mathbb{H}(l-1)$, respectively, such that $B = B_0 \parallel \cdots \parallel B_{l-1}$ is an orthonormal basis of $\mathbb{H}(0) + \cdots + \mathbb{H}(l-1)$.

Schur Decomposition

Theorem

Let $A \in \mathbb{C}^{(k,k)}$. Assume that A has the l distinct eigenvalues $\lambda_0, \ldots, \lambda_{l-1}$ with algebraic multiplicities $m_0, \ldots, m_{l-1}$. Then $k = \sum_{i=0}^{l-1} m_i$ and there is a unitary matrix $U \in \mathbb{C}^{(k,k)}$ and an upper triangular matrix T with diagonal

$$(\underbrace{\lambda_0, \ldots, \lambda_0}_{m_0}, \underbrace{\lambda_1, \ldots, \lambda_1}_{m_1}, \ldots, \underbrace{\lambda_{l-1}, \ldots, \lambda_{l-1}}_{m_{l-1}}) \tag{2.11}$$

such that

$$A = UTU^*. \tag{2.12}$$

Such a representation is called *Schur decomposition* of A.

Normal Operators

Definition

A matrix $A \in \mathbb{C}^{(k,k)}$ or operator $A \in \mathrm{End}(\mathbb{H})$ is called *normal* if $A^*A = AA^*$.

Spectral theorem

Theorem

$A \in \mathbb{C}^{(k,k)}$ or $A \in \mathrm{End}(\mathbb{H})$ normal. Λ: the set of eigenvalues of A. $\lambda \in \Lambda$: P_λ the orthogonal projection onto the eigenspace E_λ. Then the following holds.

1. There are orthonormal bases B_λ of E_λ for all $\lambda \in \Lambda$ such that their concatenation is an orthonormal basis of $\mathbb{H}$.
2. The eigenspaces E_λ are orthogonal to each other, and their sum is $\mathbb{H}$.
3. $P_\lambda = \sum_{|b\rangle \in B_\lambda} |b\rangle \langle b|$.
4. $\sum_{\lambda \in \Lambda} P_\lambda = I_{\mathbb{H}}$.
5. $A = \sum_{\lambda \in \Lambda} \lambda P_\lambda$.

Spectral decomposition of Pauli X?

Singular Value Decomposition

Theorem

Let $k, l, r \in \mathbb{N}$ and let $A \in \mathbb{C}^{(k,l)}$ be of rank r. Then there are unitary matrices $U \in \mathbb{C}^{(k,k)}$ and $V \in \mathbb{C}^{(l,l)}$ such that

$$A = U \left(\begin{array}{ccc|ccc} \lambda_0 & \cdots & 0 & & \vdots & \\ \vdots & \ddots & \vdots & \cdots & 0 & \cdots \\ 0 & \cdots & \lambda_{r-1} & & \vdots & \\ \hline & \vdots & & & \vdots & \\ \cdots & 0 & \cdots & \cdots & 0 & \cdots \\ & \vdots & & & \vdots & \end{array}\right) V^* \tag{2.13}$$

where $\lambda_0, \ldots, \lambda_{r-1}$ are positive real numbers. Such a representation is called a *singular value decomposition* of the matrix A. In the decomposition, the diagonal entries $\lambda_0, \ldots, \lambda_{r-1}$ are uniquely determined by A up to reordering. They are called the *singular values* of A.

Functions of Operators

Definition

Let $f : \mathbb{C} \to \mathbb{C}$, let A be a normal linear operator on $\mathbb{H}$, let

$$A = \sum_{\lambda \in \Lambda} \lambda P_\lambda \tag{2.14}$$

be the spectral decomposition of A. Then we define

$$f(A) = \sum_{\lambda \in \Lambda} f(\lambda) P_\lambda. \tag{2.15}$$

Theorem

An operator $U \in \mathrm{End}(\mathbb{H})$ is unitary if and only if U can be written as $U = e^{iA}$ with a Hermitian operator $A \in \mathrm{End}(\mathbb{H})$.

2.5. Tensor Products

Definition of Tensor Products

$\mathbb{H}(1)$, $\mathbb{H}(2)$ Hilbert spaces

$L = \sum_{(|\varphi\rangle, |\psi\rangle) \in \mathbb{H}(0) \times \mathbb{H}(1)} \mathbb{C}(|\varphi\rangle, |\psi\rangle)$ is $\mathbb{C}$-vector space.

S subspace generated by all

$(|\varphi\rangle, |\psi\rangle) + (|\xi\rangle, |\psi\rangle) - (|\varphi\rangle + |\xi\rangle, |\psi\rangle)$,
$(|\varphi\rangle, |\psi\rangle) + (|\varphi\rangle, |\xi\rangle) - (|\varphi\rangle, |\psi\rangle + |\xi\rangle)$,
$\alpha(|\varphi\rangle, |\psi\rangle) - (\alpha\,|\varphi\rangle, |\psi\rangle)$, $\alpha(|\varphi\rangle, |\psi\rangle) - (|\varphi\rangle, \alpha\,|\psi\rangle)$.

Tensor product of $\mathbb{H}(1)$ and $\mathbb{H}(2)$: $\mathbb{H}(1) \otimes \mathbb{H}(2) = L/S$.

Properties of Tensor Products

Distributive law: $(\alpha_1 |\varphi_1\rangle + \alpha_2 |\varphi_2\rangle) \otimes (\beta_1 |\psi_1\rangle + \beta_2 |\psi_2\rangle) = \sum_{i,j=1}^{2} \alpha_i \beta_j |\varphi_i\rangle \otimes |\psi_j\rangle$

$|\varphi\rangle \otimes |\psi\rangle = \vec{0}$ if and only if $|\varphi\rangle = \vec{0}$ or $|\psi\rangle = \vec{0}$.

B_i basis of $\mathbb{H}(i) \Rightarrow B_1 \otimes B_2$ basis of $\mathbb{H}(1) \otimes \mathbb{H}(2)$.

$\dim \mathbb{H}(1) \otimes \mathbb{H}(2) = \dim \mathbb{H}(1) \dim \mathbb{H}(2)$.

$\left\langle |\varphi_1\rangle \otimes |\varphi_2\rangle \,\middle|\, |\psi_1\rangle \otimes |\psi_2\rangle \right\rangle = \langle\varphi_1|\psi_1\rangle\langle\varphi_2|\psi_2\rangle$

$f_i \in \mathrm{End}(\mathbb{H}(i))$: $f_1 \otimes f_2 : |b_1\rangle \otimes |b_2\rangle \mapsto f_1 |b_1\rangle \otimes f_2 |b_2\rangle \in \mathrm{End}(\mathbb{H}(1) \otimes \mathbb{H}(2))$.

Tensor Product of State Spaces

$$\mathbb{H}_n = \sum_{\vec{b} \in \{0,1\}^n} \mathbb{C}\vec{b}$$

$$\mathbb{H}_m \otimes \mathbb{H}_m \mathrel{\widehat{=}} \mathbb{H}_{m+n}$$

Schmidt Decomposition Theorem

Theorem

$\mathbb{H}(0)$ and $\mathbb{H}(1)$, Hilbert spaces of dimensions k and l, respectively, $|\varphi\rangle \in \mathbb{H}(0) \otimes \mathbb{H}(1)$, $m = \min\{k, l\}$. Then there are orthonormal sequences $(|u_0\rangle, \ldots, |u_{m-1}\rangle)$ in $\mathbb{H}(0)$ and $(|v_0\rangle, \ldots, |v_{m-1}\rangle)$ in $\mathbb{H}(1)$ and $r_0, \ldots, r_{m-1} \in \mathbb{R}_{\geq 0}$ such that

$$|\varphi\rangle = \sum_{i=0}^{m-1} r_i \, |u_i\rangle \otimes |v_i\rangle \, . \tag{2.16}$$

Up to reordering, the coefficients r_i are uniquely determined by $|\varphi\rangle$.

\# of nonzero r_i: Schmidt rank or Schmidt number of φ r_i: Schmidt coefficients of φ.

$m = 1$: $|\varphi\rangle$ *separable*, $m > 1$: $|\varphi\rangle$ *insebarable* or *entangled*.

3. Quantum Mechanics

3.1. State Spaces

State Space Postulate

A closed physical system is associated with a Hilbert space, called the *state space* of the system. The system at a particular time is completely described by a unit vector in its state space, called the *state vector* or *state* of the physical system.

Quantum Bits

Qubit: physical system with state space $\mathbb{H}_1 = \mathbb{C}\,|0\rangle + \mathbb{C}\,|1\rangle$.

State of a qubit: Superposition $|\varphi\rangle = \alpha_0\,|0\rangle + \alpha_1\,|1\rangle$, $|\alpha_0|^2 + |\alpha_1|^2 = 1$

Spherical Coordinates

Proposition

Let $\vec{p} \in \mathbb{R}^3$, $\vec{p} \neq 0$. Then there are uniquely determined real numbers θ and ϕ with

$$0 \leq \theta \leq \pi \text{ and } \begin{cases} \phi = 0 & \text{if } \theta \in \{0, \pi\}, \\ 0 < \phi < 2\pi & \text{otherwise} \end{cases} \tag{3.1}$$

such that

$$\vec{p} = \|\vec{v}\|(\cos\phi \sin\theta, \sin\phi \sin\theta, \cos\theta). \tag{3.2}$$

The triplet $(\|\vec{p}\|, \theta, \phi)$ is called the *spherical coordinate representation* of $\vec{p}$. Its elements are called the *spherical coordinates* of $\vec{p}$ and are referred to as $r(\vec{p}) = \|\vec{p}\|$, $\theta(\vec{p}) = \theta$, and $\phi(\vec{p}) = \phi$.

Bloch Sphere

Theorem

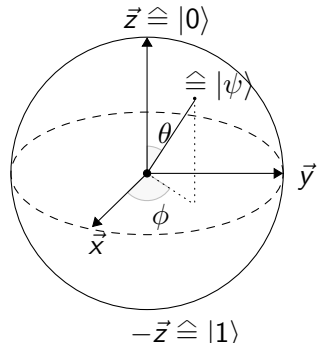
Let $|\psi\rangle \in \mathbb{H}_1$ be a single-qubit state. Then there are uniquely determined real numbers γ, θ, and ϕ such that

$$|\psi\rangle = e^{i\gamma}\left(\cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle\right)$$

and

$$0 \leq \theta \leq \pi, \quad 0 \leq \gamma, \phi < 2\pi, \quad \theta \in \{0, \pi\} \Rightarrow \phi = 0.$$

We write these numbers as $\gamma(\psi)$, $\theta(\psi)$, and $\phi(\psi)$.



Point on the Bloch sphere corresponding to $|\psi\rangle$.

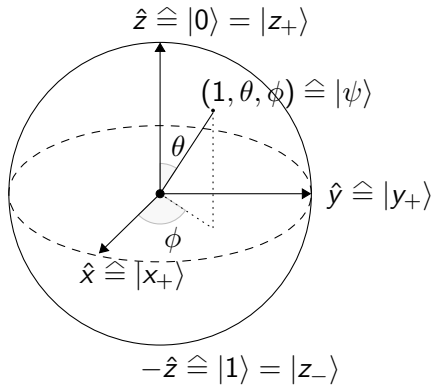
Bloch Sphere II

Definition

1. To each single-qubit state $|\psi\rangle \in \mathbb{H}_1$ we assign the point $\vec{p}(\psi)$ on the Bloch sphere with spherical coordinates $(1, \theta(\psi), \phi(\psi))$ and Cartesian coordinates $(\sin\theta(\psi)\cos\phi(\psi), \sin\theta(\psi)\sin\phi(\psi), \cos\theta(\psi))$.
2. To each point $\vec{p}$ on the Bloch sphere with spherical coordinates $(1, \theta, \phi)$ we assign the single-qubit state

$$|\psi(\vec{p})\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle\,. \tag{3.3}$$

Bloch Sphere III



Points on the Bloch sphere corresponding to $|x_+\rangle$, $|y_+\rangle$, $|z_+\rangle = |0\rangle$, $|z_-\rangle = |1\rangle$, and a general single-qubit state $|\psi\rangle$.

Global Phase Factors

Definition

Let $|\varphi\rangle, |\psi\rangle \in \mathbb{H}$ and let $\gamma \in \mathbb{R}$ be such that $|\psi\rangle = e^{i\gamma}|\varphi\rangle$. Then we say that $|\varphi\rangle$ and $|\psi\rangle$ *are equal up to the global phase factor* $e^{i\gamma}$ or that these states *differ by the global phase factor* $e^{i\gamma}$.

Proposition

Let S be the set of all quantum states in the Hilbert space $\mathbb{H}$. Then the subset of S^2 of all pairs of quantum states that are equal up to a global phase factor is an equivalence relation on S. For $|\psi\rangle \in \mathbb{H}$, we denote the equivalence class of $|\psi\rangle$ with respect to this relation by $[\psi]$.

Quantum States Corresponding to the Same Point on the Bloch Sphere

Theorem

Denote by S_1 the set of quantum states in $\mathbb{H}_1$ and by R_1 the equivalence relation on S_1 from the Proposition on the previous slide. Then the map

$$S_1/R_1 \to \{\vec{p} \in \mathbb{R}^3 : \|\vec{p}\| = 1\}, \quad [\psi] \mapsto \vec{p}(\psi) \tag{3.4}$$

is a bijection. Its inverse is

$$\{\vec{p} \in \mathbb{R}^3 : \|\vec{p}\| = 1\} \to S_1/R_1, \quad \vec{p} \mapsto [\psi(\vec{p})]. \tag{3.5}$$

Quantum Registers

n-qubit quantum register: quantum system comprising n qubits.

State space $\mathbb{H}_n = \sum_{\vec{b} \in \{0,1\}^n} \mathbb{C} \left| \vec{b} \right\rangle$.

n-qubit state : Superposition $|\varphi\rangle = \sum_{\vec{b} \in \{0,1\}^n} \alpha_{\vec{b}} \left| \vec{b} \right\rangle$, $\sum_{\vec{b} \in \{0,1\}^n} |\alpha_{\vec{b}}|^2 = 1$.

$\alpha_{\vec{b}}$: amplitude of $\left| \vec{b} \right\rangle$.

3.2. State spaces of composite systems

Composite Systems Postulate

The state space of the composition of finitely many physical system is the tensor product of the state spaces of the component physical systems. Moreover, if we have systems numbered 0 through $m-1$ and if system i is in state $|\psi_i\rangle$ for $0 \leq i < m$, then the composite system is in state $|\psi_0\rangle \otimes \cdots \otimes |\psi_{m-1}\rangle$.

Example: $\mathbb{H}_m \otimes \mathbb{H}_n \mathrel{\widehat{=}} \mathbb{H}_{m+n}$

Entangled and Separable States

Definition

A state of the composition of two physical systems is called *entangled* if it cannot be written as the tensor product of states of the component systems. Otherwise, this state is called *separable* or *non-entangled*.

Theorem

The state of the composition of two quantum systems is separable if and only if its Schmidt rank is 1 and it is entangled if and only if its Schmidt rank is greater than 1.

3.3. Time Evolution of Quantum Systems

Evolution Postulate

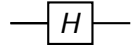
The evolution of a closed quantum system is described by a unitary transformation. More precisely, if $t, t' \in \mathbb{R}$, $t < t'$, then the state $|\varphi'\rangle$ of the system at time t' is obtained from the state $|\varphi\rangle$ of the system at time t as $|\varphi'\rangle = U|\varphi\rangle$ where U is a unitary operator on the state space of the system that depends only on t and t'.

Quantum Gates $\equiv$ Quantum Operators

Hadamard gate:

$$H : \mathbb{H}_1 \to \mathbb{H}_1, \quad |0\rangle \mapsto |x_+\rangle, \quad |1\rangle \mapsto |x_-\rangle$$

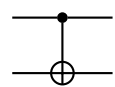
$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \quad$$



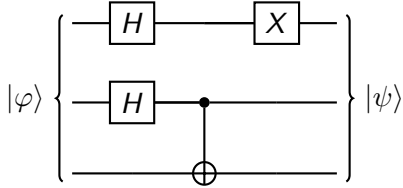
CNOT gate:

$$\text{CNOT} : \mathbb{H}_2 \to \mathbb{H}_2, \quad |c\rangle\,|t\rangle \mapsto |c\rangle\, X^c\,|t\rangle$$

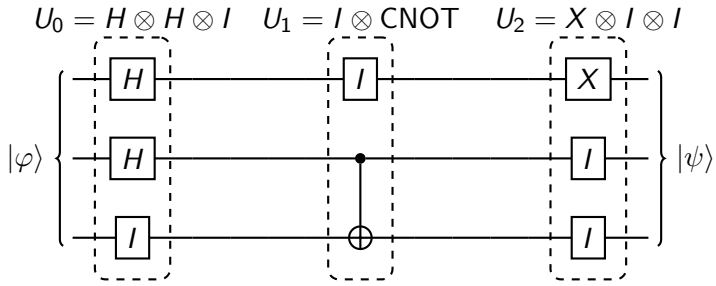
$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} =$$



Quantum Circuits

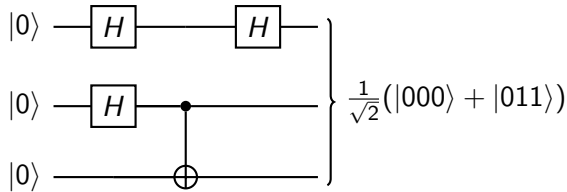


Operators Implemented by Quantum Circuits



This quantum circuit implements the unitary operator $U = U_2 \circ U_1 \circ U_0$.

