Quantum Complexity Theory

Markus Lohrey

Universität Siegen

Summer semester 2024

General information

On the web page

`https://www.eti.uni-siegen.de/ti/lehre/sommer_2024/quantumct/index.html`

you find all informations, in particular:

- most recent version of the slides,
- exercise sheets,
- literature recommendations.

Notice: The lecture slides may not be suitable for self studying (and may at some point be incomplete)!

Basic notations

A bit string is a sequence $b_1 b_2 \cdots b_n$ of bits $b_1, b_2, \ldots, b_n \in \{0,1\}$.

The length of the bit string $u = b_1 b_2 \cdots b_n$ is $|u| = n$.

The set of all bit strings is denoted with $\{0,1\}^*$; the set of all bit strings of length n is denoted with $\{0,1\}^n$.

A language L is a subset of $\{0,1\}^*$.

Complexity theory investigates the computational resources needed to check, whether a given $u \in \{0,1\}^*$ belongs to a certain language L.

Often we are actually interested in sets of other finite objects (e.g. numbers, matrices, finite graphs, etc.) instead of bit strings.

In such situations we assume that these objects are suitably encoded by bit strings.

Turing machines

Classical complexity theory is usually formalized using Turing machines.

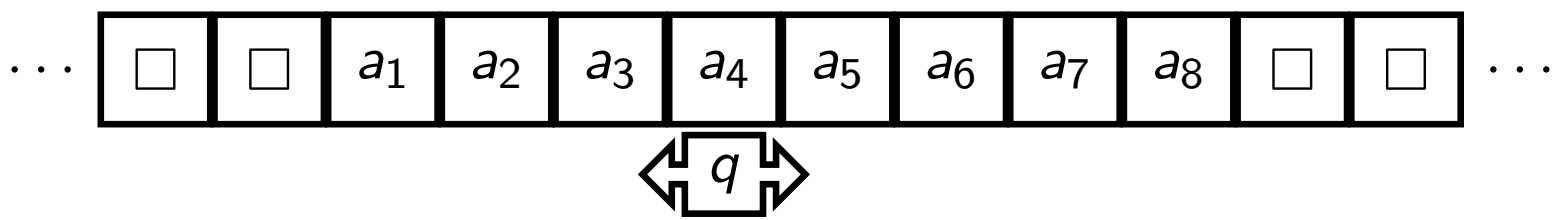
A deterministic Turing machine is a tuple $M = (Q, \Gamma, q_0, q_Y, q_N, \delta)$, where

- Q is the finite set of (control) states,
- Γ is the finite tape alphabet with $0, 1, \square \in \Gamma$,
- $\square$ is the blank symbol,
- 0 and 1 are the input symbols,
- $q_0 \in Q$ is the initial state,
- $q_Y \in Q$ is the accepting state,
- $q_N \in Q$ is the rejecting state, and
- $\delta : (Q \setminus \{q_Y, q_N\}) \times \Gamma \to Q \times \Gamma \times \{-1, 0, 1\}$ is the transition function.

We only consider deterministic Turing machines in this lecture and will mostly omit "deterministic" in the following.

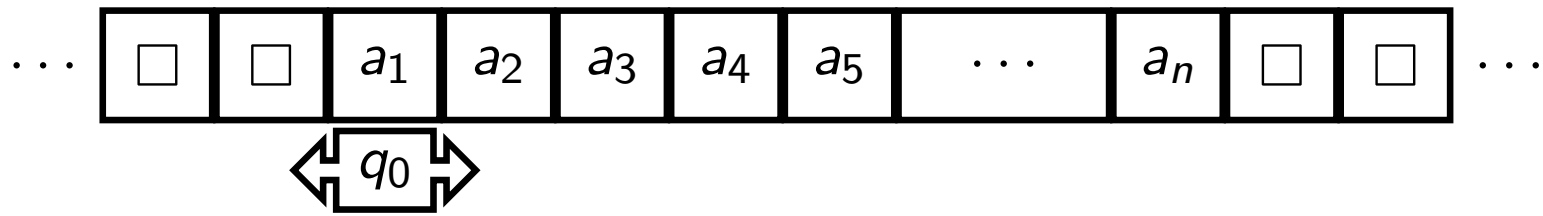
Intuition of Turing machines

- The Turing machine works on an 2-sided infinite tape of cells that contain tape symbols from Γ.
- Only finitely many cells contain a tape symbol from $\Gamma \setminus \{\square\}$.
- There is a read-write head that scans at each time instant a certain cell of the tape.
- Moreover, at each time instant the machine is in a certain control state $q \in Q$.



Intuition of Turing machines

For an input string $u = a_1 a_2 \cdots a_n$ with $a_i \in \{0, 1\}$ the machine is started in the following configuration, called the initial configuration for u:



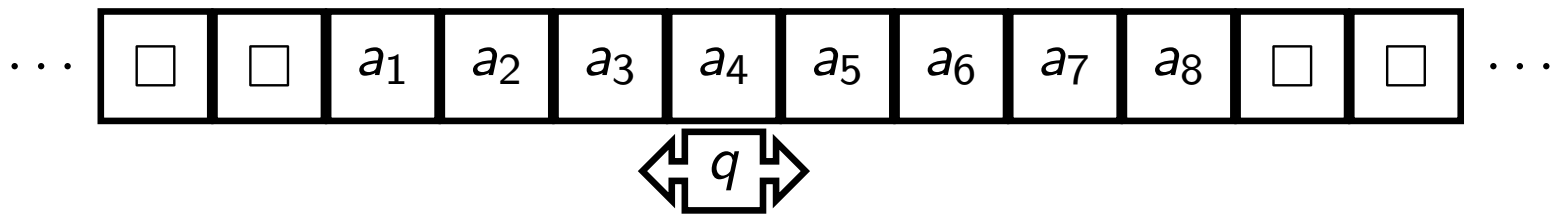
The machine moves on according to the transition function δ:

If the current control state is $q \in Q$, the currently scanned tape cell contains the symbol $a \in \Gamma$, and $\delta(q, a) = (p, b, d)$, then the machine executes the following steps:

- Replace the symbol a in the current cell by b.
- Change into control state p.
- If $d = -1/d = 1$ move the read-write head one cell to the left/right (no movement if $d = 0$).

Intuition of Turing machines

Example: Assume that $\delta(q, a_4) = (p, b, -1)$ and $\delta(p, a_3) = (r, c, 1)$:

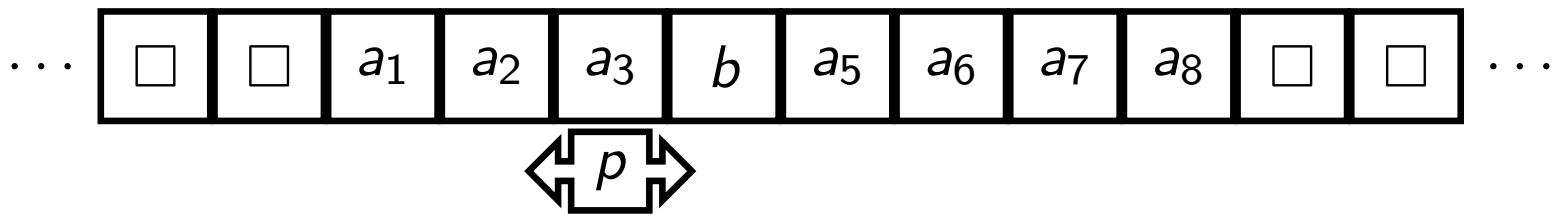


The machine stops when it reaches the accepting state q_Y or the rejecting state q_N.

The language $L(M)$ accepted by the machine M consists of all strings $u \in \{0, 1\}^*$ such that M finally reaches the accepting state q_Y when started in the initial configuration for u.

Intuition of Turing machines

Example: Assume that $\delta(q, a_4) = (p, b, -1)$ and $\delta(p, a_3) = (r, c, 1)$:

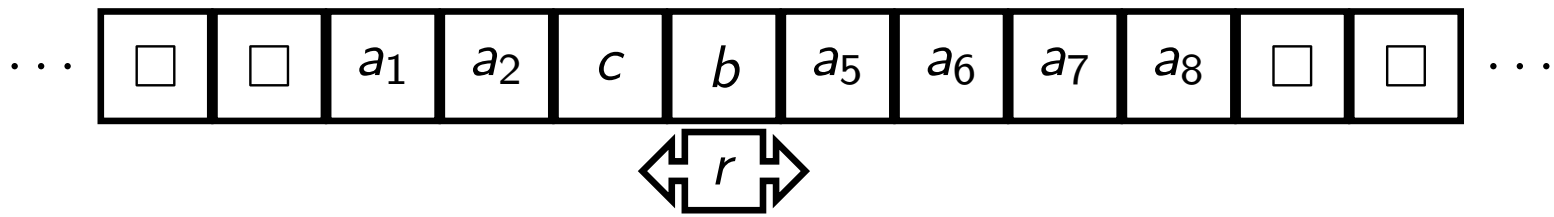


The machine stops when it reaches the accepting state q_Y or the rejecting state q_N.

The language $L(M)$ accepted by the machine M consists of all strings $u \in \{0, 1\}^*$ such that M finally reaches the accepting state q_Y when started in the initial configuration for u.

Intuition of Turing machines

Example: Assume that $\delta(q, a_4) = (p, b, -1)$ and $\delta(p, a_3) = (r, c, 1)$:



The machine stops when it reaches the accepting state q_Y or the rejecting state q_N.

The language $L(M)$ accepted by the machine M consists of all strings $u \in \{0, 1\}^*$ such that M finally reaches the accepting state q_Y when started in the initial configuration for u.

The class **P**

A (deterministic) polynomial time machine (PTM for short) is a Turing machine M for which there is a polynomial $p(n)$ such that:

For every input string $u \in \{0,1\}^n$ the machine M stops after at most $p(n)$ computation steps.

The class **P** (deterministic polynomial time) is the class of all languages $L(M)$ such that M is a deterministic polynomial time machine.

Traditionally **P** is identified with the class of those languages that can be decided in an efficient way.

In the definition of **P** one can replace the Turing machine model by more practical models of computations (e.g. register machines).

The class **P**

Example: An important problem in **P** is the circuit value problem:

- Input: a boolean circuit C.
- Question: Does C evaluate to 1?

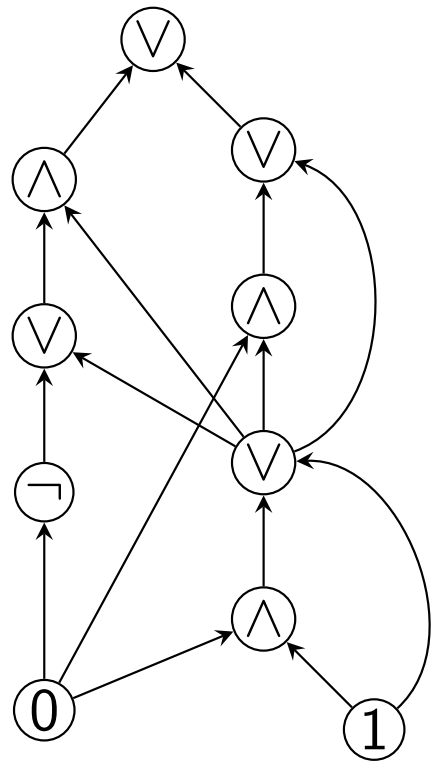
Example of a boolean circuit:

The class **P**

Example: An important problem in **P** is the circuit value problem:

- Input: a boolean circuit C.
- Question: Does C evaluate to 1?

Example of a boolean circuit:

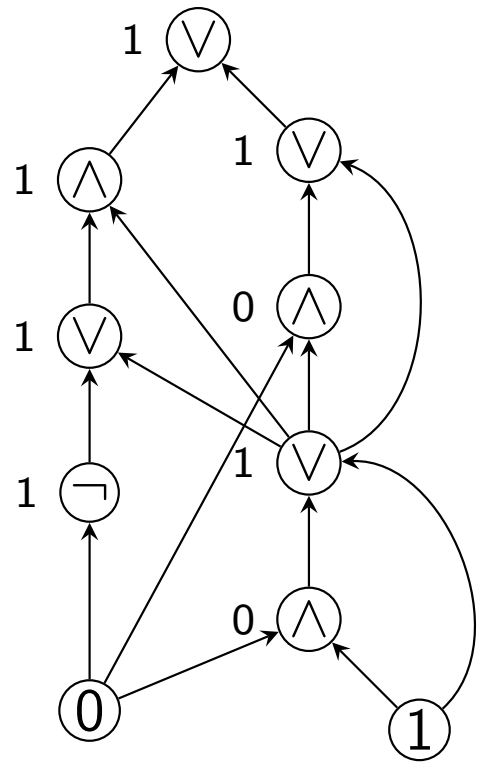


The class **P**

Example: An important problem in **P** is the circuit value problem:

- Input: a boolean circuit C.
- Question: Does C evaluate to 1?

Example of a boolean circuit:



The class **NP**

In the following we need to consider PTMs that take two input strings $u, v \in \{0,1\}^*$.

For this, we assume some encoding $\langle u, v \rangle \in \{0,1\}^*$ of u and v into a single bit string.

One possibility: if $u = a_1 a_2 \cdots a_n$ then $\langle u, v \rangle = a_1 0 a_2 0 \cdots a_{n-1} 0 a_n 1 v$.

The class **NP** consists of all languages L, for which there is a PTM M and a polynomial $r(n)$ such that the following hold for every $u \in \{0,1\}^n$:

- If $u \in L$ then there is $v \in \{0,1\}^{r(n)}$ such that $\langle u, v \rangle \in L(M)$.
- If $u \notin L$ then $\langle u, v \rangle \notin L(M)$ for every string $v \in \{0,1\}^{r(n)}$.

NP stands for non-deterministic polynomial time; it is usually defined by non-deterministic polynomial time Turing machines.

The class **NP**

The string $v \in \{0,1\}^{r(n)}$ with $\langle u, v\rangle \in L(M)$ in case $u \in L$ can be seen as a proof for the fact that $u \in L$.

Intuitively, **NP** contains all languages L for which membership in L is equivalent to the existence of an efficiently verifiable short proof.

Example: Consider boolean formulas such as for instance $F(x_1, x_2, x_3, x_4) = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_4)$.

A boolean formula F is satisfiable if one can set the x_i to truth values (0 or 1) such that the formula evaluates to 1.