Example


$|0\rangle$
$|0\rangle$
$|0\rangle$
H
H
H
$\frac{1}{\sqrt{2}}(|000\rangle + |011\rangle)$

3.4. Measurements

Measurement Postulate

A *projective measurement* is described by an *observable* O which is a Hermitian operator on the state space of the system being observed. Let $O = \sum_{\lambda \in \Lambda} \lambda P_\lambda$ be the spectral decomposition of O. The possible outcomes of the measurement are the eigenvalues λ of the observable. When measuring O while the quantum system is in the state $|\varphi\rangle$, the probability of getting the result corresponding to λ is $\Pr(\lambda) = \langle\varphi|P_\lambda|\varphi\rangle = \|P_\lambda|\varphi\rangle\|^2$. If this outcome occurs, the state of the quantum system immediately after the measurement is $\frac{P_\lambda|\varphi\rangle}{\sqrt{\Pr(\lambda)}}$.

Expectation value of O: Expectation value of the random variable that sends the measurement outcome to the corresponding eigenvalue $= \langle\varphi|O|\varphi\rangle$.

Expectation Value of an Observable O

For all $|\varphi\rangle \in \mathbb{H}$, the Measurement Postulate defines the discrete probability space $(\Lambda, \Pr_{O,\varphi})$. Since O is Hermitian we have $\Lambda \subset \mathbb{R}$. So the identity map I_Λ on Λ is a random variable associated with the probability space.

Lemma

We have $E[I_\Lambda] = \langle\psi|O|\psi\rangle$.

Measuring States that Differ by a Global Phase Factor

$$|\psi\rangle = e^{i\gamma}\,|\varphi\rangle$$

$$\|P_\lambda\,|\psi\rangle\,\| = \|P_\lambda\,|\varphi\rangle\,\|$$

$$\frac{P_\lambda|\psi\rangle}{\sqrt{\Pr(\lambda)}} = e^{i\gamma}\,\frac{P_\lambda|\psi\rangle}{\sqrt{\Pr(\lambda)}}$$

Measuring in an Orthonormal Basis

Definition

Let $\mathbb{H}$ be the state space of a quantum system and let $B = (|b_0\rangle, \ldots, |b_{k-1}\rangle)$ be an orthonormal basis of $\mathbb{H}$. By measuring the quantum system in the basis B we mean measuring the observable

$$O = \sum_{j=0}^{k-1} j \, |b_j\rangle \langle b_j| \tag{3.6}$$

of $\mathbb{H}$.

Measurement in the Computational Basis

$n \in \mathbb{N}$, $N = 2^n$, $B = (|\lambda\rangle)_{\lambda \in \mathbb{Z}_N}$.

$O_n = \sum_{\lambda \in \mathbb{Z}_N} \lambda \, |\lambda\rangle \langle\lambda|$.

Measuring O_n when the system is in state $|\varphi\rangle = \sum_{\lambda \in \mathbb{Z}_N} \alpha_\lambda \, |\lambda\rangle$ gives λ with probability $\| P_\lambda \, |\varphi\rangle \|^2 = |\alpha_\lambda|^2$.

If λ is measured then system is in state $\frac{\alpha}{|\alpha|} \, |\lambda\rangle$.

Partial Measurement

Measure the first qubit of $|\xi\rangle = \frac{|00\rangle+|11\rangle}{\sqrt{2}}$,

$O = 0 \cdot (|0\rangle\langle 0| \otimes I_1) + 1 \cdot (|1\rangle\langle 1| \otimes I_1)$, $P_0 = |0\rangle\langle 0| \otimes I_1$.

$P_0|00\rangle = (|0\rangle\langle 0| \otimes I_1)|0\rangle \otimes |0\rangle = |0\rangle\langle 0|0\rangle \otimes |0\rangle = |00\rangle$

$P_0|11\rangle = (|0\rangle\langle 0| \otimes I_1)|1\rangle \otimes |1\rangle = |0\rangle\langle 0|1\rangle \otimes |1\rangle = \vec{0}$

$P_0|\xi\rangle = \frac{1}{\sqrt{2}}|00\rangle$, $\Pr(0) = \| P_0 \frac{1}{\sqrt{2}}|00\rangle \| = \frac{1}{2}$, state after measuring 0: $\sqrt{2}P_0|\xi\rangle = |00\rangle$.

$P_1 = |1\rangle\langle 1| \otimes I_1$, $P_1|\xi\rangle = \frac{1}{\sqrt{2}}|11\rangle$

$\Pr(1) = \frac{1}{2}$, state after measuring 1: $|11\rangle$.

3.5. Density Operators

Definition

Definition

A *density operator* on $\mathbb{H}$ is a linear operator ρ on $\mathbb{H}$ that satisfies the following conditions.

1. *Trace condition:* $\operatorname{tr} \rho = 1$,
2. *Positivity condition:* ρ is positive-semidefinite.

Mixed States

Definition

1. A *mixed state* of the quantum system Q is a sequence

$$((p_0, |\psi_0\rangle), \ldots (p_{k-1}, |\psi_{l-1}\rangle)) \tag{3.7}$$

   where $l \in \mathbb{N}$, the $|\psi_i\rangle$ are quantum states in $\mathbb{H}$ for $0 \leq i < l$ and $p_i \in \mathbb{R}_{\geq 0}$ for $0 \leq i < l$ such that $\sum_{i=0}^{l-1} p_i = 1$.

2. A *pure state* of the quantum system Q is a quantum state in its state space $\mathbb{H}$.

Proposition

Let $((p_0, |\psi_0\rangle), \ldots (p_{l-1}, |\psi_{l-1}\rangle))$ be a mixed state of the quantum system Q. Then

$$\rho = \sum_{i=0}^{l-1} p_i \, |\psi_i\rangle \, \langle\psi_i| \tag{3.8}$$

is a density operator on the state space $\mathbb{H}$ of Q.

Density Operators of Pure and Mixed States

Definition

1. The density operator of a mixed state

$$S = (p_0, |\psi_0\rangle), \dots (p_{l-1}, |\psi_{l-1}\rangle) \tag{3.9}$$

of Q is defined as

$$\rho_S = \sum_{i=0}^{l-1} p_i \, |\psi_i\rangle \, \langle\psi_i| \, . \tag{3.10}$$

2. The density operator of a pure state $|\psi\rangle \in \mathbb{H}$ is defined as

$$\rho_\psi = |\psi\rangle \, \langle\psi| \, . \tag{3.11}$$

Correspondence Between States and Density Operators I

Proposition

Every density operator on $\mathbb{H}$ is the density operator of some mixed state of the quantum system Q.

Construction

$$\rho = \sum_{i=0}^{k-1} \lambda_i \left|b_i\right\rangle \left\langle b_i\right| \tag{3.12}$$

$$S = ((\lambda_0, |b_0\rangle), \ldots, (\lambda_{k-1}, |b_{k-1}\rangle))$$

Correspondence Between States and Density Operators II

Theorem

1. The density operators of two pure states of Q are the same if and only these states are equal up to a global phase factor.
2. Let $l \in \mathbb{N}$. The density operators of two mixed states

$$((p_0, |\varphi_0\rangle), \ldots, (p_{l-1}, |\varphi_{l-1}\rangle)\,, \quad ((q_0, |\psi_0\rangle), \ldots, (q_{l-1}, |\psi_{l-1}\rangle) \tag{3.13}$$

of Q are the same if and only if there is a unitary matrix $U \in \mathbb{C}^{(l,l)}$ such that

$$(\sqrt{p_0}\,|\varphi_0\rangle), \ldots, \sqrt{p_{l-1}}\,|\varphi_{l-1}\rangle) = (\sqrt{q_0}\,|\psi_0\rangle), \ldots, \sqrt{q_{l-1}}\,|\psi_{l-1}\rangle)U. \tag{3.14}$$

Theorem

The set R of all pairs of mixed states of Q with the same density operator is an equivalence relation on the set of all mixed states of Q.

Distinguishing Between Pure and Mixed States

Theorem

Let ρ be a density operator on $\mathbb{H}$. Then the following statements hold.

1. ρ is the density operator of a pure state if and only if $\rho^2 = \rho$, which is true if and only if $\operatorname{tr} \rho^2 = 1$.
2. ρ is not the density operator of a pure state if and only if $\rho^2 \neq \rho$, which is true if and only if $\operatorname{tr} \rho^2 < 1$.

3.6. The quantum postulates for mixed states

The State Spaces Postulate

Postulate

Associated with any physical system is a Hilbert space, called the state space of the system. The system is completely described by a density operator on the state space.

The Composite Systems Postulate

Postulate

The state space of the composition of finitely many physical systems is the tensor product of the state spaces of the component physical systems. Moreover, if we have systems numbered 0 through $m-1$ and if system i is in the state ρ_i where ρ_i is a density operator on the state space of the ith component system for $0 \leq i < k$, then the composite system is in the state $\rho_0 \otimes \cdots \otimes \rho_{m-1}$.

The Evolution Postulate

Postulate

The evolution of a quantum system with state space $\mathbb{H}$ is described by a unitary transformation on $\mathbb{H}$. More precisely, if $t, t' \in \mathbb{R}$, $t < t'$. Assume that the state of the system at time t is described by the density operator ρ on $\mathbb{H}$. Then the state of the system at time t' is obtained from ρ as $\rho' = U\rho U^*$ where U is a unitary operator on $\mathbb{H}$ that only depends on t and t'.

The Measurement Postulate

Postulate

A projective measurement is described by an observable O that is a Hermitian operator on the state space of the system being observed. Let $O = \sum_{\lambda \in \Lambda} \lambda P_\lambda$ be the spectral decomposition of O. The possible outcomes of the measurement are the eigenvalues of the observable. When measuring the state ρ the probability of getting the result corresponding to λ is $\Pr(\lambda) = \mathrm{tr}(P_\lambda \rho)$. If this outcome occurs, the state immediately after the measurement is $\frac{P_\lambda \rho P_\lambda}{\Pr(\lambda)}$.

Expectation Values of Observables

Definition

Let O be an observable of a quantum system with state space $\mathbb{H}$. Suppose that we measure this observable when the system is in a state described by the density operator ρ. Then the *expectation value of this measurement* is defined as $\mathrm{tr}(O\rho)$.

Measuring in an Orthonormal Basis

Proposition

Suppose that we measure the quantum system in the orthonormal basis $B = (|b_0\rangle, \ldots, |b_{k-1}\rangle)$ when it is in the mixed state. Then measuring the observable $\sum_{\lambda=0}^{k-1} \lambda |\lambda_j\rangle \langle\lambda_j|$ gives $\lambda \in \mathbb{Z}_k$ with probability $\Pr(\lambda) = \sum_{i=0}^{l-1} p_i |\langle b_\lambda | \varphi_i \rangle|^2$. Immediately after this measurement, the quantum system is in the state $|b_\lambda\rangle \langle b_\lambda|$.

3.7. Partial Trace and Reduced Density Operators

Definition of Partial Trace

Theorem

Let $m \in \mathbb{N}$, let $V_0, \dots V_{m-1}$ be vector spaces, and let $J \subset \mathbb{Z}_m$. Then there is a uniquely determined linear map

$$\mathrm{tr}_J : \mathrm{End}\left(\bigotimes_{j\in\mathbb{Z}_m} V_j\right) \to \mathrm{End}\left(\bigotimes_{j\in\mathbb{Z}_m\setminus J} V_j\right) \tag{3.15}$$

that satisfies

$$\mathrm{tr}_J\left(\bigotimes_{j\in\mathbb{Z}_m} f_j\right) = \prod_{j\in J} \mathrm{tr}\, f_j \bigotimes_{j\in\mathbb{Z}_m\setminus J} f_j. \tag{3.16}$$

for all $(f_0, \dots, f_{m-1}) \in \prod_{j=0}^{m-1} \mathrm{End}(V_j)$. It is called the *partial trace* over the V_j, $j \in J$.

Reduced Density Operator

Let A and B be quantum systems with state spaces $\mathbb{H}_A$ and $\mathbb{H}_B$ and let AB be the composite quantum system with state space $\mathbb{H}_{AB} = \mathbb{H}_A \otimes \mathbb{H}_B$.

Proposition

Let ρ be a density operator on $\mathbb{H}_{AB}$. Then $\mathrm{tr}_B(\rho)$ is a density operator on $\mathbb{H}_A$. It is called the *reduced density operator* of ρ on the subsystem A and is denoted by ρ^A.

Tracing out Subsystems I

Theorem

1. Let O_A be an observable of system A, let $O_{AB} = O_A \otimes I_B$, and assume that the state of the quantum system ρ. Then the expectation value of O_{AB} is the same as the expectation value of O_A when system A is in the reduced state ρ^A, i.e.,

$$\mathrm{tr}(O_{AB}\rho) = \mathrm{tr}(O_A\rho^A). \tag{3.17}$$

2. The function

$$\mathrm{End}(\mathbb{H}_{AB}) \to \mathrm{End}(\mathbb{H}_A), \quad \rho \mapsto \rho^A = \mathrm{tr}_B(\rho) \tag{3.18}$$

is the only linear map that satisfies (3.17) for all observables O_A of A and all states ρ of AB.

Tracing out Subsystems II

Proposition

Let $l \in \mathbb{N}$ and for $0 \leq i < l$ let $|\varphi_i\rangle$ and $|\psi_i\rangle$ be quantum states in $\mathbb{H}_A$ and $\mathbb{H}_B$, respectively, such that the states $|\psi_i\rangle$ are orthogonal to each other. Also, let ρ be the density operator of the state

$$|\xi\rangle = \frac{1}{\sqrt{l}} \sum_{i=0}^{l-1} |\varphi_i\rangle |\psi_i\rangle \tag{3.19}$$

Then ρ^A is the density operator of the mixed state

$$\left(\left(\frac{1}{l}, |\varphi_0\rangle \right), \ldots, \left(\frac{1}{l}, |\varphi_{l-1}\rangle \right) \right). \tag{3.20}$$

In other words, if the composite system AB is in the state $\rho = |\xi\rangle \langle\xi|$, then the state of system A after tracing out system B can be described by the mixed state (3.20)

Tracing out Subsystems III

Corollary

Assume that the composite system AB is in the state $\rho = |\xi\rangle\langle\xi|$ where $|\xi\rangle = |\varphi\rangle|\psi\rangle$ with $|\varphi\rangle \in \mathbb{H}_A$ and $|\psi\rangle \in \mathbb{H}_B$. Then $\rho^A = |\varphi\rangle\langle\varphi|$. This means that the state of system A after tracing out system B can be described by the state vector $|\varphi\rangle$.

Theorem

Let $|\varphi\rangle$ be the state of the composite system AB and let $\rho = |\varphi\rangle\langle\varphi|$ be its density operator. Then $|\varphi\rangle$ is entangled with respect to the decomposition of AB into the subsystems A and B if and only if the reduced density operator ρ^A is not the density operator of a pure state.

4. The Theory of Quantum Algorithms

4.1. Simple Single-Qubit Operators

The Identity Gate

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

The Pauli Gates

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

$$\begin{aligned} X &= |x_+\rangle \langle x_+| - |x_-\rangle \langle x_-|, \\ Y &= |y_+\rangle \langle y_+| - |y_-\rangle \langle y_-|, \\ Z &= |z_+\rangle \langle z_+| - |z_-\rangle \langle z_-|. \end{aligned}$$

where

$$\begin{aligned} (|x_+\rangle, |x_-\rangle) &= \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}, \frac{|0\rangle - |1\rangle}{\sqrt{2}} \right), \\ (|y_+\rangle, |y_-\rangle) &= \left(\frac{|0\rangle + i|1\rangle}{\sqrt{2}}, \frac{|0\rangle - i|1\rangle}{\sqrt{2}} \right), \\ (|z_+\rangle, |z_-\rangle) &= (|0\rangle, |1\rangle). \end{aligned}$$

Properties of the Pauli Gates

Theorem

The Pauli gates are Hermitian and unitary involutions which satisfy

$$XY = iZ = -YX, \quad ZX = iY = -XZ, \quad YZ = iX = -ZY,$$

and

$$-iXYZ = I.$$

Proposition

The sequence (I, X, Y, Z) is a $\mathbb{C}$-basis of $\mathrm{End}(\mathbb{H}_1)$ which is orthogonal with respect to the Hilbert-Schmidt inner product.

The Hadamard Gate

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

$$HXH = Z, \quad HYH = -Y, \quad HZH = X.$$

$$H = \frac{1}{\sqrt{2}}(X + Z)$$

.

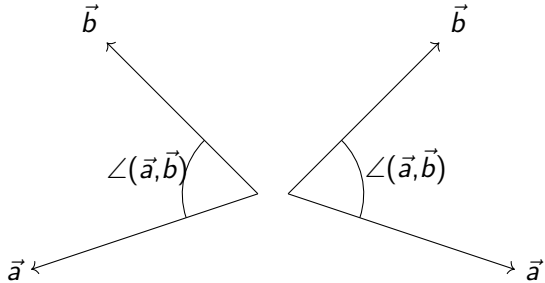
4.2. More geometry of $\mathbb{R}^3$

The Angle between two Vectors

Definition

Let $\vec{a}, \vec{b} \in \mathbb{R}^3$ be nonzero. Then the *angle between* $\vec{a}$ *and* $\vec{b}$ is defined as

$$\angle(\vec{a}, \vec{b}) = \arccos \frac{\langle \vec{a} | \vec{b} \rangle}{\|\vec{a}\| \|\vec{b}\|}.$$



Properties of the Angle

Proposition

Let $\vec{a}, \vec{b} \in \mathbb{R}^3$ be nonzero vectors . Then we have

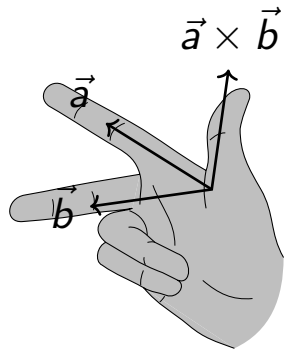
1. $0 \leq \angle(\vec{a}, \vec{b}) = \angle(\vec{b}, \vec{a}) \leq \pi$.
2. $\angle(\vec{a}, \vec{b}) = 0$ if and only if $\vec{b} = r\vec{a}$ with $r \in \mathbb{R}_{>0}$.
3. $\angle(\vec{a}, \vec{b}) = \pi/2$ if and only if $\langle \vec{a} | \vec{b} \rangle = 0$, that is, $\vec{a}$ and $\vec{b}$ are orthogonal to each other.
4. $\angle(\vec{a}, \vec{b}) = \pi$ if and only if $\vec{b} = r\vec{a}$ with $r \in \mathbb{R}_{<0}$.

The Cross Product

Definition

Let $\vec{a} = (a_x, a_y, a_z), \vec{b} = (b_x, b_y, b_z) \in \mathbb{R}^3$. Then the *cross product* or *outer product* of $\vec{a}$ and $\vec{b}$ is

$$\vec{a} \times \vec{b} = (a_y b_z - a_z b_y, a_z b_x - a_x b_z, a_x b_y - a_y b_x).$$



Properties of the Cross Product I

Proposition

Let $\vec{a}, \vec{b} \in \mathbb{R}^3$ and let θ be the angle between $\vec{a}$ and $\vec{b}$. Then the following holds.

1. $\|\vec{a} \times \vec{b}\| = \|\vec{a}\|\|\vec{b}\| \sin\theta$.
2. $\det(\vec{a}, \vec{b}, \vec{c}) = \langle \vec{a} \times \vec{b} | \vec{c} \rangle$ for all $\vec{c} \in \mathbb{R}^3$.
3. $\vec{a} \times \vec{b}$ is orthogonal to $\vec{a}$ and $\vec{b}$.
4. $\vec{a} \times \vec{b} = 0$ if and only if $\vec{a}$ and $\vec{b}$ are linearly dependent.

Properties of the Cross Product II

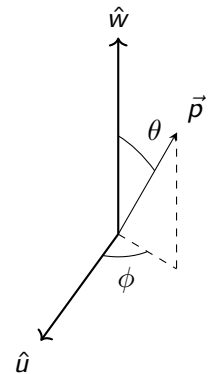
Theorem

Let $\hat{a}, \hat{b} \in \mathbb{R}^3$ be unit vectors that are orthogonal to each other. Then $\hat{p} = \hat{r} = \hat{a} \times \hat{b}$ and $\hat{q} = \hat{b} \times \hat{a}$ are the uniquely determined vectors in $\mathbb{R}^3$ such that $(\hat{p}, \hat{a}, \hat{b})$, $(\hat{a}, \hat{q}, \hat{b})$, and $(\hat{a}, \hat{b}, \hat{r})$ are orthonormal bases of $\mathbb{R}^3$ with determinant 1.

General Spherical Coordinates

Definition

Let $\hat{u}, \hat{w}$ be unit vectors that are orthogonal to each other. Let $B = (\hat{u}, \hat{v}, \hat{w})$ be an orthonormal basis of $\mathbb{R}^3$ with determinant 1 which according to Theorem exists and is uniquely determined. Also, let $\vec{p} \in \mathbb{R}^3$. Then the *spherical coordinate representation of $\vec{p}$ with respect to the azimuth reference $\hat{u}$ and the zenith $\hat{w}$* or *with respect to* $(\hat{u}, \hat{w})$ for short is defined as the spherical coordinate representation of $B^{-1}\vec{p}$.



Changing the Azimuth Reference

Proposition

Let $\hat{u}, , \hat{w} \in \mathbb{R}^3$ be unit vectors and assume that both $\hat{u}$ and are orthogonal to $\hat{w}$. Then the following holds.

1. The spherical coordinate representation of with respect to $(\hat{u}, \hat{w})$ is $(1, \pi/2, \delta)$ where $\cos\delta = \langle\hat{u}|\rangle$ and $\sin\delta = \langle\hat{w} \times \hat{u}|\rangle$.
2. Let $\vec{p} \in \mathbb{R}^3$ and let (r, θ, ϕ) and (r', θ', ϕ') be the spherical coordinate representations of $\vec{p}$ with respect to $(\hat{u}, \hat{w})$ and $(, \hat{w})$, respectively. Then we have $r' = r$, $\theta' = \theta$ and

$$\phi' = \begin{cases} 0 & \text{if } \phi = 0, \\ \phi - \delta \bmod 2\pi & \text{otherwise.} \end{cases}$$

Orthogonal Matrices

Definition

1. A matrix $O \in \mathbb{R}^{(3,3)}$ is called *orthogonal* if O is invertible and $O^{-1} = O^{\mathrm{T}}$.
2. The set of all orthogonal matrices is denoted by $\mathrm{O}(3)$.

Proposition

Let $O \in \mathbb{R}^{(3,3)}$. Then the following statements are equivalent.

1. $O \in \mathrm{O}(3)$.
2. The columns of O form an orthonormal basis of $\mathbb{R}^3$.
3. The rows of O form an orthonormal basis of $\mathbb{R}^3$.
4. $\langle O\hat{v} | O\hat{w} \rangle = \langle \hat{v} | \hat{w} \rangle$ for all $\hat{v}, \hat{w} \in \mathbb{R}^3$.
5. $\|O\hat{v}\| = \|\hat{v}\|$ for all $\hat{v} \in \mathbb{R}^3$.

The Orthogonal and the Special Orthogonal Group

Theorem

1. The set $\mathrm{O}(3)$ of all orthogonal matrices is a group with respect to matrix multiplication. It is called the *orthogonal group* of rank 3.
2. The set of all orthogonal matrices with determinant 1 is a subgroup of $\mathrm{O}(3)$. It is denoted by $\mathrm{SO}(3)$ and called the *special orthogonal group* of rank 3.

Rotations in $\mathbb{R}^3$

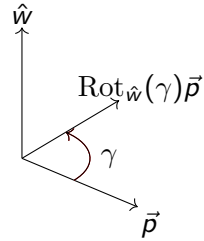
Theorem

Let $\hat{u}, \hat{w} \in \mathbb{R}^3$ be unit vectors and orthogonal to each other, and let $\gamma \in \mathbb{R}$. Consider the map $\mathbb{R}^3 \to \mathbb{R}^3$ that sends $\vec{p} \in \mathbb{R}^3$ with spherical coordinates (r, θ, ϕ) with respect to $(\hat{u}, \hat{w})$ to the vector in R^3 with the following spherical coordinate representation with respect to $(\hat{u}, \hat{w})$:

$$\begin{cases} (r, \theta, \phi) & \text{if } \theta \in \{0, \pi\}, \\ (r, \theta, (\phi + \gamma) \bmod 2\pi) & \text{otherwise.} \end{cases}$$

Then this map depends only on $\hat{w}$ and γ and is independent of $\hat{u}$. It is denoted by $\mathrm{Rot}_{\hat{w}}(\gamma)$ and called the *rotation about $\hat{w}$ through the angle γ*. Also, $\hat{w}$ and γ are called the *axis* and the *angle* of this rotation, respectively.

Rotations II



Rotation of $\vec{p}$ about $\hat{w}$ through the angle γ.

The Set of Rotations

Theorem

The set of rotations in $\mathbb{R}^3$ is $\mathrm{SO}(3)$.

Decomposition of Rotations

Theorem

For every $O \in \mathrm{SO}(3)$ there are $\alpha, \beta, \gamma \in \mathbb{R}$ such that

$$O = \mathrm{Rot}_{\hat{z}}(\alpha)\,\mathrm{Rot}_{\hat{y}}(\beta)\,\mathrm{Rot}_{\hat{z}}(\gamma).$$

The real numbers α, β, γ are called the *Euler angles* of O.

Theorem

Let $\hat{a}, \hat{b} \in \mathbb{R}^3$ be non-parallel unit vectors. Denote by φ the angle between $\hat{a}$ and $\hat{b}$. Then for all $O \in \mathrm{SO}(3)$ there are $k \in \mathbb{N}$ and $\alpha_1, \ldots, \alpha_k, \beta_1, \ldots, \beta_k \in \mathbb{R}$ such that $k = \mathrm{O}(1/\varphi)$ and

$$O = \prod_{i=1}^{k} \mathrm{Rot}_{\hat{a}}(\alpha_i)\,\mathrm{Rot}_{\hat{b}}(\beta_i).$$

4.3. Rotation Operators

Preliminaries

The triplet of Pauli operators: $\sigma = (X, Y, Z)$.

For all $\vec{p} = (p_x, p_y, p_z) \in \mathbb{R}^3$ set $\vec{p} \cdot \sigma = p_x X + p_y Y + p_z Z$.

Proposition

Let $\hat{p} \in \mathbb{R}^3$ be a unit vector. Then $\hat{p} \cdot \sigma$ is a Hermitian unitary involution with trace 0 and eigenvalues ± 1.

Proposition

For all unit vectors $\hat{w} \in \mathbb{R}^3$ and all $\gamma \in \mathbb{R}$

$$e^{-i\gamma\, \hat{w}\cdot\sigma/2} = \cos\frac{\gamma}{2} I - i \sin\frac{\gamma}{2} \hat{w} \cdot \sigma$$

is a unitary operator on $\mathbb{H}_1$ with determinant 1, i.e., in $\mathrm{SU}(2)$.

Definition of Rotation Operators

Definition

A *rotation gate* or *rotation operator* is an operator

$$R_{\hat{w}}(\gamma) = e^{-i\gamma\,\hat{w}\cdot\sigma/2} = \cos\frac{\gamma}{2}I - i\sin\frac{\gamma}{2}\hat{w}\cdot\sigma$$

on $\mathbb{H}_1$ where $\hat{w} \in \mathbb{R}^3$ is a unit vector and $\gamma \in \mathbb{R}$.

Proposition

For all unit vectors $\hat{w} \in \mathbb{R}^3$ and all $\beta, \gamma \in \mathbb{R}$ we have

$$R_{\hat{w}}(\beta)R_{\hat{w}}(\gamma) = R_{\hat{w}}(\beta + \gamma).$$

Rotations about the x, y, and z-axes

Definition

Let $\gamma \in \mathbb{R}$. The *rotation operators about the x, y, and z axes through the angle γ* are defined as

$$R_{\hat{x}}(\gamma) = e^{-i\gamma X/2}, \quad R_{\hat{y}}(\gamma) = e^{-i\gamma Y/2}, \quad R_{\hat{z}}(\gamma) = e^{-i\gamma Z/2}.$$

Proposition

Let $\gamma \in \mathbb{R}$, then we have

$$R_{\hat{x}}(\gamma) = \begin{pmatrix} \cos\frac{\gamma}{2} & -i\sin\frac{\gamma}{2} \\ -i\sin\frac{\gamma}{2} & \cos\frac{\gamma}{2} \end{pmatrix},$$

$$R_{\hat{y}}(\gamma) = \begin{pmatrix} \cos\frac{\gamma}{2} & -\sin\frac{\gamma}{2} \\ \sin\frac{\gamma}{2} & \cos\frac{\gamma}{2} \end{pmatrix},$$

$$R_{\hat{z}}(\gamma) = \begin{pmatrix} e^{-i\frac{\gamma}{2}} & 0 \\ 0 & e^{i\frac{\gamma}{2}} \end{pmatrix}.$$

$\mathrm{su}(2)$

Definition

The set of all Hermitian operators on $\mathbb{H}_1$ with trace 0 is denoted by $\mathrm{su}(2)$.

Proposition

The set $\mathrm{su}(2)$ is a real three-dimensional vector space. The triplet $\sigma = (X, Y, Z)$ of the three Pauli operators is an $\mathbb{R}$-basis of $\mathrm{su}(2)$ that is orthogonal with respect to the Hilbert-Schmidt inner product.

The Operators in $\mathrm{SU}(2)$ are Exactly the Rotation Operators

Theorem

The set of rotation operators on $\mathbb{H}_1$ is $\mathrm{SU}(2)$. Moreover, if $U \in \mathrm{SU}(2)$, then the following holds.

1. $U = I$ if and only if $U = R_{\hat{w}}(\gamma)$ with a unit vector $\hat{w} \in \mathbb{R}^3$ and $\gamma/2 \equiv 0 \bmod 2\pi$.
2. $U = -I$ if and only if $U = R_{\hat{w}}(\gamma)$ with a unit vector $\hat{w} \in \mathbb{R}^3$ and $\gamma/2 \equiv \pi \bmod 2\pi$.
3. Let $U \neq \pm I$.
   - 3.1 There are a unit vector $\hat{w} \in \mathbb{R}^3$ and $\gamma \in \mathbb{R}$ such that $U = R_{\hat{w}}(\gamma)$.
   - 3.2 If $\in \mathbb{R}^3$ is a unit vector and $\gamma' \in \mathbb{R}$, then $U = R(\gamma')$ if and only if $\hat{w} =$ and $\gamma/2 \equiv \gamma'/2 \bmod 2\pi$ or $\hat{w} = -$ and $\gamma/2 \equiv -\gamma'/2 \bmod 2\pi$.

Corollary

Let $U \in \mathrm{U}(2)$. Then there is $\delta \in \mathbb{R}$ such that $e^{-i\delta}U$ is a rotation operator on $\mathbb{H}_1$.

Rotation Operators and Rotations on the Bloch Sphere

Corollary

Let $U \in \mathrm{SU}(2)$. Then for all unit vectors $\hat{w} \in \mathbb{R}$ and $\gamma \in \mathbb{R}$ such that $U = R_{\hat{w}}(\gamma)$ the rotation $\mathrm{Rot}_{\hat{w}}(\gamma)$ is the same.

Definition

Let $U \in \mathrm{SU}(2)$ and let $U = R_{\hat{w}}(\gamma)$ with a unit vector $\hat{w} \in \mathbb{R}^3$ and $\gamma \in \mathbb{R}$. Then we set $\mathrm{Rot}(U) = \mathrm{Rot}_{\hat{w}}(\gamma)$.

Theorem

The map

$$\mathrm{Rot} : \mathrm{SU}(2) \to \mathrm{SO}(3), \quad U \mapsto \mathrm{Rot}(U)$$

is a surjective group homomorphism with kernel $\pm I$. Furthermore, for all $U \in \mathrm{SU}(2)$ and all quantum states $|\psi\rangle$ in $\mathbb{H}_1$ the point on the Bloch sphere corresponding to $|\psi\rangle$ is

$$\vec{p}(U|\psi\rangle) = \mathrm{Rot}(U)\vec{p}(\psi).$$

Decomposition of Rotation Operators I

Theorem

For every $U \in \mathrm{U}(2)$ there are $\alpha, \beta, \gamma, \delta \in \mathbb{R}$ such that

$$U = e^{i\delta} R_{\hat{z}}(\alpha) R_{\hat{y}}(\beta) R_{\hat{z}}(\gamma).$$

If $U \in \mathrm{SU}(2)$, then there is such a representation with $\delta = 0$.

Theorem

Let $\vec{a}, \vec{b} \in \mathbb{R}^3$ be non-parallel unit vectors. Denote by φ the angle between $\vec{a}$ and $\vec{b}$. Then for all unitary operators U on $\mathbb{H}_1$ there are $k \in \mathbb{N}$ and real numbers $\alpha_1, \ldots, \alpha_k, \beta_1, \ldots, \beta_k, \delta$ such that $k = \mathrm{O}(1/\varphi)$ and

$$U = e^{i\delta} \prod_{i=1}^{k} R_{\vec{a}}(\alpha_i) R_{\vec{b}}(\beta_i).$$

If $U \in \mathrm{SU}(2)$, then there is such a representation with $\delta = 0$.

Decomposition of Rotation Operators II

Theorem

Let U be a unitary operator on $\mathbb{H}_1$. Let $\alpha, \beta, \gamma, \delta \in \mathbb{R}$ such that

$$U = e^{i\delta} R_{\hat{z}}(\alpha) R_{\hat{y}}(\beta) R_{\hat{z}}(\gamma).$$

Such a representation exists by Theorem . Set

$$A = R_{\hat{z}}(\alpha) R_{\hat{y}}\left(\frac{\beta}{2}\right), B = R_{\hat{y}}\left(-\frac{\beta}{2}\right) R_{\hat{z}}\left(-\frac{\alpha+\gamma}{2}\right), C = R_{\hat{z}}\left(-\frac{\alpha-\gamma}{2}\right).$$

Then we have

$$ABC = I \text{ and } U = e^{i\delta} AXBXC.$$

Phase Shift Gates

Definition

For $\gamma \in \mathbb{R}$ the *phase shift gate* $P(\gamma)$ is defined as

$$P(\gamma) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\gamma} \end{pmatrix}.$$

It shifts the phase of the amplitude of $|1\rangle$ by an angle γ while it does not change the amplitude of $|0\rangle$.