SAT is the set of all satisfiable boolean formulas.

SAT $\in$ **NP**: the proof for a satisfiable formula $F(x_1, \ldots, x_n)$ is a bit string $a_1 a_2 \cdots a_n \in \{0,1\}^n$ such that $F(a_1, \ldots, a_n)$ is true.

The class **BPP**

The class **BPP** consists of all languages L for which there is a PTM M and a polynomial $r(n)$ such that the following hold for every $u \in \{0,1\}^n$:

- If $u \in L$ then $\langle u, v \rangle \in L(M)$ for $\geq \frac{2}{3} \cdot 2^{r(n)}$ many $v \in \{0,1\}^{r(n)}$.
- If $u \notin L$ then $\langle u, v \rangle \in L(M)$ for $\leq \frac{1}{3} \cdot 2^{r(n)}$ many $v \in \{0,1\}^{r(n)}$.

BPP stands for bounded-error probabilistic polynomial time.

The string $v \in \{0,1\}^{r(n)}$ is also called the random string.

Intuition: If one randomly sets the bits in $v \in \{0,1\}^{r(n)}$ then with probability $\geq 2/3$ the machine M correctly tells us whether $u \in L$.

Many researchers view **BPP** as the class of those languages that can be decided in an efficient way.

The class **BPP**

Error reduction:

- Run the machines M k times with independently chosen random strings $v_1, \ldots, v_k \in \{0,1\}^{r(n)}$.
- At the end the input u is accepted if $\langle u, v_i \rangle \in L(M)$ for at least $k/2$ many $i \in [1, k]$.
- **Exercise:** Show that the error probability of this new algorithm is $2^{-\Theta(k)}$.

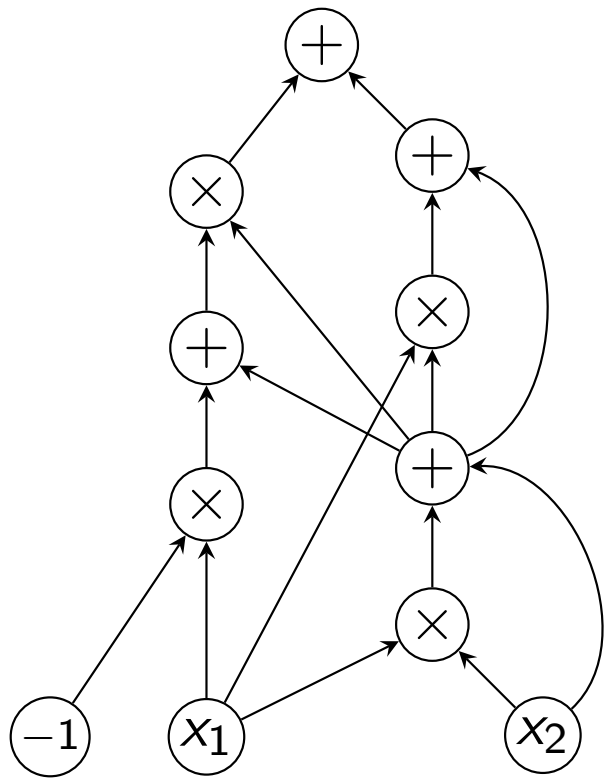
  Hint: use the Chernoff bound.

The class **BPP**

Example: A famous problem in **BPP** that is not known to be **P** is polynomial identity testing (**PIT**):

- Input: an arithmetic circuit C.
- Question: Does C evaluate to the zero polynomial?

An arithmetic circuit:



The class **PSPACE**

A polynomial space machine is a Turing machine M for which there is a polynomial $p(n)$ such that:

For every input string $u \in \{0,1\}^n$ the read-write head never moves more than $p(n)$ cells to the left or right of its initial position.

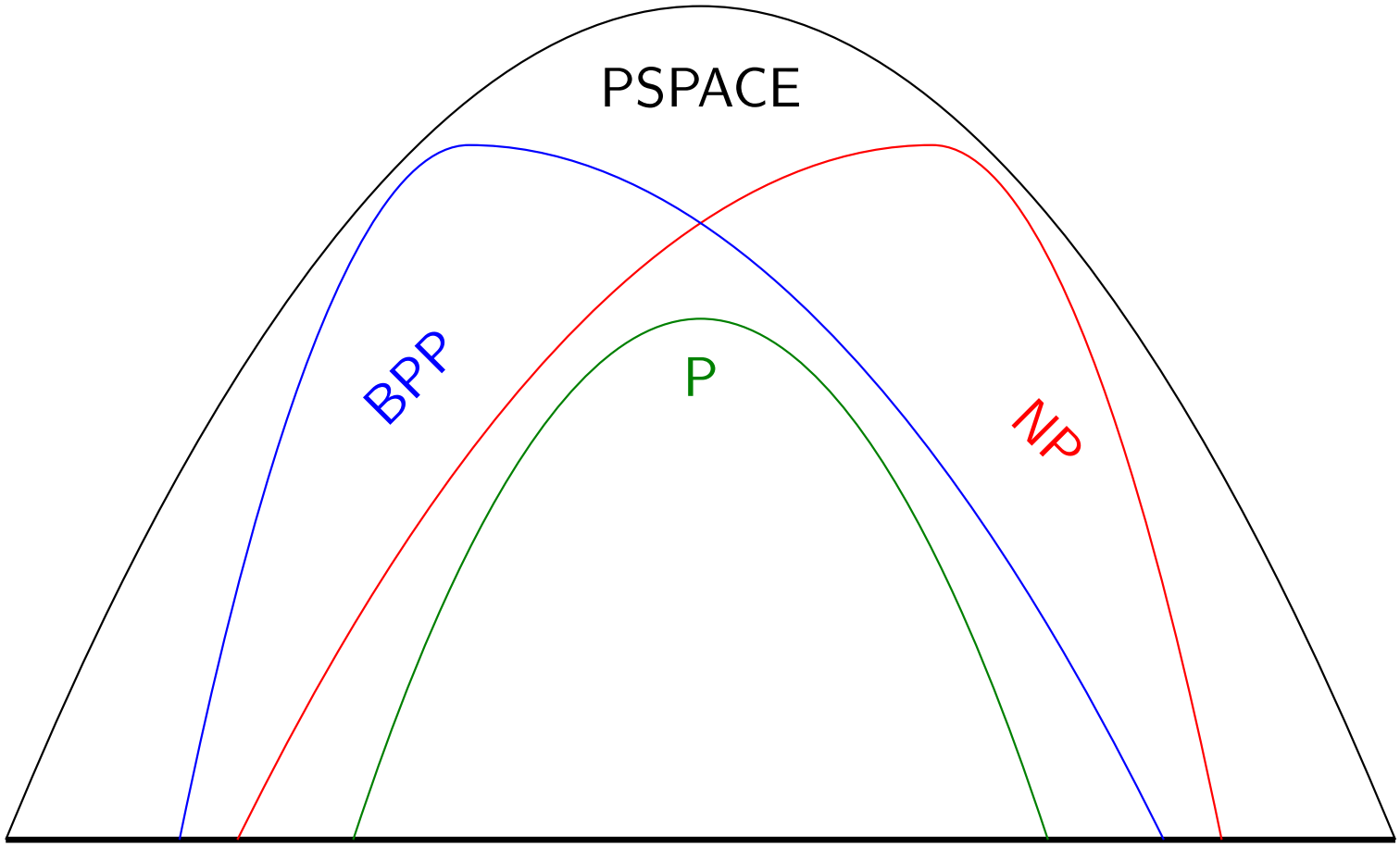
The class **PSPACE** (polynomial space) is the class of all languages $L(M)$ such that M is a polynomial space machine.