The Phase Gate

For $k \in \mathbb{N}$ we set

$$R_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^k}} \end{pmatrix} = e^{\frac{2\pi i}{2^{k-1}}} R_{\hat{z}} \left(\frac{2\pi}{2^{k-1}} \right).$$

For $k = 2$ we obtain the *phase gate*

$$S = R_2 = P\left(\frac{\pi}{2}\right) = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}.$$

The $\pi/8$ Gate

$$T = R_3 = P\left(\frac{\pi}{4}\right) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}.$$

T is called "$\pi/8$ gate" since it can be written as

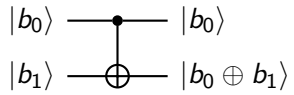
$$T = e^{\frac{i\pi}{8}} R_{\hat{z}}\left(\frac{\pi}{8}\right).$$

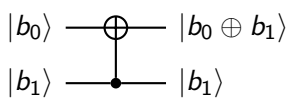
We note that

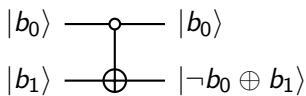
$$T^2 = S.$$

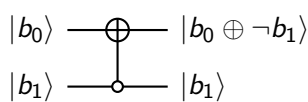
4.4. Controlled operators

Controlled NOT Gate CNOT





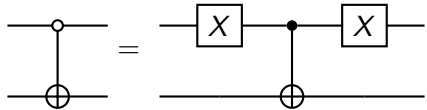


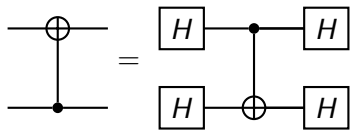


Matrix Representation of CNOT

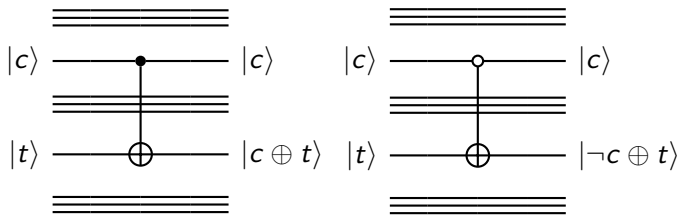
$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

Implementation of Non-Standard CNOT Gates

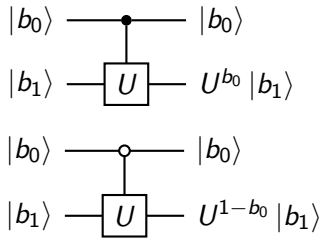


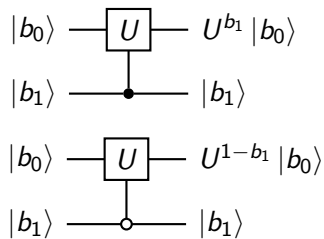


General CNOT Gates



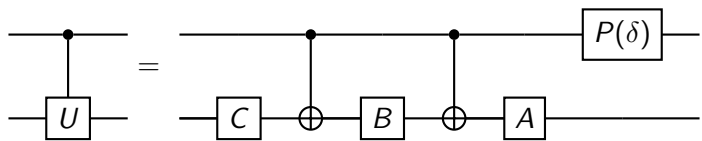
Controlled-U operators





Implementation of a Controlled-U Operator

Implementation of the controlled-U gate with the first qubit as control using the decomposition $U = e^{i\delta} AXBXC$.



General Controlled Operators

Definition

Let C_0, C_1, and T be pairwise disjoint subsets of the index set $\mathbb{Z}_n$. Let $m = |T| > 0$. and let $T = \{t, t+1, \ldots, t+m-1\}$ with $t \in \mathbb{Z}_n$. So T is a set of m consecutive integers in the index set $\mathbb{Z}_n$. Also, let U be a unitary operator on $\mathbb{H}_m$. Then the linear operator $C^{C_0,C_1,T}(U)$ is defined by its action on the computational basis states $|b_0 \cdots b_{n-1}\rangle$ of $\mathbb{H}_n$ as follows. It applies U to the *target qubits* $|b_t \cdots b_{t+m-1}\rangle$ conditioned on the *control qubits* $|b_i\rangle$ with $i \in C_0$ being $|0\rangle$ and the *control qubits* $|b_i\rangle$ with $i \in C_1$ being $|1\rangle$, i.e.,

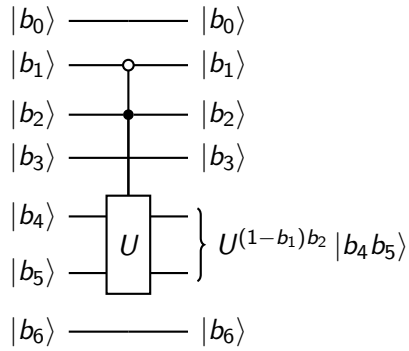
$$C^{C_0,C_1,T}(U)\,|b_0 \cdots b_{n-1}\rangle = |b_0 \cdots b_{t-1}\rangle\, U^c\, |b_t \cdots b_{t+m-1}\rangle\, |b_m \cdots b_{n-1}\rangle\,.$$

where

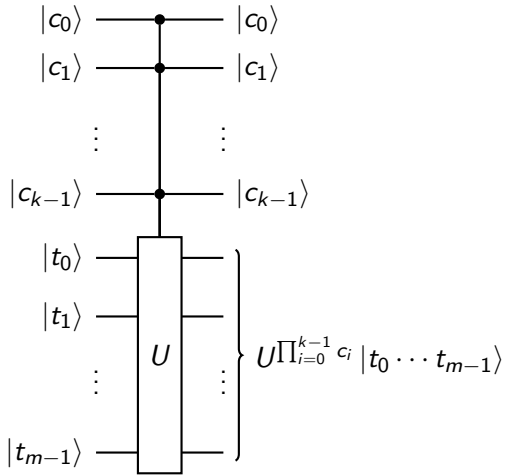
$$c = \prod_{i \in C_0} (1 - b_i) \prod_{i \in C_1} b_i.$$

If any of the index sets C_0, C_1, or T has only one element, then we replace the set in the superscript by this element.

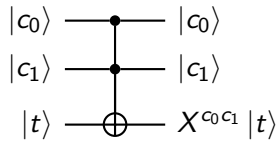
The Operator $C^{1,2,\{4,5\}}(U)$



The Operator $C^k(U)$

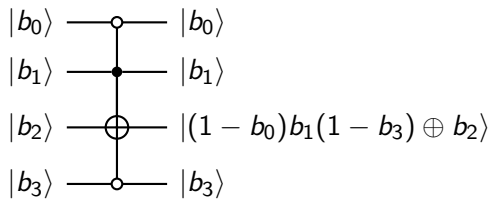


The Quantum Toffoli Gate



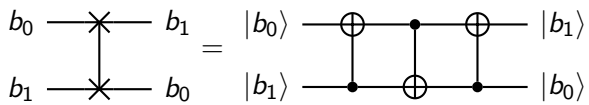
Transposition Operators

The operator $\text{TRANS}^{(01*0)}$, which exchanges $|0100\rangle$ and $|0110\rangle$.

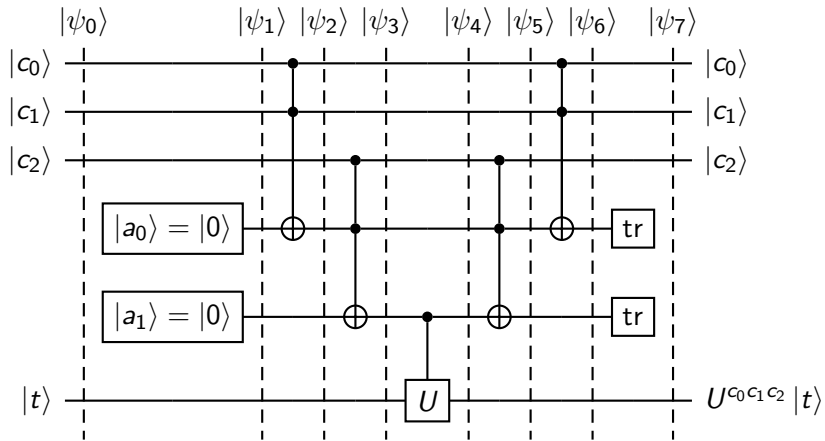


The Quantum Swap Gate

The quantum SWAP gate and its implementation using CNOT gates.



A Quantum Circuit for $C^3(U)$ with Ancillary and Erasure Gates



Algorithmic Representation of the Quantum Circuit for $C^3(U)$

Input: $|c_0\rangle |c_1\rangle |c_2\rangle |t\rangle$
Output: $|c_0\rangle |c_1\rangle |c_2\rangle U^{c_0 \cdot c_1 \cdot c_2} |t\rangle$

1: $C^3(U)$
2: Insert ancilla qubits $|a_0\rangle, |a_1\rangle$ after $|c_3\rangle$ and initialize them to $|0\rangle$
3: $|a_0\rangle \leftarrow X^{c_0 \cdot c_1} |a_0\rangle$
4: $|a_1\rangle \leftarrow X^{c_2 \cdot a_0} |a_1\rangle$
5: $|t\rangle \leftarrow U^{a_1} |t\rangle$
6: $|a_1\rangle \leftarrow X^{c_2 \cdot a_0} |a_1\rangle$
7: $|a_0\rangle \leftarrow X^{c_0 \cdot c_1} |a_0\rangle$
8: Trace out $|a_0\rangle$ and $|a_1\rangle$
9: The final state is $|c_0\rangle |c_1\rangle |c_2\rangle |t\rangle$
10: **end**

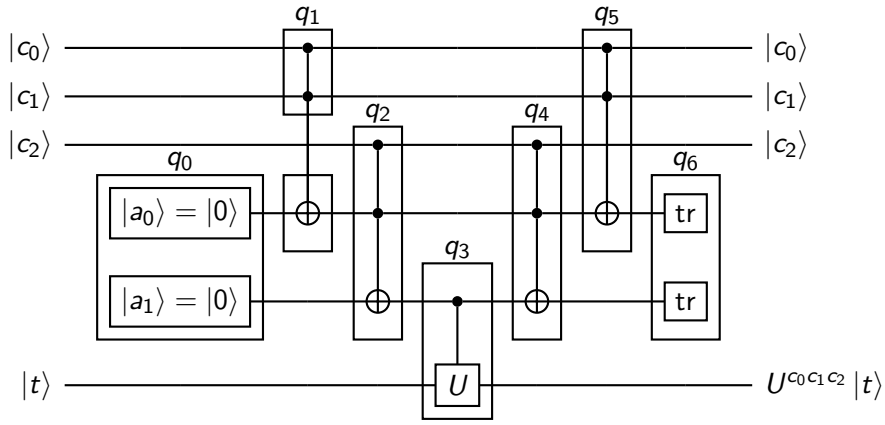
Definition of Quantum Circuits

Definition

A *quantum circuit* Q is specified by two positive integers n and k and a finite sequence $(q_0, \ldots, q_{k-1})$. Here, n is the number of input qubits and for all $i \in \mathbb{Z}_k$ the component q_i contains the following.

1. a tuple of quantum gates that are either all ancillary gates, all unitary gates, or all erasure gates and
2. the information how the ancilla qubits are initialized and where they are inserted or to which qubits the unitary or erasure gates are applied, respectively. At most one gate is applied to each qubit.

Illustration of the Definition with the Quantum Circuit for $C^3(U)$



Computation of a Quantum Circuit

The computation of a quantum circuit Q can be described as an evolution

$$|\psi_0\rangle, |\psi_1\rangle, \cdots, |\psi_k\rangle$$

of quantum states that are defined as follows.

1. The initial state $|\psi_0\rangle \in \mathbb{H}_n$ is the input of the computation.
2. For $i \in \mathbb{Z}_k$ the state $|\psi_{i+1}\rangle$ is obtained by applying the quantum gates in q_i to $|\psi_i\rangle$ as specified in q_i.
3. The final state is $|\psi_k\rangle = |c_0 \cdots c_{m-1}\rangle \in \mathbb{H}_m$ with $m = n + n_a - n_e$ where n_a is the number of ancillary gates and n_e is the number of erasure gates used in the quantum circuit.

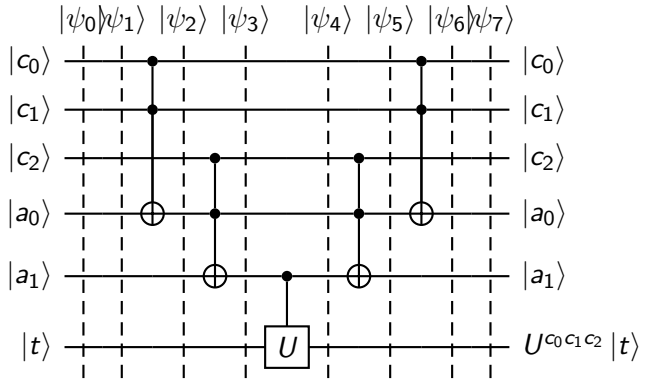
The *quantum operator implemented by* Q is

$$\mathbb{H}_n \to \mathbb{H}_m, \quad |\psi_0\rangle \mapsto |\psi_k\rangle$$

Computation of the Quantum Circuit for $C^3(U)$

i	$\lvert\psi_i\rangle$
0	$\lvert c_0\rangle \lvert c_1\rangle \lvert c_2\rangle \lvert t\rangle$
1	$\lvert c_0\rangle \lvert c_1\rangle \lvert c_2\rangle \lvert 0\rangle \lvert 0\rangle \lvert t\rangle$
2	$\lvert c_0\rangle \lvert c_1\rangle \lvert c_2\rangle \lvert c_0 \cdot c_1\rangle \lvert 0\rangle \lvert t\rangle$
3	$\lvert c_0\rangle \lvert c_1\rangle \lvert c_2\rangle \lvert c_0 \cdot c_1\rangle \lvert c_0 \cdot c_1 \cdot c_2\rangle \lvert t\rangle$
4	$\lvert c_0\rangle \lvert c_1\rangle \lvert c_2\rangle \lvert c_0 \cdot c_1\rangle \lvert c_0 \cdot c_1 \cdot c_2\rangle U^{c_0 \cdot c_1 \cdot c_1} \lvert t\rangle$
5	$\lvert c_0\rangle \lvert c_1\rangle \lvert c_2\rangle \lvert c_0 \cdot c_1\rangle \lvert 0\rangle U^{c_0 \cdot c_1 \cdot c_1} \lvert t\rangle$
6	$\lvert c_0\rangle \lvert c_1\rangle \lvert c_2\rangle \lvert 0\rangle \lvert 0\rangle U^{c_0 \cdot c_1 \cdot c_1} \lvert t\rangle$
7	$\lvert c_0\rangle \lvert c_1\rangle \lvert c_2\rangle U^{c_0 \cdot c_1 \cdot c_1} \lvert t\rangle = C^3(U) \lvert\psi\rangle_0$

Purification of the Quantum Circuit for $C^3(U)$



4.5. Universal sets of quantum gates

Constructing All Quantum Circuits

Theorem

Let S be a set of quantum gates such that for every $n \in \mathbb{N}$ and every unitary operator U on $\mathbb{H}_n$ there is a quantum circuit that implements U and uses only gates from S. Then S is uncountable.

The Distance Between Unitary Operators

Definition

Let U and V be two unitary operators on $\mathbb{H}_n$. Then we define the *error when V is implemented instead of U* as

$$E(U,V) = \sup\{\|(U-V)\,|\varphi\rangle\| : |\varphi\rangle \in \mathbb{H}_n, \langle\psi|\psi\rangle = 1\}.$$

We also call this error the *distance* between U and V.

Proposition

Let U and V be unitary operators on $\mathbb{H}_n$, let O be an observable on $\mathbb{H}_n$, let $O = \sum_{\lambda\in\Lambda} \lambda P_\lambda$ be the spectral decomposition of O. Then for all $\lambda \in \Lambda$ and all quantum states $|\psi\rangle \in \mathbb{H}_n$ we have

$$\left|\left\langle U\,|\psi\rangle \middle| P_\lambda \middle| U\,|\psi\rangle \right\rangle - \left\langle V\,|\psi\rangle \middle| P_\lambda \middle| V\,|\psi\rangle \right\rangle\right| \leq 2E(U,V).$$

Universal Sets of Quantum Gates

Definition

Let S be a set of unitary quantum gates.

1. We say that S is *universal* for a set T of unitary quantum operators if for every $\epsilon \in \mathbb{R}_{>0}$ and every $U \in T$ there is a unitary operator V which can – up to a global phase factor – be implemented by a quantum circuit that uses only gates from S, ancillary gates, and erasure gates such that $E(U, V) < \epsilon$.
2. We say that S is *universal for quantum computation* if S is universal for the set of all unitary quantum operators.

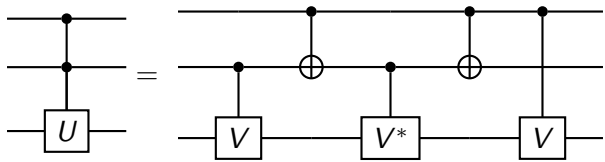
Perfectly Universal Sets of Quantum Gates

Definition

We call a set S of unitary quantum gates *perfectly universal for quantum computation* or *perfectly universal* for short, if all unitary operators U on $\mathbb{H}_n$ can, up to a global phase factor, be implemented by a quantum circuit that uses only gates from S, ancillary gates, and erasure gates.

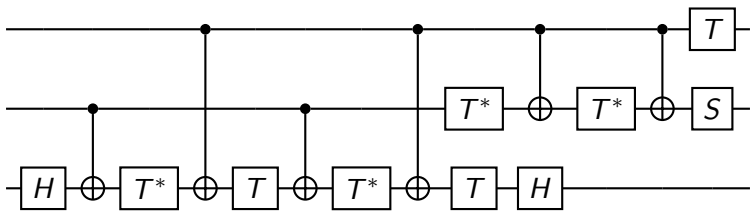
4.6. Implementation of Controlled Operators

Implementation of $C^2(U)$ if $U = V^2$

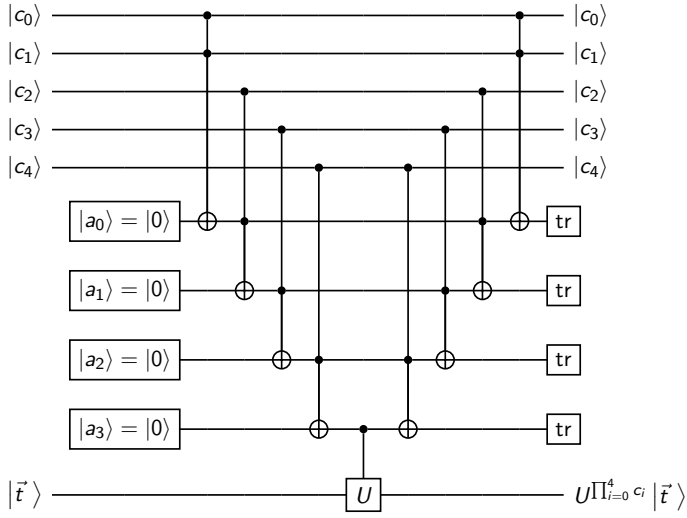


Implementation of the Toffoli Gate

An implementation of the Toffoli gate that uses the Hadamard, phase, CNOT, and $\pi/8$ gates



Implementation of $C^5(U)$ Using Only CCNOT and $C^1(U)$



Implementation of $C^k(U)$ Using Only CCNOT and $C^1(U)$

Input: $|c_0 \cdots c_{k-1}\rangle\,|t_0 \cdots t_{m-1}\rangle$
Output: $C^k(U)\,|c_0 \cdots c_{k-1}\rangle\,|t_0 \cdots t_{m-1}\rangle$

1: $C^k(U)$
2: Insert $k-1$ ancilla qubits $|a_0\rangle, \ldots, |a_{k-2}\rangle$ after the control qubit $|c_{k-1}\rangle$ and initialize them to $|0\rangle$
3: $|a_0\rangle \leftarrow X^{c_0 \cdot c_1}\,|a_0\rangle$
4: **for** $j = 1 \cdots k-2$ **do**
5: $|a_j\rangle \leftarrow X^{c_{j+1} \cdot a_{j-1}}\,|0\rangle$
6: **end for**
7: $|t_0 \cdots t_{m-1}\rangle \leftarrow U^{a_{k-2}}\,|t_0 \cdots t_{m-1}\rangle$
8: **for** $j = k-2, \ldots, 1$ **do**
9: $|a_j\rangle \leftarrow X^{c_{j+1} \cdot a_{j-1}}\,|a_j\rangle$
10: **end for**
11: $|a_0\rangle \leftarrow X^{c_0 \cdot c_1}\,|a_0\rangle$
12: Trace out $|a_0 \cdots a_{k-2}\rangle$
13: The final state is $|c_0 \cdots c_{k-1}\rangle\,|t_0 \cdots t_{m-1}\rangle$
14: **end**

Complexity of the $C^k(U)$ Implementation

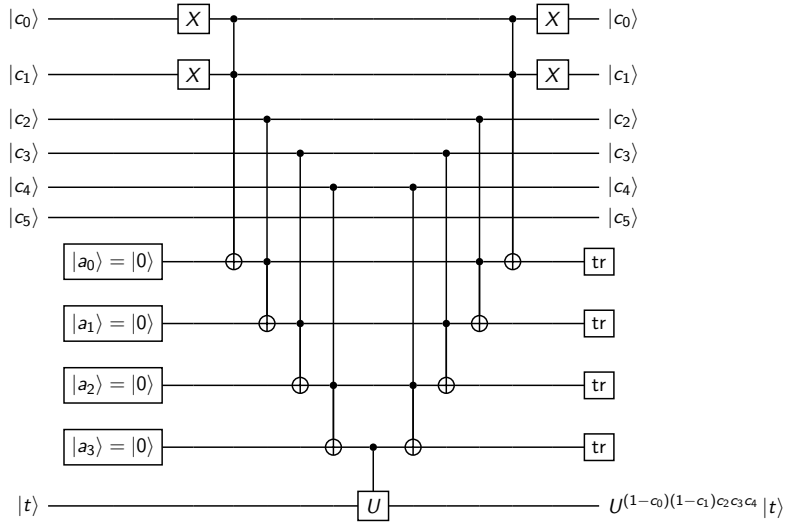
Proposition

Let U be a unitary operator on $\mathbb{H}_m$ and let $k \in \mathbb{N}$, $k \geq 2$. Then the algorithm on slide 228 implements $C^k(U)$. It uses $2k-2$ CCNOT gates, one $C^1(U)$ gate, and $k-1$ ancillary and erasure gates.

Proposition

Let U be a unitary operator on $\mathbb{H}_m$ and let $k \in \mathbb{N}$, $k \geq 2$. Then the algorithm on slide 228 implements $C^k(U)$ using $\mathrm{O}(k)$ Hadamard, phase, $\pi/8$, inverse $\pi/8$, CNOT, ancillary, and erasure gates, and one $C^1(U)$ gate.

Implementing $C^{\{0,1\},\{2,3,4\},6}(U)$ Using Only X, CCNOT, and $C^1(U)$



Complexity of Implementing General Controlled Operators

Theorem

Let C_0, C_1, and T be pairwise disjoint subsets of $\mathbb{Z}_n$, let $m = |T| > 0$ and assume that $T = \{i, i+1, \ldots, i+m-1\}$. Also, let U be a unitary operator on $\mathbb{H}_m$. Set $k_0 = |C_0|$, $k_1 = |C_1|$, $k = k_0 + k_1$. Then the unitary operator $C^{C_0,C_1,T}(U)$ can be implemented by a quantum circuit that uses $2k_0$ Pauli X gates, $2k-2$ CCNOT gates, one $C^1(U)$ gate, and $k-1$ ancillary and erasure gates.

Theorem

Let U be a unitary single-qubit operator, let C_0, C_1 be disjoint subsets of $\mathbb{Z}_n$, let $k_0 = |C_0|$, $k_1 = |C_1|$, $k = k_0 + k_1 < n$ and $t \in \mathbb{Z}_n \setminus (C_0 \cup C_1)$. Then the unitary operator $C^{C_0,C_1,t}(U)$ can be implemented by a quantum circuit that uses $\mathrm{O}(k)$ Pauli X, Hadamard, $\pi/8$, inverse $\pi/8$, CNOT, ancillary, and erasure gates, and four other single-qubit gates.

4.7. Perfectly universal sets of quantum gates

The Set of Two-level Operators is Perfectly Universal

Definition

Let $A \in \mathbb{C}^{(k,k)}$. Then A is called a *two-level matrix*, *two-level operator*, or *two-level gate*, if there are $i, j \in \mathbb{Z}_k$ such that for every $\vec{v} \in \mathbb{C}^k$ all entries of the vectors $\vec{v}$ and $A\vec{v}$ with indices different from i and j are equal.

$$A = \begin{pmatrix} 1 & 1 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ two-level}, \qquad B = \begin{pmatrix} 1 & 1 & 0 \\ 1 & -1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \text{ not two-level.}$$

Theorem

Let $U \in \mathbb{C}^{(k,k)}$ be unitary. Then U can be written as a product of $k(k-1)/2$ unitary two-level matrices.

Corollary

The set of all two-level unitary operators is perfectly universal for quantum computation.

Another Perfectly Universal Set of Quantum Gates

Theorem

For every two-level unitary operator U on $\mathbb{H}_n$ there is unitary single-qubit operator V such that U can be implemented by a quantum circuit that uses V and $\mathrm{O}(n^2)$ Pauli X, Hadamard, (inverse) $\pi/8$, standard CNOT, ancillary, and erasure gates, and four other single-qubit gates.

Theorem

The set that contains all rotation gates and the standard CNOT gate is perfectly universal for quantum computing.

4.8. A universal set of quantum gates

The Universal Set

Theorem

The set containing the Hadamard and $\pi/8$ gates is universal for the set of all unitary single-qubit operators.

Theorem

The set containing the Hadamard, $\pi/8$, and standard CNOT gates is universal for quantum computation.

Efficiency of Approximation - The Solovay-Kitaev Theorem

Theorem

Let G be a finite set of rotation gates containing its own inverses which is universal for the set of all rotation operators. Then for all $\epsilon \in \mathbb{R}_{>0}$ and all rotation operators U there is $l \in \mathbb{N}$ and a sequence $V_0, \ldots, V_{l-1}$ such that $l = \mathrm{O}(\log^c 1/\epsilon)$ such $E\left(U, \prod_{i=0}^{l-1} V_i\right) < \epsilon$.

4.9. Quantum Algorithms and Quantum Complexity

Families of Quantum Circuits

Definition

A *family of quantum circuits* is a sequence $(Q_n)_{n\in\mathbb{N}}$ of quantum circuits Q_n such that Q_n operates on n-qubit input registers for all $n \in \mathbb{N}$.

Definition

A quantum circuit family (Q_n) is called *P-uniform* if there is a deterministic polynomial-time algorithm that on input of 1^n, $n \in \mathbb{N}$, outputs an encoding of Q_n.

Quantum Coin Toss Algorithm

Input: $\emptyset$
Output: 0 or 1

1: $\mathrm{coinToss}$
2: $\quad |\psi\rangle \leftarrow |0\rangle$
3: $\quad b \leftarrow \mathrm{QcoinToss}\,|\psi\rangle$
4: $\quad$ **return** b
5: **end**

Quantum Algorithms

Definition

A *quantum algorithm* is a probabilistic algorithm with the following additional features.

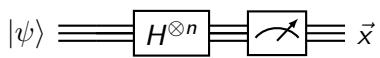
1. The algorithm may invoke elements from a P-uniform quantum circuit family. To do so, it prepares an input state for the invoked quantum circuit unless this state is already part of this circuit. The return value of the invoked quantum circuit is the outcome obtained by measuring the final state computed by it.
2. The algorithm may also invoke other quantum algorithms as subroutines.

Quantum Random String Selection

Input: $n \in \mathbb{N}$
Output: $\vec{b} \in \{0,1\}^n$

1: $\mathrm{randomString}(n)$
2: $\quad |\psi\rangle \leftarrow |0\rangle^{\otimes n}$
3: $\quad \vec{x} \leftarrow \mathrm{QrandomString}_n\, |\psi\rangle$
4: $\quad$ **return** $\vec{x}$
5: **end**

The quantum circuit used by the algorithm



The Quantum Computing Platform

The quantum computing platform provides the following perfectly universal set of quantum operators.

- All rotation gates, CNOT, the ancillary and erasure gates.

For the convenience of the exposition, it also provides the following gates that can all be implemented using $\mathrm{O}(1)$ gates from the list above.

- The Pauli X, Y, and Z gates, the Hadamard gate, the phase shift gates R_k including the phase gate $S = R_2$ and the $\pi/8$ gate R_3, as well as the Toffoli gate CCNOT.

These gates are referred to as *elementary quantum gates*.

Implementation of $C^1(U)$

Theorem

Let $n \in \mathbb{N}$ and let U be a unitary operator on $\mathbb{H}_n$. Assume that there is a quantum circuit Q that implements U and uses $k \in \mathbb{N}$ elementary and no other quantum gates. Then there is a quantum circuit Q' that implements the controlled operator $C^1(U)$ and uses $\mathrm{O}(k)$ elementary gates and no other gates.

Corollary

For any unitary elementary gate U, the controlled-U operator can be implemented using a quantum circuit that uses $\mathrm{O}(1)$ elementary quantum gates.

Time and Space Complexity of Quantum Circuits

Definition

Let Q be a quantum circuit.

1. The *running time* or *time complexity* of Q is its *size*, i.e., the number of input qubits plus the number of elementary gates used by the circuit.
2. The *space complexity* of Q is the number of input qubits plus the number of ancilla qubits used by Q.

Time and Space Complexity of Quantum Algorithms

Definition

Let (Q_n) be a P-uniform family of quantum circuits and let A be the quantum algorithm corresponding to it.

1. The *time complexity* or *running time* of A is the function $\mathrm{qTime} : \mathbb{N} \to \mathbb{N}$ that sends an input length $n \in \mathbb{N}$ to the maximum time complexity of a quantum circuit used in the execution of A with an input of length n.
2. The *space complexity* of A is the function $\mathrm{qSpace} : \mathbb{N} \to \mathbb{N}$ that sends an input length $n \in \mathbb{N}$ to the maximum space complexity of a quantum circuit used in the execution of A with an input of length n.

The complexity of quantum algorithms is defined using the corresponding concepts for probabilistic algorithms, while also accounting for the complexity of quantum subroutines. The names of the asymptotic time and space complexities for classical probabilistic algorithms apply directly to quantum algorithms. The concepts of expected running time, success probability, and its amplification all seamlessly carry over to quantum algorithms.

Quantum Complexity Classes

If $f : \mathbb{N} \to \mathbb{N}$ is a function, we say that an algorithmic problem can be solved in quantum time $\mathrm{O}(f)$ if there is a quantum Monte Carlo algorithm that solves this problem with success probability $\geq \frac{2}{3}$ and has running time $\mathrm{O}(f)$. We also say that a computational problem is solvable in quantum linear, quasilinear, quadratic, cubic, polynomial, subexponential, and exponential time, if there is a quantum Monte Carlo algorithm with this running time that solves this problem with success probability $\geq \frac{2}{3}$.

Definition

The *complexity class BQP* is the set of all languages L for which there is a quantum polynomial time Monte Carlo algorithm A that decides L and satisfies $\Pr(A(s) = 1) \geq \frac{2}{3}$ for all $s \in L$ and $\Pr(A(s) = 0) \geq \frac{2}{3}$ for all $s \in \{0,1\}^* \setminus L$.

Theorem

We have $\mathrm{P} \subset \mathrm{BPP} \subset \mathrm{BQP} \subset \mathrm{PSPACE}$.

5. The Algorithms of Deutsch and Simon

5.1. The Deutsch algorithm

The Classical Deutsch Problem

$f : \{0,1\} \to \{0,1\}$.

f *constant*, if $f(0) = f(1)$, i.e., $f(0) \oplus f(1) = 0$.
f *balanced*, if $f(0) \neq f(1)$, i.e., $f(0) \oplus f(1) = 1$.

Classical Deutsch problem: Given a black-box for f, determine $f(0) \oplus f(1)$.

In order to determine $f(0) \oplus f(1)$ classically, the black-box must be queried twice.

The Quantum Deutsch Problem

$f : \{0,1\} \to \{0,1\}$,

$U_f : \mathbb{H}_2 \to \mathbb{H}_2$, $|x\rangle\,|y\rangle \mapsto |x\rangle\,|f(x) \oplus y\rangle = |x\rangle\,X^{f(x)}\,|y\rangle$

Quantum Deutsch problem: Given a black-box for U_f, determine $f(0) \oplus f(1)$.

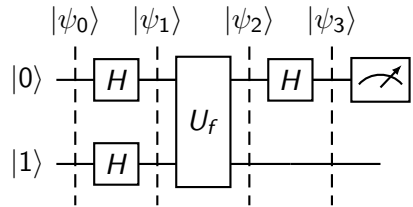
A Quantum Circuit that Solves The Deutsch Problem

$f : \{0,1\} \to \{0,1\}$

$|x_-\rangle = \dfrac{|0\rangle - |1\rangle}{\sqrt{2}}$

$U_f : |x\rangle\,|x_-\rangle \mapsto (-1)^{f(x)}\,|x\rangle\,|x_-\rangle$

Global phase shift



$|\psi_0\rangle = |0\rangle\,|1\rangle$

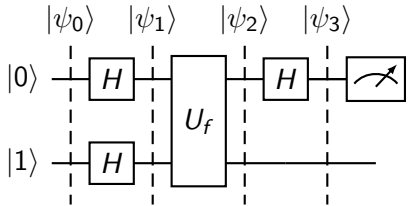
$|\psi_1\rangle = |x_+\rangle\,|x_-\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle\,|x_-\rangle + |1\rangle\,|x_-\rangle\right)$

$|\psi_2\rangle = \frac{1}{\sqrt{2}}\left(U_f\,|0\rangle\,|x_-\rangle + U_f\,|1\rangle\,|x_-\rangle\right) = \frac{1}{\sqrt{2}}\left((-1)^{f(0)}\,|0\rangle\,|x_-\rangle + (-1)^{f(1)}\,|1\rangle\,|x_-\rangle\right)$

$= \frac{(-1)^{f(0)}}{\sqrt{2}}\left(|0\rangle + (-1)^{f(0)\oplus f(1)}\,|1\rangle\right)|x_-\rangle$ Phase kickback

$|\psi_3\rangle = (-1)^{f(0)}\,|f(0)\oplus f(1)\rangle\,|x_-\rangle.$

Complexity of Solving the Deutsch Problem



$f(0) \oplus f(1)$

Theorem

The above quantum circuit gives $f(0) \oplus f(1)$ with probability 1. It uses the black-box U_f once and, in addition, three Hadamard gates.