Example: QSAT (quantified satisfiability) is the set of all true quantified boolean formulas.

An example of such a formula is $\forall x_1 \exists x_2 : (x_1 \wedge x_2) \vee (\neg x_1 \wedge \neg x_2)$.

QSAT belongs to **PSPACE**.

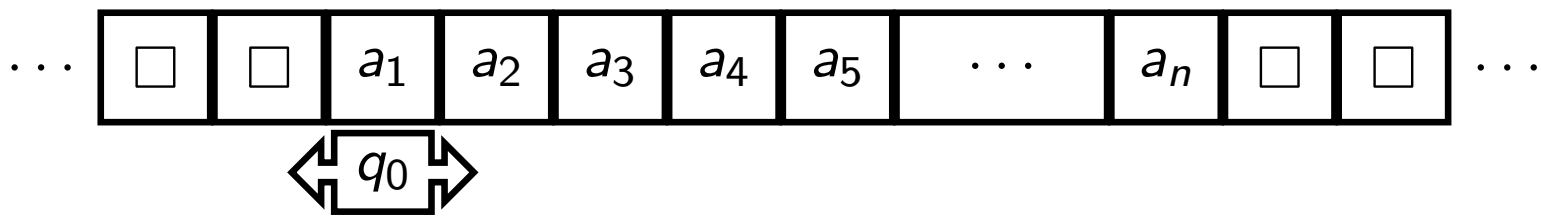
Part of the classical complexity world



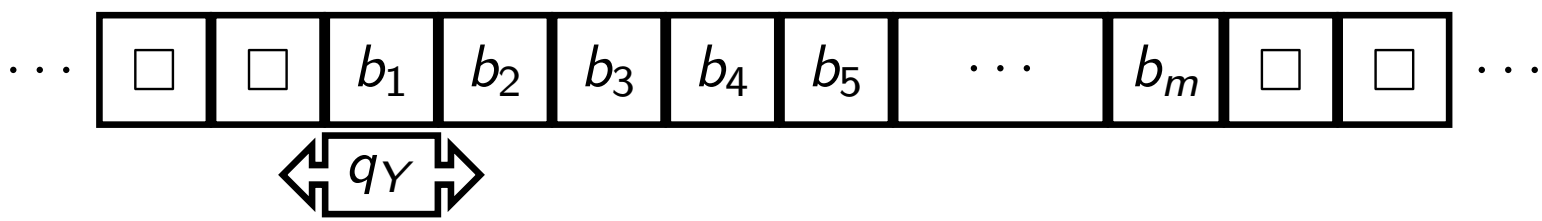
Computing functions with Turing machines

One may also use a Turing machine to M in order to compute an in general partially defined function $f_M : \{0,1\}^* \to \{0,1\}^*$:

$f_M(a_1 a_2 \cdots a_n) = b_1 b_2 \cdots b_m$ iff M reaches from the initial configuration



after a finite number of computations steps the configuration



Karp reductions and completeness

A Karp polynomial time reduction from a language $K \subseteq \{0,1\}^*$ to a language $L \subseteq \{0,1\}^*$ is a totally defined function $f : \{0,1\}^* \to \{0,1\}^*$ such that:

- $f = f_M$ for a PTM M (f can be computed in polynomial time),
- $\forall u \in \{0,1\}^* : u \in K$ if and only if $f(u) \in L$.

We write $K \leq L$ if there is a Karp polynomial time reduction from K to L.

Let **C** be a complexity class (e.g., **NP** or **PSPACE**).

We say that a language $L \subseteq \{0,1\}^*$ is **C**-complete if the following holds:

- $L \in \mathbf{C}$
- $\forall K \in \mathbf{C} : K \leq L$

Karp reductions and completeness

C-complete languages should be seen as the most difficult languages in **C**.

Examples:

- **SAT** is **NP**-complete.
- **QSAT** is **PSPACE**-complete.

Remarks:

- All non-trivial languages in **P** are **P**-complete.

  To get interesting **P**-complete problems, one has to replace Karp polynomial time reductions by Karp logspace reductions.

  Then the circuit value problem is **P**-complete.
- It is open whether **BPP** has complete problems!

Circuits instead of Turing machines

PTMs can be (almost) replaced by boolean circuits.

Let us consider a Boolean circuit $C(x_1, \ldots, x_n)$ with n input gates labelled with the variables $x_1, \ldots, x_n$.

Such a circuit naturally computes a Boolean function $f_C : \{0,1\}^n \to \{0,1\}$.

Example:

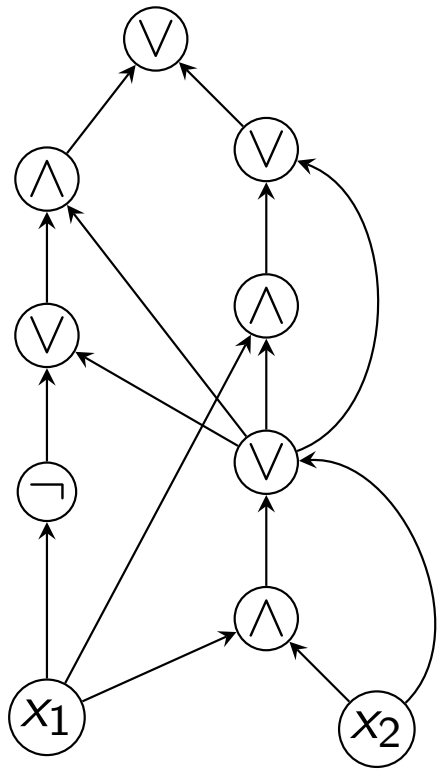
Circuits instead of Turing machines

PTMs can be (almost) replaced by boolean circuits.

Let us consider a Boolean circuit $C(x_1, \ldots, x_n)$ with n input gates labelled with the variables $x_1, \ldots, x_n$.

Such a circuit naturally computes a Boolean function $f_C : \{0,1\}^n \to \{0,1\}$.

Example:



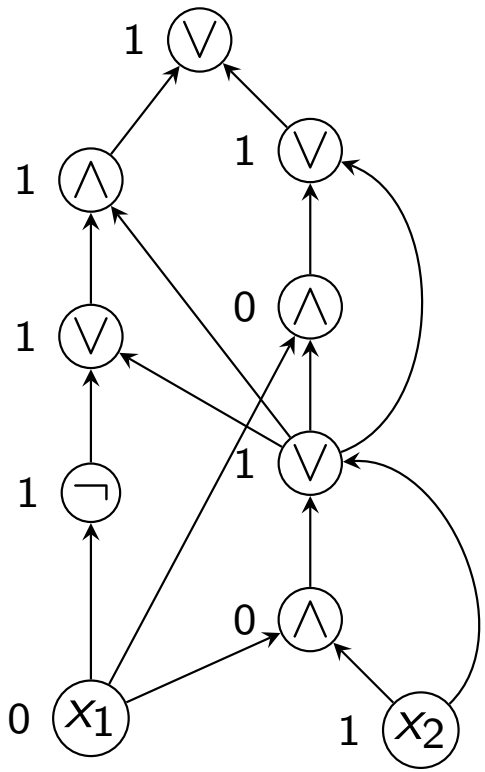
Circuits instead of Turing machines

PTMs can be (almost) replaced by boolean circuits.