5.2. Oracle Complexity

Oracle Complexity

The only input required by the Deutsch algorithm is the oracle U_f. However, inputs of this kind are not accounted for in the quantum algorithm concept discussed thus far. The analysis of the algorithm's time complexity includes consideration of the number of calls to this oracle.

5.3. The Deutsch-Josza Algorithm

The Classical Deutsch-Josza Problem

$f : \{0,1\}^n \to \{0,1\}$,

f constant: $f(\vec{x})$ is the same for all $\vec{x} \in \{0,1\}^n$,
f balanced: $f(\vec{x}) = 0$ for half of the $\vec{x} \in \{0,1\}^n$ and $f(\vec{x}) = 1$ for the other half.

Classical Deutsch-Josza problem: Given a black-box for f, decide whether f is constant or balanced.

Solution requires $2^{n-1} + 1$ queries of f.

The Quantum Deutsch-Josza Problem

$f : \{0,1\}^n \to \{0,1\}^n$

$U_f : |\vec{x}\rangle\,|y\rangle \mapsto |\vec{x}\rangle\,|f(x) \oplus y\rangle = |\vec{x}\rangle\,X^{f(x)}\,|y\rangle$

Quantum Deutsch-Josza problem: Given a black-box for U_f, decide whether f is constant or balanced.

Superpositions

$$\vec{x} = (x_0, \ldots, x_{n-1}) \in \{0,1\}^n$$

$$H^{\otimes n} |\vec{x}\rangle = \bigotimes_{i=0}^{n-1} \frac{|0\rangle + (-1)^{x_i}}{\sqrt{2}} = \frac{1}{\sqrt{2^n}} \sum_{\vec{z} \in \{0,1\}^n} (-1)^{\vec{x} \cdot \vec{z}} |\vec{z}\rangle$$

$$\Rightarrow H^{\otimes n} \left(\frac{1}{\sqrt{2^n}} \sum_{\vec{z} \in \{0,1\}^n} (-1)^{\vec{x} \cdot \vec{z}} |\vec{z}\rangle \right) = |\vec{x}\rangle$$

$$\Rightarrow H^{\otimes n} |0\rangle_n = \frac{1}{\sqrt{2^n}} \sum_{\vec{z} \in \{0,1\}^n} |\vec{z}\rangle, \quad H^{\otimes n} \left(\frac{1}{\sqrt{2^n}} \sum_{\vec{z} \in \{0,1\}^n} |\vec{z}\rangle \right) = \vec{0}_n,$$

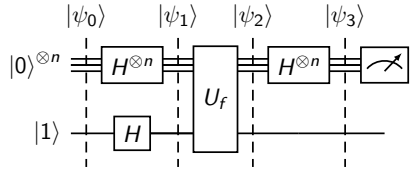
Global Phase Shift

$f : \{0,1\}^n \to \{0,1\}^n \qquad U_f : |\vec{x}\rangle\,|y\rangle \mapsto |\vec{x}\rangle\, X^{f(\vec{x})}\,|y\rangle.$

$$U_f\,|\vec{x}\rangle\,|x_-\rangle \quad = U_f\,|\vec{x}\rangle\,\tfrac{|0\rangle-|1\rangle}{\sqrt{2}} = \quad = \tfrac{1}{\sqrt{2}}\left(U_f\,|\vec{x}\rangle\,|0\rangle - U_f\,|\vec{x}\rangle\,|1\rangle\right)$$

$$= \tfrac{1}{\sqrt{2}}\left(^{f(x)}\,|0\rangle - |\vec{x}\rangle\, X^{f(x)}\,|1\rangle\right) \quad = (-1)^{f(x)}\,|\vec{x}\rangle\,|x_-\rangle.$$

Quantum Circuit that Solves The Deutsch-Josza Problem I



$$H^{\otimes n}|\vec{x}\rangle = \sum_{\vec{z}\in\{0,1\}^n} (-1)^{\vec{x}\cdot\vec{z}}|\vec{z}\rangle$$

Global phase shift: $U_f|\vec{x}\rangle|x_-\rangle = (-1)^{f(\vec{x}}|\vec{x}\rangle$.

$|\psi_0\rangle = |0\rangle_n|1\rangle$, $|\psi_1\rangle = H^{\otimes n}|0\rangle^{\otimes n}|x_-\rangle = \frac{1}{\sqrt{2^n}}\sum_{\vec{x}\in\{0,1\}^n}|\vec{x}\rangle|x_-\rangle$

$|\psi_2\rangle = \frac{1}{\sqrt{2^n}}\sum_{\vec{x}\in\{0,1\}^n} U_f|\vec{x}\rangle|x_-\rangle = \frac{(-1^{f(0)}}{\sqrt{2^n}}\sum_{\vec{x}\in\{0,1\}^n}(-1)^{f(\vec{x})\oplus f(0)}|\vec{x}\rangle|x_-\rangle$ Phase kickback

Exercise: $|\psi_3\rangle = \frac{(-1)^{f(\vec{0})}}{2^n}\sum_{\vec{z}\in\{0,1\}^n}\sum_{\vec{x}\in\{0,1\}^n}(-1)^{\vec{x}\cdot\vec{z}+f(\vec{x})\oplus f(\vec{0})}|\vec{z}\rangle|x_-\rangle$

Quantum Circuit that Solves The Deutsch Problem II

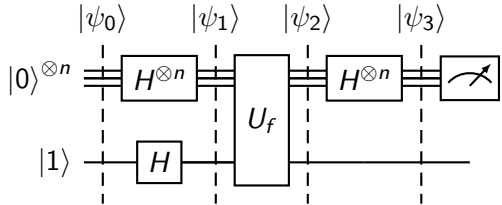
$$|\psi_3\rangle = \frac{(-1)^{f(\vec{0})}}{2^n} \sum_{\vec{z}\in\{0,1\}^n} \sum_{\vec{x}\in\{0,1\}^n} (-1)^{\vec{x}\cdot\vec{z}+f(\vec{x})\oplus f(\vec{0})} |\vec{z}\rangle |x_-\rangle$$

Amplitude of $|0\rangle_n |x_-\rangle$:

$$\frac{(-1)^{f(\vec{0})}}{2^n} \sum_{x\in\{0,1\}^n} (-1)^{f(\vec{x})\oplus f(\vec{0})} = \begin{cases} (-1)^{f(\vec{0})} & \text{if } f \text{ is constant,} \\ 0 & \text{if } f \text{ is balanced .} \end{cases}$$

Measuring $|\psi_3\rangle$ solves the problem with probability 1. Only one call of U_f required.

Complexity of Solving The Deutsch-Josza Problem



$$\begin{cases} \vec{0} & \text{if } f \text{ is constant,} \\ \vec{x} \neq \vec{0} & \text{if } f \text{ is balanced.} \end{cases}$$

Theorem

Let $n \in \mathbb{N}$, let $f : \{0,1\}^n \to \{0,1\}$ be a function that is constant or balanced, and let U_f be the unitary operator from Slide 258. Then with probability 1 the above quantum circuit returns $\vec{0}$ if f is constant and $\vec{x} \in \{0,1\}^n$, $\vec{x} \neq \vec{0}$, if f is balanced. It uses one U_f gate and $2n+1$ Hadamard gates.

5.4. Simon's Algorithm

Simon's Problem

$f : \{0,1\}^n \to \{0,1\}^n$: there is $\vec{s} \in \{0,1\}^n$, $\vec{s} \neq \vec{0}$, such that for all $\vec{x}, \vec{y} \in \{0,1\}^n$ we have $f(\vec{x}) = f(\vec{y})$ if and only if $\vec{x} = \vec{y}$ or $\vec{x} = \vec{y} \oplus \vec{s}$.

Classical Simon's problem Given a black-box for f, find $\vec{s}$.

Classical solution of Simon's problem must query f at least $2^{n-1} + 1$ times in the worst case.

Quantum Simon's problem Given a black-box for $U_f : |\vec{x}\rangle |\vec{y}\rangle \mapsto |\vec{x}\rangle |f(\vec{x}) \oplus \vec{y}\rangle$, find $\vec{s}$.

Idea for Solving of Simon's Problem

$f : \{0,1\}^n \to \{0,1\}^n$: there is $\vec{s} \in \{0,1\}^n$, $\vec{s} \neq \vec{0}$ such that $f(\vec{x}) = f(\vec{y})$ if and only if $\vec{x} = \vec{y}$ or $\vec{x} = \vec{y} \oplus \vec{s}$.

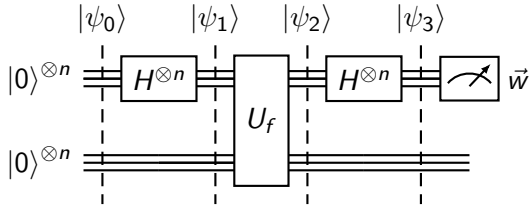
Theorem

1. $\vec{s}^{\perp}$ is a linear subspace of dimension $n-1$ of $\{0,1\}^n$.
2. Let $W = (\vec{w}_1, \ldots, \vec{w}_{n-1})$ be a basis of $\vec{s}^{\perp}$. Then $\vec{s}$ is the uniquely determined solution of the linear system $W^{\mathrm{T}}\vec{s} = \vec{0}$.

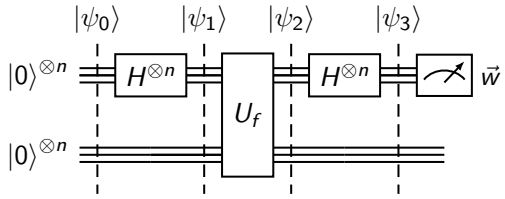
Simon's Algorithm

1: $W \leftarrow ()$
2: **for** $j = 1$ to $n - 1$ **do**
3: $\quad \vec{w}_j \leftarrow Q(U_f)$ ▷ The quantum circuit $Q(U_f)$ gives random elements in $\vec{s}^{\perp}$
4: $\quad W \leftarrow W \circ (\vec{w}_j)$
5: **end for**
6: $r \leftarrow \operatorname{rank} W$
7: **if** $r = n - 1$ **then**
8: $\quad$ Solve $W^{\mathrm{T}}\vec{s} = 0$
9: $\quad$ **return** $\vec{s}$
10: **else**
11: $\quad$ **return** "Failure"
12: **end if**

The Quantum Circuit $Q(U_f)$ that Gives Random Elements in $\vec{s}^{\perp}$



Analysis of the Quantum Circuit $Q(U_f)$ - I



I: a set of representatives of $\{0,1\}^n/\{0,\vec{s}\}$
$\Rightarrow \{0,1\}^n = \bigcup_{\vec{z}\in I}\{\vec{z}, \vec{s}\oplus\vec{z}\}$.

Lemma $H^{\otimes n}\left(\frac{|\vec{z}\rangle+|\vec{z}\oplus\vec{s}\rangle}{\sqrt{2}}\right) = \frac{1}{\sqrt{2^{n-1}}}\sum_{\vec{w}\in\vec{s}^{\perp}}(-1)^{\vec{z}\cdot\vec{w}}|\vec{w}\rangle$.

$$|\psi_1\rangle = \frac{1}{\sqrt{2^n}}\sum_{\vec{z}\in\{0,1\}^n}|\vec{z}\rangle|0\rangle^{\otimes n}$$

$$|\psi_2\rangle = U_f|\psi_1\rangle = \frac{1}{\sqrt{2^n}}\sum_{\vec{z}\in\{0,1\}^n}|\vec{z}\rangle|f(\vec{z})\rangle$$

$$|\psi_2\rangle = \frac{1}{\sqrt{2^{n-1}}}\sum_{\vec{z}\in I}\frac{|\vec{z}\rangle+|\vec{z}\oplus\vec{s}\rangle}{\sqrt{2}}|f(\vec{z})\rangle$$

$$= \frac{1}{2^{n-1}}\sum_{\vec{z}\in I}\left(\sum_{\vec{w}\in\vec{s}^{\perp}}(-1)^{\vec{z}\cdot\vec{w}}H^{\otimes n}|\vec{w}\rangle\right)|f(\vec{z})\rangle$$

$$|\psi_3\rangle = \frac{1}{2^{n-1}}\sum_{\vec{z}\in I}\left(\sum_{\vec{w}\in\vec{s}^{\perp}}(-1)^{\vec{z}\cdot\vec{w}}|\vec{w}\rangle\right)|f(\vec{z})\rangle$$

Analysis of the Quantum Circuit $Q(U_f)$ - II

Proposition

Measuring the first register of $|\psi_3\rangle = \frac{1}{2^{n-1}} \sum_{\vec{z} \in I} \left(\sum_{\vec{w} \in \vec{s}^{\perp}} (-1)^{\vec{z} \cdot \vec{w}} |\vec{w}\rangle \right) |f(\vec{z})\rangle$ gives each $\vec{w} \in \vec{s}^{\perp}$ with probability $1/2^{n-1}$.

Proof.

The observable is $\sum_{\lambda=0}^{2^n-1} \lambda |\lambda\rangle \langle\lambda| \otimes I_n$.

Measuring it gives each state

$$|\vec{w}\rangle |f(\vec{z})\rangle, \quad \vec{w} \in \vec{s}^{\perp}, \vec{z} \in I$$

with probability $1/2^{2(n-1)}$.

Since $|I| = 2^{n-1}$, each $\vec{w}$ is measured with probability $1/2^{n-1}$. □

The Success Probability and Complexity of Simon's Algorithm

Theorem

The success probability of Simon's algorithm is at least $1/4$. It requires $n-1$ applications of U_f and $\mathrm{O}(n^3)$ other operations.

Proof.

W_j: the matrix W computed in the jth iteration of the **for** loop.

Claim: $\operatorname{rank} W_j = j$ with probability $p_j = \prod_{k=n-j}^{n-1} \left(1 - \frac{1}{2^k}\right)$.

$\operatorname{rank} W_1 = 1$ with probability $\frac{2^{n-1}-1}{2^{n-1}} = 1 - \frac{1}{2^{n-1}}$.

$\operatorname{rank} W_j = j$ with probability $p_j \frac{2^{n-1}-2^{j-1}}{2^{n-1}} = p_{j-1}\left(1 - \frac{1}{2^{n-j}}\right) = p_j$.

$$p_{n-1} = \prod_{k=1}^{n-1} \left(1 - \frac{1}{2^k}\right) \underset{\text{Lemma}}{\geq} \frac{1}{4}.$$

5.5. Generalization of Simon's algorithm

Generalization of Simon's problem

$f : \{0,1\}^n \to \{0,1\}^n$: There is a subspace S of $\{0,1\}^n$ such that $f(\vec{x}) = f(\vec{y})$ if and only if $\vec{x} = \vec{y} \oplus \vec{s}$ for some $\vec{s} \in S$.

$U_f : |\vec{x}\rangle\,|\vec{y}\rangle \mapsto |\vec{x}\rangle\,|f(\vec{x}) \oplus \vec{y}\rangle$

Problem

Input: A black-box implementing U_f and the dimension m of S

Output: A basis of S.

Generalized Simon's Algorithm

1: $W \leftarrow ()$
2: **for** $j = 1$ to $n - m$ **do**
3: $\quad \vec{w_j} \leftarrow Q(U_f)$
4: $\quad W \leftarrow W \circ (\vec{w_j})$
5: **end for**
6: $r \leftarrow \operatorname{rank} W$
7: **if** $r = m$ **then**
8: $\quad$ Find a basis B of the kernel of W
9: $\quad$ **return** B
10: **else**
11: $\quad$ **return** "Failure"
12: **end if**

Complexity of the Generalized Simon's Algorithm

Theorem

Algorithm **??** returns a basis of the hidden subgroup S from the generalization of Simon's problem with probability at least $1/4$. It uses m applications of U_f and $\mathrm{O}(n^3)$ other operations.

6. Shor's Algorithms

6.1. Idea of Shor's Factoring Algorithm

The Integer Factorization Problem

Given an odd composite integer N, find a proper divisor of N.

Best-known classical and fully analyzed Monte Carlo algorithm: subexponential complexity

$$e^{(1+o(1))(\log N \log\log N)^{1/2}}.$$

Best-known heuristic Monte Carlo algorithm: subexponential complexity

$$e^{(c+o(1))(\log N)^{1/3}(\log\log N)^{2/3}}, \quad c = \sqrt[3]{64/9}.$$

Factoring Using the Order of a mod N

Find $a \in \mathbb{Z}_N$ which has one of the following properties:

(1) $\gcd(a, N) > 1 \Rightarrow \gcd(a, N)$ is a proper divisor of N.

(2) Order r of a modulo N is even and $a^{r/2} \not\equiv -1 \bmod N$

$\Rightarrow N$ divides $a^r - 1 = (a^{r/2} - 1)(a^{r/2} + 1)$ but neither of the factors

$\Rightarrow \gcd(a^{r/2} - 1, N)$ is a proper divisor of N.

How to find the order r of a modulo N?

$n = \lceil 2 \log_2 N \rceil + 1$, then $2^n \geq 2r^2$.

$$U_a : |x\rangle_n \mapsto \begin{cases} |ax \bmod N\rangle_n & \text{if } 0 \leq x < N, \\ |x\rangle_n & \text{if } N \leq x < 2^n \end{cases}$$

$0 \leq k < r$:

$|u_k\rangle = \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} e^{-2\pi i \frac{k}{r} s} |a^s \bmod N\rangle_n$ is an eigenstate of U_a with eigenvalue $e^{2\pi i \frac{k}{r}}$.