Let us consider a Boolean circuit $C(x_1, \ldots, x_n)$ with n input gates labelled with the variables $x_1, \ldots, x_n$.

Such a circuit naturally computes a Boolean function $f_C : \{0,1\}^n \to \{0,1\}$.

Example:



Circuits instead of Turing machines

Theorem 1

For every PTM M there is a PTM N that computes from the input string 1^n (n 1-bits) a description of a Boolean circuit $C_n(x_1, \ldots, x_n)$ such that:

$$\forall u = a_1 a_2 \cdots a_n \in \{0,1\}^n : u \in L(M) \iff f_{C_n}(a_1, a_2, \ldots, a_n) = 1.$$

Intuitively: The machine N builds for a given input length n the hardware needed to simulate the machine M.

Remarks:

- The family of circuits $(C_n)_{n \geq 0}$ from the above theorem is a called a **P**-uniform circuit family.
- The resources needed to compute the function $n \mapsto C_n$ are actually much smaller than polynomial time.

  Logarithmic space (on a deterministic Turing machine) suffices.

Promise problems

It was mentioned that it is not known whether the class **BPP** has complete problems (slide 19).

To get complete problems for **BPP**, we have to consider promise problems.

A promise problem is a pair (L_0, L_1) such that $L_0, L_1 \in \{0,1\}^*$ and $L_0 \cap L_1 = \emptyset$.

The class **promiseBPP** consists of all promise problems (L_0, L_1) for which there is a PTM M and a polynomial $r(n)$ such that the following hold for every $u \in \{0,1\}^n$:

- If $u \in L_1$ then $\langle u, v \rangle \in L(M)$ for at least $\frac{2}{3} \cdot 2^{r(n)}$ many $v \in \{0,1\}^{r(n)}$.
- If $u \in L_0$ then $\langle u, v \rangle \in L(M)$ for at most $\frac{1}{3} \cdot 2^{r(n)}$ many $v \in \{0,1\}^{r(n)}$.

Promise problems

A Karp polynomial time reduction f from the promise problem (K_0, K_1) to the promise problem (L_0, L_1) is a totally defined $f : \{0,1\}^* \to \{0,1\}^*$ such that:

- $f = f_M$ for a PTM M (f can be computed in polynomial time),
- $\forall u \in K_0 : f(u) \in L_0$.
- $\forall u \in K_1 : f(u) \in L_1$.

The following promise problem (L_0, L_1) is then complete for **promiseBPP**:

- L_0 is the set of all binary encodings of boolean circuits $C(x_1, \ldots, x_n)$ with $f_C(a_1, a_2, \ldots, a_n) = 0$ for $\geq \frac{2}{3} \cdot 2^n$ many $a_1 a_2 \cdots a_n \in \{0,1\}^n$.
- L_1 is the set of all binary encodings of boolean circuits $C(x_1, \ldots, x_n)$ with $f_C(a_1, a_2, \ldots, a_n) = 1$ for $\geq \frac{2}{3} \cdot 2^n$ many $a_1 a_2 \cdots a_n \in \{0,1\}^n$.

Complex number

We assume familiarity with the complex numbers $\mathbb{C}$.

Two ways of describing complex numbers:

- $x + iy$ for $x, y \in \mathbb{R}$ (and i satisfying $i^2 = -1$)
- $r \cdot e^{i\varphi}$ for $r \in \mathbb{R}_{\geq 0}$ and $\varphi \in [0, 2\pi)$

If $x + iy = z = r \cdot e^{i\varphi}$ then we have

- $r = \sqrt{x^2 + y^2} =: |z|$ and $\varphi = \arctan(y/x)$
  ($\varphi = \pi/2$ if $x = 0$ and $y > 0$ and $\varphi = 3\pi/2$ if $x = 0$ and $y < 0$)
- $x = r\cos\varphi = \Re(z)$ and $y = r\sin\varphi = \Im(z)$

The complex conjugate of $z = x + iy$ is $z^* = x - iy$.

Note $z \cdot z^* = x^2 + y^2 = |z|^2$.

Complex vector spaces

Quantum computing uses finite dimensional vector spaces over $\mathbb{C}$.

We use Dirac's bra-ket notation:

- Column vectors are denoted with $|x\rangle$.
- Row vectors are denoted with $\langle y|$.

Moreover, for a ket-vector

$$|x\rangle = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_d \end{pmatrix} \in \mathbb{C}^d$$

its conjugated transposed bra-vector is

$$\langle x| = (\alpha_1^*,\ \alpha_2^*,\ \ldots,\ \alpha_d^*).$$

Inner product and norm

The inner product of

$$|x\rangle = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_d \end{pmatrix} \text{ and } |y\rangle = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_d \end{pmatrix}$$

is the complex number $\langle x|y\rangle = \sum_{i=1}^{d} \alpha_i^* \beta_i$.

This is a special case of a matrix product (row times column).

The norm of $|x\rangle$ is $\||x\rangle\| = \sqrt{\langle x|x\rangle} = \sqrt{\sum_{i=1}^{d} |\alpha_i|^2} \in \mathbb{R}_{\geq 0}$.

A unit vector is a vector $|x\rangle$ with $\|x\| = 1$.

For better readability we write $\|x\|$ for $\||x\rangle\|$ in the following.

Inner product and norm

The vector space $\mathbb{C}^d$ together with the inner product $\langle \cdot | \cdot \rangle$ is a so-called (finite dimensional) Hilbert space.

In general, a Hilbert space may have infinite dimension, but we will only consider finite dimensional Hilbert spaces.

Note that $\langle \cdot | \cdot \rangle$ satisfies the following laws, where $\alpha \in \mathbb{C}$:

- $\langle x_1 + x_2 | y \rangle = \langle x_1 | y \rangle + \langle x_2 | y \rangle$ (here, we write $\langle x_1 + x_2 |$ for $\langle x_1 | + \langle x_2 |$)
- $\langle x | y_1 + y_2 \rangle = \langle x | y_1 \rangle + \langle x | y_2 \rangle$
- $\langle \alpha x | y \rangle = \alpha^* \langle x | y \rangle$
- $\langle x | \alpha y \rangle = \alpha \langle x | y \rangle$
- $\langle y | x \rangle = \langle x | y \rangle^*$

Cauchy-Schwarz

Cauchy–Schwarz inequality

For all $|x\rangle, |y\rangle \in \mathbb{C}^d$ we have

$$\langle x|y\rangle \cdot \langle y|x\rangle = |\langle x|y\rangle|^2 \leq \langle x|x\rangle \cdot \langle y|y\rangle$$

with equality if and only if $|x\rangle$ and $|y\rangle$ are linearly dependent (i.e., $\alpha |x\rangle + \beta |y\rangle = 0$ where $\alpha, \beta \in \mathbb{C}$ and $\alpha \neq 0$ or $\beta \neq 0$).

By taking the square root on both sides of the Cauchy–Schwarz inequality, one gets

$$|\langle x|y\rangle| \leq \|x\| \cdot \|y\| .$$

Orthonormal base

A base of $\mathbb{C}^d$ is a set of (ket-)vectors $\{|x_1\rangle, \ldots, |x_d\rangle\}$ such that for every $|x\rangle$ there exist unique $\alpha_1, \ldots, \alpha_d \in \mathbb{C}$ with $|x\rangle = \sum_{i=1}^{d} \alpha_i |x_i\rangle$.

Note: every base of $\mathbb{C}^d$ consists of exactly d non-zero vectors (the dimension of the vector space).

An orthonormal base of $\mathbb{C}^d$ is a base $\{|x_1\rangle, \ldots, |x_d\rangle\}$ such that

$$\langle x_i | x_j \rangle = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases}$$

For every $|y\rangle \in \mathbb{C}^d$ we have $|y\rangle = \sum_{i=1}^{d} \langle x_i | y \rangle \cdot |x_i\rangle$

Exercise: If $|x\rangle$ is a unit vector and $\{|x_1\rangle, \ldots, |x_d\rangle\}$ an orthonormal base then $|x\rangle = \sum_{i=1}^{d} \alpha_i |x_i\rangle$ for unique $\alpha_1, \ldots, \alpha_d \in \mathbb{C}$ with $\sum_{i=1}^{d} |\alpha_i|^2 = 1$.

Orthonormal base

A particular orthonormal base is the standard base consisting of vectors

$$|0\rangle := \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}, \quad |1\rangle := \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \\ 0 \end{pmatrix}, \quad \ldots, \quad |d-1\rangle := \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}.$$

In quantum computing we will work in a vector space $\mathbb{C}^{2^n}$ of dimension 2^n and the standard base $\{|u\rangle : u \in \{0,1\}^n\}$.

Here, the bit strings in $\{0,1\}^n$ are identified with the numbers $0, \ldots, 2^n - 1$, e.g. $0 \mathrel{\widehat{=}} 00,\ 1 \mathrel{\widehat{=}} 01,\ 2 \mathrel{\widehat{=}} 10,\ 3 \mathrel{\widehat{=}} 11$ for $n = 2$.

The base $\{|u\rangle : u \in \{0,1\}^n\}$ is also called the computational base; its elements can be identified with the possible values of an n-bit register.

Linear mappings

A mapping $f : \mathbb{C}^d \to \mathbb{C}^d$ is linear if for all $|x\rangle, |y\rangle \in \mathbb{C}^d$ and all $\alpha \in \mathbb{C}$:

- $f(|x\rangle + |y\rangle) = f|x\rangle + f|y\rangle$ (we write $f|x\rangle$ for $f(|x\rangle)$)
- $f(\alpha\,|x\rangle) = \alpha f|x\rangle$

After fixing a base $\{|x_1\rangle, \ldots, |x_d\rangle\}$, one can identify the linear mapping f with the $(d \times d)$-matrix $A = (A_{i,j})_{1 \le i,j \le d}$, where

$$f|x_j\rangle = \sum_{i=1}^{d} A_{i,j}\,|x_i\rangle\,.$$

We then have: if $|x\rangle = \sum_{i=1}^{d} \alpha_i\,|x_i\rangle$ and $f|x\rangle = \sum_{i=1}^{d} \beta_i\,|x_i\rangle$ then

$$A \cdot \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_d \end{pmatrix} = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_d \end{pmatrix}.$$

Note: If $\{|x_1\rangle, \ldots, |x_d\rangle\}$ is orthonormal then $A_{i,j} = \langle x_i|\, f\,|x_j\rangle$.

Changing the basis

Let f be linear mapping and let $\{|x_1\rangle, \ldots, |x_d\rangle\}$ and $\{|y_1\rangle, \ldots, |y_d\rangle\}$ be two bases of $\mathbb{C}^d$.

Let A (resp., B) be the matrix for f in the basis $\{|x_1\rangle, \ldots, |x_d\rangle\}$ (resp., $\{|y_1\rangle, \ldots, |y_d\rangle\}$).

Then there is an invertible matrix $C \in \mathbb{C}^{d\times d}$ such that $B = C^{-1}AC$.

Exercise: Find the matrix C explicitly.

Two matrices $A, B \in \mathbb{C}^{d\times d}$ are similar if there is an invertible matrix $C \in \mathbb{C}^{d\times d}$ such that $B = C^{-1}AC$.

Operations for matrices

Composition of linear mappings corresponds to matrix multiplication:

$$(AB)_{i,j} = \sum_{k=1}^{d} A_{i,k} B_{k,j}.$$

Recall: $AB \neq BA$ in general!

Transposed matrix: $(A^{\mathsf{T}})_{i,j} = A_{j,i}$

Conjugated matrix: $(A^*)_{i,j} = A^*_{i,j}$

Adjoint matrix: $A^\dagger = (A^*)^{\mathsf{T}} = (A^{\mathsf{T}})^*$

We have: $(A^\dagger)^\dagger = A$, $(A+B)^\dagger = A^\dagger + B^\dagger$, $(AB)^\dagger = B^\dagger A^\dagger$ and $(A^{-1})^\dagger = (A^\dagger)^{-1}$ for A invertible.

The operators $\mathsf{T}, *, \dagger$ can be defined also for rectangular matrices, in particular for bra- or ket-vectors. Note that $|x\rangle^\dagger = \langle x|$.

The outer product of vectors

For a ket-vector $|x\rangle$ and a bra-vector $\langle y|$ (both of dimension d) we can form their outer product $|x\rangle\langle y| \in \mathbb{C}^{d\times d}$.

It is a special case of a rectangular matrix product.

More specifically: if

$$|x\rangle = \begin{pmatrix} a_1 \\ \vdots \\ a_d \end{pmatrix} \quad \text{and} \quad \langle y| = (b_1, \ldots, b_d)$$

then $|x\rangle\langle y| = (a_i b_j)_{1\leq i,j\leq d}$.

Note that for every matrix $A = (A_{i,j})_{0\leq i,j\leq d-1}$ we have $A = \sum_{i,j} A_{i,j} |i\rangle\langle j|$ for the standard base $|0\rangle, \ldots, |d-1\rangle$.

Trace of a matrix

Trace of a matrix: $\mathrm{tr}(A) = \sum_{i=1}^{d} A_{i,i}$ (sum of the diagonal entries)

Important facts of the trace:

- $\mathrm{tr}(A+B) = \mathrm{tr}(A) + \mathrm{tr}(B)$
- $\mathrm{tr}(\alpha A) = \alpha \cdot \mathrm{tr}(A)$
- $\mathrm{tr}(AB) = \mathrm{tr}(BA)$

Exercise: Prove this.

Warning: $\mathrm{tr}(AB) \neq \mathrm{tr}(A)\mathrm{tr}(B)$ in general!

$\mathrm{tr}(AB) = \mathrm{tr}(BA)$ implies $\mathrm{tr}(C^{-1}AC) = \mathrm{tr}(A)$ for an invertible matrix C.

Consequence: If A and B are similar then $\mathrm{tr}(A) = \mathrm{tr}(B)$.

Hence, the trace of a matrix is invariant under a basis change.