Quantum circuit gives $x \in \mathbb{Z}_{2^n}$: $2\pi \frac{x}{2^n}$ is good approximation of phase $2\pi \frac{k}{r}$ of an eigenvalue.

$\Rightarrow \frac{x}{2^n}$ is good approximation of $\frac{k}{r}$ and continued fraction of $\frac{x}{2^n}$ gives r.

6.2. The Quantum Fourier Transform

Definition

Definition: $\omega \in \mathbb{R}$: $|\psi_n(\omega)\rangle = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{2\pi i y\omega} |y\rangle_n$.

Exercise: $|\psi_n(\omega)\rangle = \bigotimes_{j=0}^{n-1} \frac{|0\rangle + e^{2\pi i \cdot 2^{n-j-1}\omega}|1\rangle}{\sqrt{2}}$.

Definition: $\mathrm{QFT}_n |x\rangle_n = \left|\psi_n\left(\frac{x}{2^n}\right)\right\rangle = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{2\pi i \frac{x}{2^n} y} |y\rangle_n$

Exercise $\Rightarrow$ $\mathrm{QFT}_n |x\rangle_n = \bigotimes_{j=0}^{n-1} \frac{|0\rangle + e^{2\pi i \frac{x}{2^{j+1}}}}{\sqrt{2}} = \bigotimes_{j=0}^{n-1} \frac{|0\rangle + e^{2\pi i \cdot 0.x_{n-j-1}x_{n-j-2}\cdots x_{n-1}}|1\rangle}{\sqrt{2}}$.

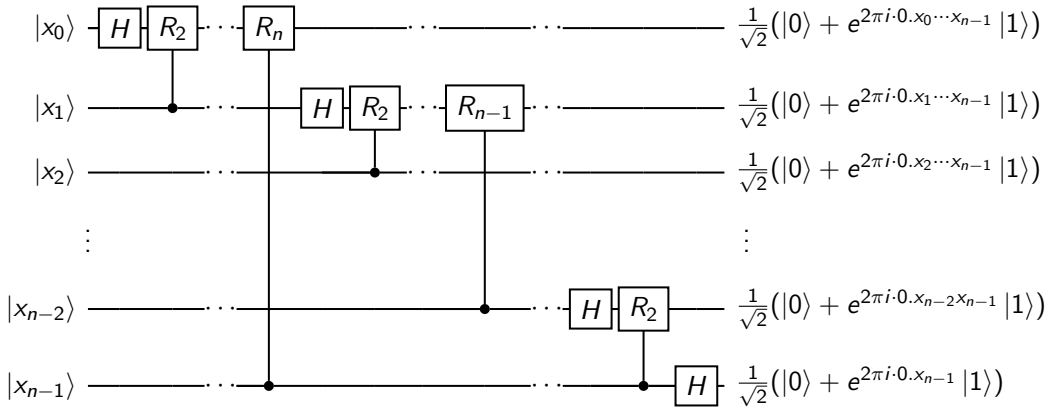
Properties of the QFT_n

The Quantum Fourier Transform QFT_n is a unitary operator on $\mathbb{H}_n$.

$$\mathrm{QFT}_n^{-1} |x\rangle_n = \left|\psi_n\left(\tfrac{-x}{2^n}\right)\right\rangle = \tfrac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{-2\pi i \frac{x}{2^n} y} |y\rangle_n$$

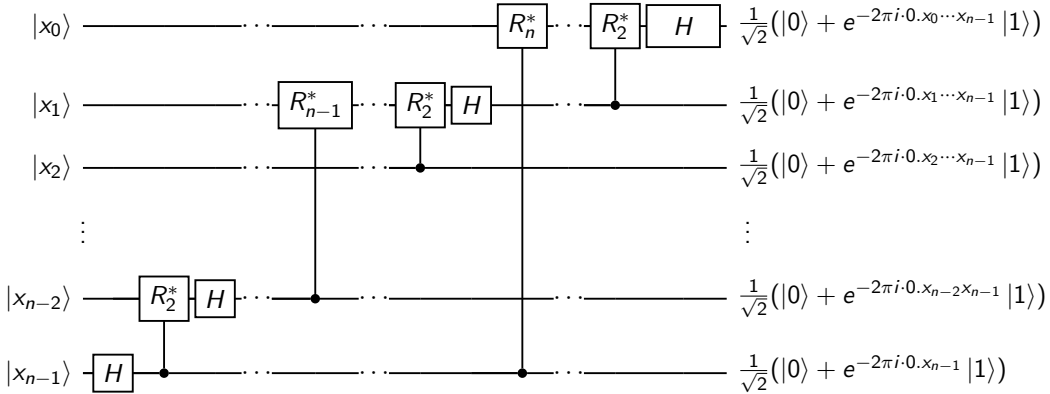
$$= \bigotimes_{j=0}^{n-1} \frac{|0\rangle + e^{-2\pi i \frac{x}{2^j}}}{\sqrt{2}} = \bigotimes_{j=0}^{n-1} \frac{|0\rangle + e^{-2\pi i \cdot 0.x_{n-j-1}x_{n-j-2}\cdots x_{n-1}}|1\rangle}{\sqrt{2}}.$$

Quantum Circuit that Implements QFT_n



$$R_k : \mathbb{H}_1 \to \mathbb{H}_1, \quad |0\rangle \mapsto |0\rangle, \quad |1\rangle \mapsto e^{2\pi i \cdot \frac{1}{2^k}} |1\rangle$$

Quantum Circuit that Implements QFT^{-1}



6.3. Quantum Phase Estimation

The problem

$m, n \in \mathbb{N}$, U unitary operator on $\mathbb{H}_m$, $|\psi\rangle$ eigenstate of U.

Eigenvalue associated with $|\psi\rangle$: $e^{2\pi i\omega}$, $\omega \in \mathbb{R}$ is unique modulo 1.

Goal: quantum circuit that gives $x \in \mathbb{Z}_{2^n}$ such that $\frac{x}{2^n}$ is a good approximation of ω.

Approximation quality of $\frac{x}{2^n}$: $\min\{\left|\omega - \frac{x}{2^n} - z\right| : z \in \mathbb{Z}\}$

Definition: $\Delta(\omega, n, x) = \omega - \frac{x}{2^n} - z$ such that $\left|\omega - \frac{x}{2^n} - z\right|$ is minimal.

Approximation quality: $|\Delta(\omega, x, n)|$.

Phase estimation problem: Given a black-box for U, m, n and $|\psi\rangle$, find $x \in \mathbb{Z}_{2^n}$ with $|\Delta(\omega, x, n)| \leq \frac{1}{2^n}$

Solving the phase estimation problem

Recall: $\omega \in \mathbb{R}$, $x \in \mathbb{Z}_{2^n}$, $|\psi_n(\omega)\rangle = \bigotimes_{j=0}^{n-1} \frac{|0\rangle + e^{2\pi i \cdot 2^{n-j-1}\omega}|1\rangle}{\sqrt{2}}$, $\left|\psi_n\left(\frac{x}{2^n}\right)\right\rangle = \mathrm{QFT}_n |x\rangle_n$.
$\Rightarrow$ Measuring $\mathrm{QFT}_n^{-1}\psi_n\left(\frac{x}{2^n}\right) = |x\rangle_n$ gives x with probability 1.

Theorem

$\omega \in \mathbb{R}$, $2^n\omega \notin Z$. Measuring $\mathrm{QFT}_n^{-1}|\psi_n(\omega)\rangle$ gives

1. any $x \in \mathbb{Z}_{2^n}$ with probability $p(x) = \frac{1}{2^{2n}} \frac{\sin^2(\pi(2^n\omega - x))}{\sin^2(\pi(\omega - x/2^n))}$,
2. with probability at least $4/\pi^2$ an $x \in \mathbb{Z}_{2^n}$ such that $\Delta(\omega, n, x) \leq \frac{1}{2^{n+1}}$,
3. with probability at least $8/\pi^2$ an $x \in \mathbb{Z}_{2^n}$ such that $\Delta\omega, n, x) \leq \frac{1}{2^n}$.

Strategy for approximation the phase of eigenvalue $e^{2\pi i \cdot \omega}$ of eigenstate $|\psi\rangle$ of U:

Construct $|\psi_n(\omega)\rangle$, measure $\mathrm{QFT}_n^{-1}|\psi_n(\omega\rangle$.

Open problem

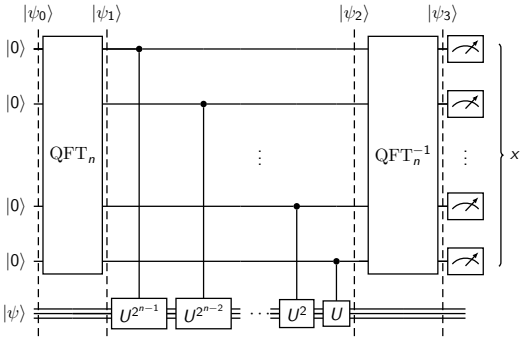
For $0 < \theta < 1$ the function

$$f(x) = \sin^2(\pi\theta)\left(\frac{1}{\theta^2} + \frac{1}{(1-\theta)^2}\right) \tag{6.1}$$

attains its minimum 8 for $\theta = \frac{1}{2}$.

Phase estimation circuit

$$|\psi_n(\omega)\rangle = \bigotimes_{j=0}^{n-1} \frac{|0\rangle + e^{2\pi i \cdot 2^{n-j-1}\omega}|1\rangle}{\sqrt{2}}.$$



$$|\psi_0\rangle = |0\rangle^{\otimes n} |\psi\rangle \qquad |\psi_1\rangle = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)^{\otimes n} |\psi\rangle$$

$$C\text{-}U^{2^{n-j-1}} \frac{|0\rangle + |1\rangle}{\sqrt{2}} |\psi\rangle$$
$$= \frac{|0\rangle + \left(e^{2\pi i\omega}\right)^{2^{n-j-1}} |1\rangle}{\sqrt{2}} |\psi\rangle$$

$$|\psi_2\rangle = \bigotimes_{j=0}^{n-1} \frac{|0\rangle + e^{2\pi i 2^{n-j-1}\omega}|1\rangle}{\sqrt{2}} |\psi\rangle$$

$$= |\psi_n(\omega)\rangle |\psi\rangle$$

$$|\psi_3\rangle = \left(\mathrm{QFT}_n^{-1} |\psi_n(\omega)\rangle\right) |\psi\rangle.$$

6.4. Continued Fractions

A *simple finite continued fraction* is a sequence $(a_0, a_1 \ldots a_n) \in \mathbb{N}_0 \times \mathbb{N}^n$ where $n \in \mathbb{N}_0$. It represents the nonnegative rational number

$$[a_0, a_1 \ldots, a_n] \mathrel{\widehat{=}} a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \ddots + \cfrac{1}{a_{n-1} + \cfrac{1}{a_n}}}}$$

.

$$[1, 6, 2] \mathrel{\widehat{=}} 1 + \cfrac{1}{6 + \cfrac{1}{2}} = 1 + \cfrac{1}{\cfrac{13}{2}} = 1 + \frac{2}{13} = \frac{15}{13}$$

Continued Fraction Algorithm

Continued fraction expansion of $\frac{15}{13}$:

$15 = 1 \cdot 13 + 2$, $13 = 6 \cdot 2 + 1$, $2 = 2 \cdot 1 + 0$, $\frac{15}{13} \mathrel{\widehat{=}} [1, 6, 2]$.

$p \in \mathbb{N}_0$, $q \in \mathbb{N}$, continued fraction expansion of $\frac{p}{q}$:

Initial remainders: $r_{-1} = p$, $r_0 = q$.

Division with remainder: $r_{i-1} = a_i r_i + r_{i+1}$, $a_i = \left\lfloor \frac{r_{i-1}}{r_i} \right\rfloor$,

until $r_{i+1} = 0$. Then $n = i$ and $\frac{p}{q} \mathrel{\widehat{=}} [a_0.a_1, \ldots, a_n]$.

Complexity: $l = \max\{\text{bitLength}(p), \text{bitLength}(q)\})$, $n = \mathrm{O}(l)$, running time $\mathrm{O}(l^3)$.

Convergents

convergents of $[a_0, \ldots, a_n]$: the rational numbers represented by $[a_0, \ldots, a_i]$, $0 \leq i \leq n$.

Convergents of $[1, 6, 2] \widehat{=} \frac{15}{3}$:

$$[1] \widehat{=} 1,$$

$$[1, 6] \widehat{=} 1 + \frac{1}{6} = \frac{7}{6},$$

$$[1, 6, 2] \widehat{=} 1 + \frac{1}{6 + \frac{1}{2}} = 1 + \frac{1}{\frac{13}{2}} = 1 + \frac{2}{13} = \frac{15}{13}.$$

Convergents of Continued Fractions

Definition

Let $[a_0, \ldots, a_n]$ be a continued fraction. Then for $0 \leq i \leq n$ the rational numbers $[a_0, \ldots, a_i]$ are called its *convergents*.

Theorem

Let $\alpha \in \mathbb{Q}_{>0}$ and let $p, q \in \mathbb{N}$ such that

$$\left| \frac{p}{q} - \alpha \right| \leq \frac{1}{2q^2}$$

Then $\frac{p}{q}$ is a convergent of the continued fraction expansion of α.

6.5. Order finding

The problem

Given $a \in \mathbb{Z}_N^*$ find $r = \min\{t \in \mathbb{N} : a^t \equiv 1 \bmod N\}$.

$$U_a : |x\rangle_n \mapsto \begin{cases} |ax \bmod N\rangle_n & \text{if } 0 \leq x < N, \\ |x\rangle_n & \text{if } N \leq x < 2^n \end{cases}$$

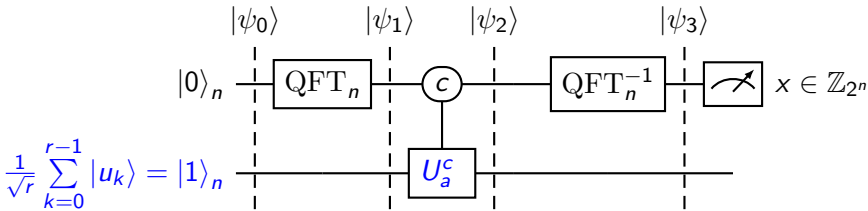
$0 \leq k < r$:

$|u_k\rangle = \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} e^{-2\pi i \frac{k}{r} s} |a^s \bmod N\rangle_n$ is an eigenstate of U_a with eigenvalue $e^{2\pi i \frac{k}{r}}$.

The eigenstates $|u_k\rangle$ cannot be prepared ☹.

Exercise: $\sum_{j=0}^{r-1} |u_k\rangle = |1\rangle_n$. $|1\rangle_n$ can be prepared!

The Quantum Circuit Q_a



Theorem

1. With probability at least $\frac{8}{\pi^2}$ the quantum circuit Q_a gives $x \in \mathbb{Z}_{2^n}$ such that the continued fraction expansion of $\frac{x}{2^n}$ has a convergent $\frac{k}{r}$ with $k \in \mathbb{Z}_r$.
2. If this happens, $\frac{k}{r}$ is the uniquely determined convergent $\frac{p}{q}$ with

$$\left| \frac{x}{2^n} - \frac{p}{q} \right| \leq \frac{1}{2^n}.$$

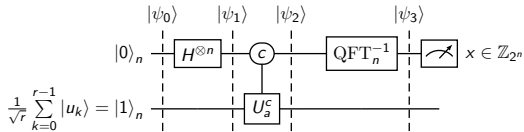
The Algorithm

1. Run the quantum circuit Q_a and obtain $x \in \mathbb{Z}_{2^n}$.
2. Compute the continued fraction expansion of $\frac{x}{2^n}$.
3. If $|\frac{x}{2^n} - \frac{p}{q}| \leq \frac{1}{2^n}$ and $q < 2^n$ for a convergent $\frac{p}{q}$, set $r = q$.
4. If $a^r \equiv 1 \bmod N$, return r else return "FAILURE".

Analysis of the Algorithm - Overview

1. Prove the theorem on Frame 298.
2. If Q_a gives x with $|\frac{x}{2^n} - \frac{k}{r}| \leq \frac{1}{2^n}$ for some $k \in \mathbb{Z}_r$, then the CFA finds $p \in \mathbb{N}_0$, $q \in \mathbb{N}$ with $\gcd(p, q) = 1$ such that $\frac{p}{q} = \frac{k}{r}$. If $\gcd(k, r) = 1$, then $r = q$.
3. Determine the probability for $\gcd(k, r) = 1$.
4. Use the above results to analyze the success probability and running time.