Special matrices

A matrix A is

- normal if $AA^\dagger = A^\dagger A$,
- unitary if $A^\dagger = A^{-1}$.
- Hermitian (or self-adjoint) if $A = A^\dagger$,
- positive semi-definite if A is Hermitian and $\langle x| A |x\rangle \geq 0$ for every $|x\rangle$.
- positive definite if A is Hermitian and $\langle x| A |x\rangle > 0$ for every $|x\rangle$.
- a projector if A is Hermitian and $A^2 = A$,

Note that Hermitian and unitary matrices are also normal.

If A and B describe the same linear function in two different orthonormal bases then there is a unitary matrix U with $B = U^{-1}AU = U^\dagger AU$.

Projectors onto subspaces

Let $S \leq \mathbb{C}^d$ be a subspace of $\mathbb{C}^d$ and let $\{|x_1\rangle, \ldots, |x_k\rangle\}$ be an orthonormal basis of S. The projector onto the subspace S is

$$\Pi_S = \sum_{i=1}^{k} |x_i\rangle\langle x_i| .$$

We have $\Pi_S^\dagger = \Pi_S$ and

$$\begin{aligned}
\Pi_S^2 &= \left(\sum_{i=1}^{k} |x_i\rangle\langle x_i| \right) \left(\sum_{j=1}^{k} |x_j\rangle\langle x_j| \right) \\
&= \sum_{i,j} |x_i\rangle\langle x_i|\, |x_j\rangle\langle x_j| = \sum_{i,j} |x_i\rangle\langle x_i|x_j\rangle\langle x_j| \\
&= \sum_{i=1}^{k} |x_i\rangle\langle x_i| = \Pi_S .
\end{aligned}$$

Projectors onto subspaces

Moreover, for every $|x\rangle \in \mathbb{C}^d$ we have

$$\Pi_S |x\rangle = \sum_{i=1}^{k} |x_i\rangle \langle x_i|x\rangle = \sum_{i=1}^{k} \langle x_i|x\rangle |x_i\rangle \in S$$

If $|x\rangle \in S$ then we can write $|x\rangle = \sum_{j=1}^{k} \alpha_j |x_j\rangle$ and we get

$$\begin{aligned} \Pi_S |x\rangle &= \left(\sum_{i=1}^{k} |x_i\rangle \langle x_i| \right) \left(\sum_{j=1}^{k} \alpha_j |x_j\rangle \right) \\ &= \sum_{i,j} \alpha_j |x_i\rangle \langle x_i|x_j\rangle = \sum_{j=1}^{k} \alpha_j |x_j\rangle = |x\rangle . \end{aligned}$$

Exercise: If Π is any projector ($\Pi^2 = \Pi$) find a subspace S with $\Pi = \Pi_S$.

The Pauli matrices

The matrices

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

are the famous Pauli matrices.

They are unitary as well as Hermitian.

In particular, they satisfy: $X^2 = Y^2 = Z^2 = \mathrm{Id}_2$.

Eigenvectors and eigenvalues

For a $(d \times d)$-matrix A, a vector $|x\rangle \in \mathbb{C}^d$ is called an eigenvector of A if there is $\lambda \in \mathbb{C}$ with $A|x\rangle = \lambda |x\rangle$.

λ is the eigenvalue of A for $|x\rangle$ and $|x\rangle$ is an eigenvector for λ.

Let λ be an eigenvalue of A. Then the set of all eigenvectors for λ form a subspace of $\mathbb{C}^d$: if $A|x\rangle = \lambda |x\rangle$ and $A|y\rangle = \lambda |y\rangle$ then:

$$A(|x\rangle + |y\rangle) = \lambda(|x\rangle + |y\rangle) \quad \text{and} \quad A(\alpha |x\rangle) = \lambda\alpha |x\rangle$$

This subspace is called the eigenspace of λ and its dimension is the geometric multiplicity of λ.

The eigenspaces for the different eigenvalues of A are linearly independent.

Similar matrices have the same eigenvalues with the same geometric multiplicities (but the eigenvectors change).

The spectral theorem

Theorem 2 (spectral theorem, see e.g. Nielsen Chuang, page 72)

Let $A \in \mathbb{C}^{d\times d}$ be a matrix. Then there is an orthonormal basis of $\mathbb{C}^d$ consisting of eigenvectors of A if and only if A is normal.

Let A be a normal matrix and $\{|x_1\rangle, \ldots |x_d\rangle\}$ an orthonormal basis of eigenvectors of A. Let λ_i be the eigenvalue for $|x_i\rangle$. Then we have

$$A = \sum_{i=1}^{d} \lambda_i |x_i\rangle\langle x_i| .$$

In the orthonormal basis $\{|x_1\rangle, \ldots |x_d\rangle\}$, A becomes the diagonal matrix

$$\begin{pmatrix} \lambda_1 & 0 & \ldots & 0 \\ 0 & \lambda_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \lambda_d \end{pmatrix}$$

The spectral theorem

The diagonal entries are the eigenvalues of A.

Every eigenvalue appears its geometric multiplicity many times.

The spectral theorem says that the following two conditions are equivalent:

- A is normal.
- There is a unitary matrix U such that $U^\dagger AU$ is diagonal (A is unitarily diagonizable).

For a general (not necessarily normal) matrix $A \in \mathbb{C}^{d\times d}$ the following are equivalent:

- A is similar to a diagonal matrix.
- $\mathbb{C}^d$ has a basis (not necessarily orthonormal) consisting of eigenvectors of A.
- The sum of the geometric multiplicities of the eigenvalues of A is d.

Eigenvalues of special matrices

Theorem 3

The following are equivalent for a matrix U:

- U is unitary.
- The set of columns of U form an orthonormal basis.
- All eigenvalues λ of U satisfy $|\lambda| = 1$ (i.e. $\lambda = e^{i\phi}$ for some ϕ).
- U preserves the inner product. In formulas:
  $\langle x| U^\dagger U |y\rangle = \langle x|y\rangle$ (note that $(U|y\rangle)^\dagger = |y\rangle^\dagger U^\dagger = \langle y| U^\dagger$).

Theorem 4

The following are equivalent for a matrix H:

- H is Hermitian.
- All eigenvalues of H are real numbers.

Eigenvectors and eigenvalues

Theorem 5

The following are equivalent for a matrix P:

- P is positive definite (positive semi-definite).
- All eigenvalues of P are real and > 0 (≥ 0).

Exercise: For every matrix $A \in \mathbb{C}^{d \times d}$ the matrix $A^\dagger A$ is positive semi-definite.

Theorem 6

The following are equivalent for a matrix Π:

- Π is a projector
- All eigenvalues λ of Π are 0 or 1.

Tensor product of vector spaces

Theorem 7

Let U and V be vector spaces. Then there is up to isomorphism of vector spaces a unique vector space $U \otimes V$ with the following properties:

- There is a bilinear mapping $f : U \times V \to U \otimes V$.
- For every bilinear mapping $g : U \times V \to W$, where W is a vector space, there is a unique linear mapping $h : U \otimes V \to W$ such that $g(|x\rangle, |y\rangle) = h(f(|x\rangle, |y\rangle))$ for all $|x\rangle \in U$ and $|y\rangle \in V$.

The vector space $U \otimes V$ is the tensor product of U and V.

In the following we write $|x\rangle \otimes |y\rangle$ for $f(|x\rangle, |y\rangle)$.