Sucess probability of the order finding algorithm I



$$|\psi_0\rangle = \tfrac{1}{\sqrt{2}} \sum_{k=0}^{r-1} |0\rangle^{\otimes n} |u_k\rangle, \quad |\psi_1\rangle = \tfrac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \left(\tfrac{|0\rangle+|1\rangle}{\sqrt{2}} \right)^{\otimes n} |u_k\rangle,$$

$$|\psi_2\rangle = \tfrac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \left|\psi_n\left(\tfrac{k}{r}\right)\right\rangle |u_k\rangle, \quad |\psi_3\rangle = \sum_{k=0}^{r-1} \tfrac{1}{\sqrt{r}} \left(\mathrm{QFT}_n^{-1} \left|\psi_n\left(\tfrac{k}{r}\right)\right\rangle\right) |u_k\rangle.$$

$(|u_k\rangle)$ orthonormal $\Rightarrow \forall k \in \mathbb{Z}_r$: $\Pr\left[\left|\Delta\left(\tfrac{k}{r}, n, x\right)\right| \leq \tfrac{1}{2^n}\right] \geq \tfrac{8}{r\pi^2}$

$\Rightarrow \forall k \in \mathbb{Z}_r$: $\Pr\left[\tfrac{k}{r} \text{ is a convergent of } \tfrac{1}{2^n}\right] \geq \tfrac{8}{r\pi^2}$

Success Probability of the Order Finding Algorithm II

$\forall k \in \mathbb{Z}_r$: $\Pr\left[\frac{k}{r} \text{ is a convergent of } \frac{1}{2^n}\right] \geq \frac{8}{r\pi^2}$

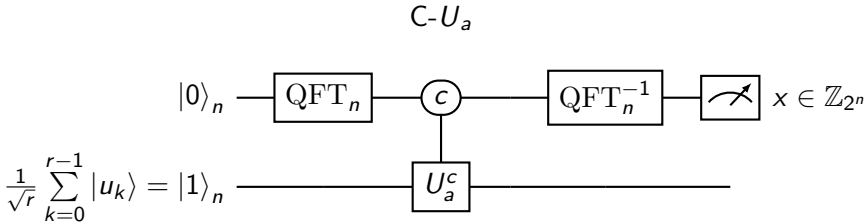
Convergents $\frac{p}{q}$ have a reduced representation. Probability that $\frac{k}{r}$ is convergent with $\gcd(k, r) = 1$?

$\varphi(n) = |\{k \in \mathbb{Z}_r : \gcd(k, r) = 1\}| > \frac{\log 2}{2} \frac{r}{\log r}$

$2^n > 2r^2 \Rightarrow \frac{1}{\log r} > \frac{2r}{\log 2} \Rightarrow \varphi(n) > \frac{r}{n}$

Success probability $\geq \frac{8}{n\pi^2}$.

Bottleneck of the Order Finding Algorithm



Bottleneck: Implementation of

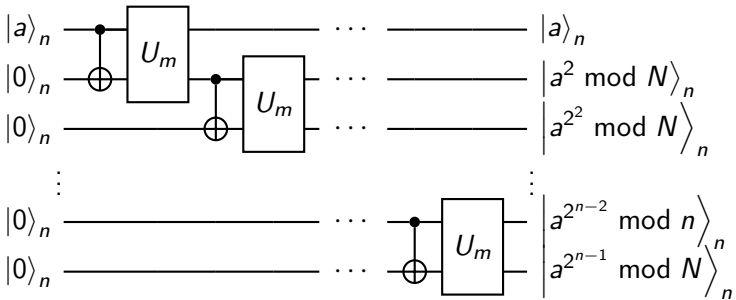
$$\text{C-}U_a \,|c\rangle_n \,|t\rangle_n = \begin{cases} |c\rangle_n \,|a^c t \bmod N\rangle_n & \text{if } 0 \leq t < N, \\ |t\rangle_n & \text{if } N \leq t < 2^n. \end{cases}$$

The Operators U_m and U_p

$$U_m |x\rangle |y\rangle = \begin{cases} |x\rangle |xy \bmod N\rangle & \text{if } (x,y) \in \mathbb{Z}_N^2 \wedge \gcd(y,N) = 1, \\ |x\rangle |y\rangle & \text{if } (x,y) \notin \mathbb{Z}_N^2 \vee \gcd(y,N) > 1. \end{cases}$$

Can be implemented by a quantum circuit of size $\mathrm{O}(n^2)$

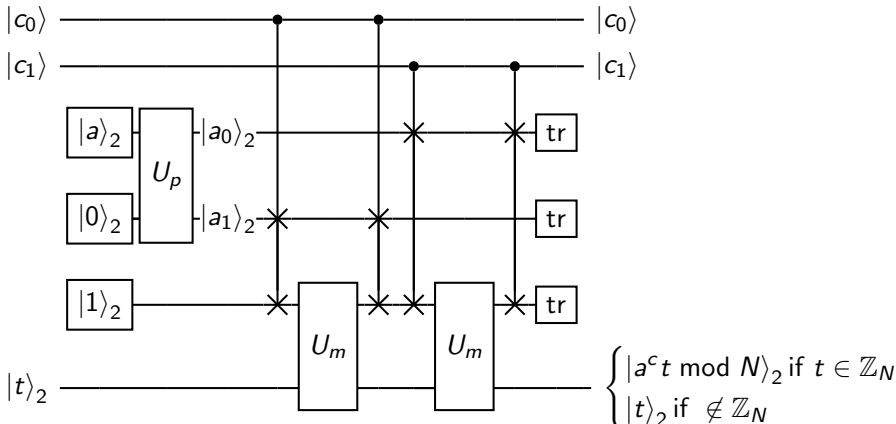
U_p:



Size: $\mathrm{O}(n^3)$

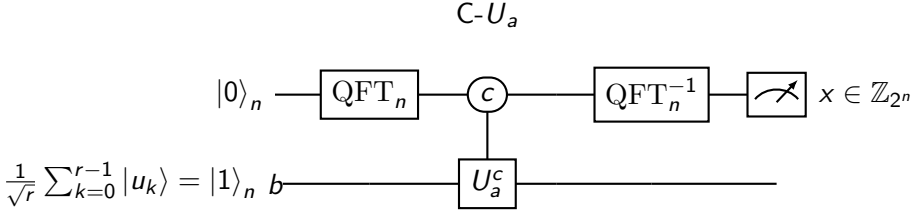
Quantum Circuit that Implements C-U_a

C-U_a:



Size: $\mathrm{O}(n^3)$

Complexity of the Order Finding Algorithm



$n = \lceil 2 \log_2 N \rceil + 1 \Rightarrow 2^n > 2r^2$, $n = \mathrm{O}(\log N)$.

$\mathrm{QFT}_n, \mathrm{QFT}_n^{-1}$: $\mathrm{O}(n^2) = \mathrm{O}((\log N)^2)$, C-$U_a$: $\mathrm{O}(n^3) = \mathrm{O}((\log N)^3)$, total: $\mathrm{O}((\log N)^3)$

Success probability: $\frac{4}{n\pi^2}$, repeat $\frac{(\log 2) n \pi^2}{4}$ times to obtain success probability $\geq \frac{1}{2}$.

Running time: $\mathrm{O}((\log N)^4)$.

6.6. Integer Factorization II

The problem

Given an odd composite integer N, find a proper divisor of N.

Find $a \in \mathbb{Z}_N$ which has one of the following properties:

(1) $\gcd(a, N) > 1 \Rightarrow \gcd(a, N)$ is a proper divisor of N.

(2) Order r of a modulo N is even and $a^{r/2} \not\equiv -1 \bmod N$

N divides $a^r - 1 = (a^{r/2} - 1)(a^{r/2} + 1)$

$\Rightarrow \gcd(a^{r/2} - 1, N)$ proper divisor of N.

Quantum Factoring Algorithm

1. $a = \mathrm{randomInt}(N)$
2. If $d = \gcd(a, N) > 1$, return d.
3. $r = \mathrm{FindOrder}(N, a)$.
4. If $r \in 2\mathbb{N}$ and $d = \gcd(a^{r/2} - 1, N) > 1$, return d.
5. Return "Failure".

Analysis

Theorem

The probability that the order r of a random $a \in \mathbb{Z}_N^*$ modulo N is even and $a^{r/2} \not\equiv -1 \bmod N$ is at least $\frac{1}{2}$.

Two success cases:

Case I: $\gcd(a, r) > 1$. Probability: $\frac{N-\varphi(N)}{N}$.

Case II: $\gcd(a, N) = 1$, $r = \operatorname{order}_N(a)$ is found and even, and $a^{r/2} \not\equiv -1 \bmod N$. This happens with probability $\geq \frac{\varphi(N)}{N}\frac{1}{4}$.

Total success probability: $\frac{4N-4\varphi(N)+\varphi(N)}{4N} = 1 - \frac{3\varphi(N)}{4N} \geq \frac{1}{4}$.

6.7. Discrete Logarithm Computation

The Problem

Given an odd integer N, $a, b \in \mathbb{Z}_N^*$ such that $b \equiv a^t \bmod N$ where $t \in \mathbb{Z}_r$ and r is the order of a modulo N

Find t

Reduction to basis elements a of prime order.

1. The quantum factoring algorithm finds the prime number decomposition of $\varphi(N)$ in polynomial time.
2. The *Pohlig-Hellman algorithm* reduces the general DL problem to the problem of computing discrete logarithms for basis elements a of prime order.

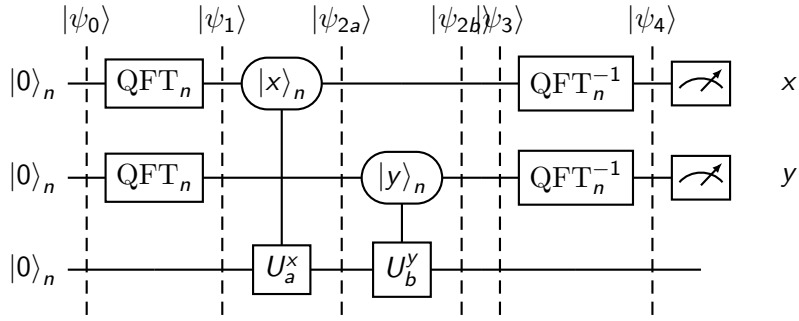
Idea of the Algorithm

$b \equiv a^t \bmod N$, $r = \operatorname{order}_N a \in \mathbb{P}$, $t \in \mathbb{Z}_r^*$. Find t!

$$U_c : |x\rangle_n \mapsto \begin{cases} |cx \bmod N\rangle_n & \text{if } 0 \leq x < N, \\ |x\rangle_n & \text{if } N \leq x < 2^n, \end{cases} \qquad |u_k\rangle = \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} e^{-2\pi i \frac{k}{r} s} |a^s \bmod N\rangle_n.$$

1. Quantum order computation gives r which is also the order of b modulo N.
2. $|u_k\rangle$ is an eigenstate of U_a with eigenvalue $e^{2\pi i \frac{k}{r}}$.
3. $|u_k\rangle$ is an eigenstate of $U_{a^t} = U_b$ with eigenvalue $e^{2\pi i \frac{tk}{r}}$.
4. Simultaneous quantum phase estimation gives $\frac{k}{r}$ and $\frac{kt}{r}$ for some $k \in \mathbb{Z}_r^*$ and thus k and $l = kt \bmod r$.
5. $t \equiv k^{-1} l \bmod r$.

Quantum Circuit for DL Computation



The Quantum Discrete Logarithm Algorithm

1. $n = \lfloor \log r \rfloor + 1$
2. Apply the quantum circuit from Frame 314 and obtain $(x, y) \in \mathbb{Z}_{2^n}^2$
3. $k = \lfloor xr/2^n \rceil \bmod r$
4. If $k \neq 0$ then set $l = \lfloor yr/2^n \rceil \bmod r$, $t = lk^{-1} \bmod r$, retuen t
5. return "FAILURE"

7. The Grover Search Algorithm

Classical Search Problem

Classical: Given black-box for $f : \{0,1\}^n \to \{0,1\}^n$, find $\vec{x} \in \{0,1\}^n$ with $f(x) = 1$.

Number of function calls required in the worst case $\geq 2^n - 1$.

Quantum:

Given a black-box for

$$U_f : \mathbb{H}_n \otimes \mathbb{H}_1 \to \mathbb{H}_n \otimes \mathbb{H}_1, \quad |\vec{x}\rangle |y\rangle \mapsto |\vec{x}\rangle |f(\vec{x}) \oplus y\rangle = |\vec{x}\rangle X^{f(\vec{x})} |y\rangle ,$$

find $\vec{x} \in \{0,1\}^n$ with $f(x) = 1$.

Idea of the Algorithm

$f : \{0,1\}^n \to \{0,1\}$, $M = \left|f^{-1}\{1\}\right|$.

$$|s_0\rangle = \frac{1}{\sqrt{N-M}} \sum_{\vec{x}\in\{0,1\}^n,\, f(\vec{x})=0} |\vec{x}\rangle, \qquad |s_1\rangle = \frac{1}{\sqrt{M}} \sum_{\vec{x}\in\{0,1\}^n,\, f(\vec{x})=1} |\vec{x}\rangle\,,$$

$$|s\rangle = \frac{1}{\sqrt{N}} \sum_{\vec{x}\in\{0,1\}^n} |\vec{x}\rangle = \sqrt{\frac{N-M}{N}}\,|s_0\rangle + \sqrt{\frac{M}{N}}\,|s_1\rangle.$$

(1) Construct $|s\rangle = H^{\otimes n}\,|0\rangle^{\otimes n}$.

(2) Measure $|s\rangle$: gives $\vec{x} \in \{0,1\}^n$ with $f(\vec{x}) = 1$ with probability $\frac{M}{N}$.

For small M amplitude of $|s_1\rangle$ is amplified.

Amplitude Amplification

$$|s_0\rangle = \frac{1}{\sqrt{N-M}} \sum_{\vec{x}\in\{0,1\}^n,\, f(\vec{x})=0} |\vec{x}\rangle\,, \quad |s_1\rangle = \frac{1}{\sqrt{M}} \sum_{\vec{x}\in\{0,1\}^n,\, f(\vec{x})=1} |\vec{x}\rangle\,.$$

$$U_1 = I_n - 2\,|s_1\rangle\,\langle s_1|\,, \quad U_s = 2\,|s\rangle\,\langle s| - I_n,$$

Grover iterator: $G = U_s U_1$, $\quad \theta = \arcsin\sqrt{\frac{M}{N}}$.

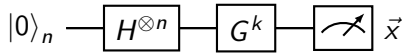
Theorem: $\alpha \in \mathbb{R}$: $G\left(\cos\alpha\,|s_0\rangle + \sin\alpha\,|s_1\rangle\right) = \cos(\alpha+2\theta)\,|s_0\rangle + \sin(\alpha+2\theta)\,|s_1\rangle$.

$$|s\rangle = \cos\theta\,|s_0\rangle + \sin\theta\,|s_1\rangle \qquad G^k\,|s\rangle = \cos(2k+1)\theta\,|s_0\rangle + \sin(2k+1)\theta\,|s_1\rangle.$$

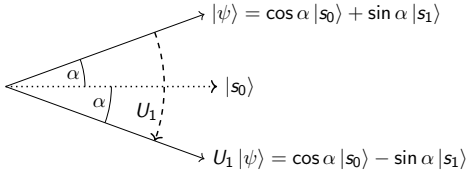
Choose k such that $(2k+1)\theta$ is closest to $\frac{\pi}{2}$: $k = \left\lfloor \frac{\pi}{4\theta} - \frac{1}{2} \right\rceil$

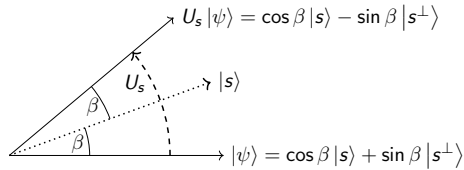
$\Rightarrow$: $k \le \frac{\pi}{4}\sqrt{\frac{N}{M}}$, $\quad \sin^2(2k+1)\theta \ge 1 - \frac{\pi^2}{4}\frac{M}{N}$.

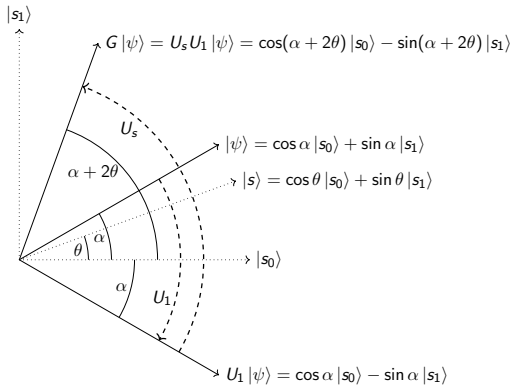
Grover Serach Quantum Circuiit



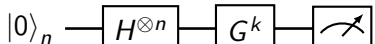
Proof of the Theorem

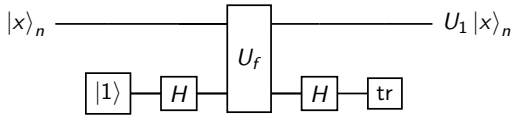




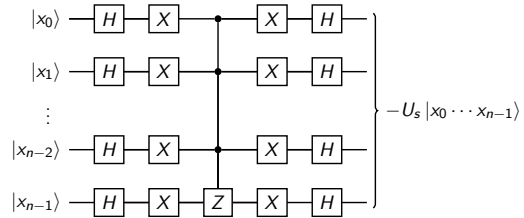


Complexity of the Grover Algorithm





$$U_1 = I_n - 2\,|s_1\rangle\,\langle s_1|$$



$$-U_s = -(2\,|s\rangle\,\langle s| - I_n).$$

Theorem The complexity of the Grover search slgorithm is $N^{1/2+o(1)}$.

Bibliography


Sanjeev Arora and Boaz Barak, *Computational complexity: a modern approach*, Cambridge University Press, Cambridge ; New York, 2009, OCLC: ocn286431654.