Assume moreover in the following that $U \cong \mathbb{C}^d$ and $U \cong \mathbb{C}^e$ are finite-dimensional (the only interesting case).

How can we construct (a vector space isomorphic to) $U \otimes V$?

Tensor product of vector spaces

Fix standard bases

- $\{|0\rangle, \ldots, |d-1\rangle\}$ for U and
- $\{|0\rangle, \ldots, |e-1\rangle\}$ for V.

Then $\mathbb{C}^d \otimes \mathbb{C}^e$ is the vector space with the standard

- $\{|i\,j\rangle : 0 \leq i \leq d-1, 0 \leq i \leq e-1\}$

(in particular, $\mathbb{C}^d \otimes \mathbb{C}^e \cong \mathbb{C}^{de}$) and the corresponding mapping $f : \mathbb{C}^d \times \mathbb{C}^e \to \mathbb{C}^d \otimes \mathbb{C}^e$ is uniquely defined by

$$f(|i\rangle, |j\rangle) = |i\rangle \otimes |j\rangle = |i\,j\rangle .$$

Bilinearity of $\otimes$ then implies that for $|x\rangle = \sum_{i=0}^{d-1} \alpha_i |i\rangle$ and $|y\rangle = \sum_{j=0}^{e-1} \alpha_j |j\rangle$ we have

$$|x\rangle \otimes |y\rangle = \sum_{i,j} \alpha_i \beta_j |i\,j\rangle . \tag{1}$$

Tensor product of vector spaces

Remark: $\otimes$ is not surjective (this will be shown on Slide 57).

If $g : U \to U$ and $h : V \to V$ are linear mappings then we can define a linear mapping

$$g \otimes h : U \otimes V \to U \otimes V$$

as follows for all $0 \leq i \leq d-1$, $0 \leq i \leq e-1$:

$$(g \otimes h)(|i\,j\rangle) = g(|i\rangle) \otimes h(|j\rangle)$$

If A (B) is the matrix for g (h) in the standard basis of $\mathbb{C}^d$ ($\mathbb{C}^e$), then the matrix for $g \otimes h$ in the standard basis of $\mathbb{C}^{de}$ is the Kronecker product of A and B, often also called the tensor product of A and B.

We define the Kronecker product for rectangular matrices on the next slide.

Tensor product of matrices

Let $A = (a_{i,j})_{1 \leq i \leq k, 1 \leq j \leq \ell}$ be a $(k \times \ell)$-matrix and $B = (b_{i,j})_{1 \leq i \leq m, 1 \leq j \leq n}$ be an $(m \times n)$-matrix.

Their Kronecker product (or tensor product) $A \otimes B$ is the following $(km \times \ell n)$-matrix:

$$A \otimes B = \begin{pmatrix} a_{1,1}B & a_{1,2}B & \dots & a_{1,\ell}B \\ a_{2,1}B & a_{2,2}B & \dots & a_{2,\ell}B \\ \vdots & \vdots & \vdots & \vdots \\ a_{k,1}B & a_{k,2}B & \dots & a_{k,\ell}B \end{pmatrix}$$

Note: If $|x\rangle \in \mathbb{C}^d$ and $|y\rangle \in \mathbb{C}^e$ then $|x\rangle \otimes |y\rangle \in \mathbb{C}^{de}$ is as in (1).

If $|i\rangle$ and $|j\rangle$ are vectors from the standard bases ($0 \leq i \leq k-1$ and $0 \leq j \leq \ell - 1$) then we also write $|i\,j\rangle$ for the tensor product $|i\rangle \otimes |j\rangle$.

Tensor product of matrices

The tensor product satisfies the following laws:

- $(A + B) \otimes C = A \otimes C + B \otimes C$ and $A \otimes (B + C) = A \otimes B + A \otimes C$
- $(\alpha A) \otimes B = A \otimes (\alpha B) = \alpha(A \otimes B)$.
- $(A \otimes B) \cdot (C \otimes D) = (A \cdot C) \otimes (B \cdot D)$, where
  A is $(k \times \ell)$, B is $(m \times n)$, C is $(\ell \times p)$, D is $(n \times q)$.
- $(A \otimes B)^\dagger = A^\dagger \otimes B^\dagger$.

A special case of the 3rd law is the following, where $|x\rangle, |x'\rangle \in \mathbb{C}^k$, $|y\rangle, |y'\rangle \in \mathbb{C}^\ell$.

$$(\langle x| \otimes \langle y|)(|x'\rangle \otimes |y'\rangle) = \langle x|x'\rangle \langle y|y'\rangle$$

In particular $|x\rangle = |x'\rangle$ and $|y\rangle = |y'\rangle$ yields $\||x\rangle \otimes |y\rangle\| = \|x\| \cdot \|y\|$.

Tensor product of matrices

Note: in general we have $A \otimes B \neq B \otimes A$

Exercise: Show the following:

$$A \text{ and } B \text{ are } \left\{ \begin{array}{l} \text{unitary} \\ \text{Hermitian} \\ \text{positive definite} \\ \text{projectors} \end{array} \right\} \implies A \otimes B \text{ is } \left\{ \begin{array}{l} \text{unitary} \\ \text{Hermitian} \\ \text{positive definite} \\ \text{a projector} \end{array} \right\}$$

For a rectangular matrix A and $k \geq 1$ we define

$$A^{\otimes k} = \underbrace{A \otimes \cdots \otimes A}_{k \text{ many}}.$$

Principles of quantum computing: the state

Classical computing: at each time instant the system is in one of say d states $0, 1, \ldots d-1$.

Quantum computing: a (quantum) state is a unit vector $|x\rangle \in \mathbb{C}^d$:

$$|x\rangle = \sum_{i=0}^{d-1} \alpha_i |i\rangle ,$$

where $\alpha_i \in \mathbb{C}$ (the amplitude of $|i\rangle$) and $\sum_{i=0}^{d-1} |\alpha_i|^2 = 1$.

Later, when we introduce quantum circuits, the standard basis

$$\{|0\rangle , \ldots , |d-1\rangle\}$$

will be replaced by the computational basis

$$\{|u\rangle : u \in \{0,1\}^n\}$$

of an n-qubit quantum register.

Principles of quantum computing: the dynamics

Classical computing: the system evolves according to some (possibly time-dependant) function $f : \{0, 1, \ldots, d-1\} \to \{0, 1, \ldots, d-1\}$ on the set of states.

Quantum computing: systems evolve according to unitary transformations.

If at time t the system is in state $|x\rangle \in \mathbb{C}^d$ (a unit vector) then at time $t+1$ the system is in state $U(t)\,|x\rangle$ for some unitary matrix $U(t)$ (that may depend on time).

Note: $U(t)\,|x\rangle$ is again a unit vector since unitary matrices preserve the norm.

Principles of quantum computing: measurements

Classical computing: In principle, the observer of a classical system knows at each time instant the current state $s \in \{0, 1, \ldots, d-1\}$.

Quantum computing: The current state $|x\rangle \in \mathbb{C}^d$ is hidden for an observer (in particular, one cannot determine the amplitudes).

All she/he can do is a measurement.

The simplest possible measurement in our setting would be a full projective measurement in the standard base:

After measuring the current state $|x\rangle = \sum_{i=0}^{d-1} \alpha_i |i\rangle$ the system collapses to the basis states $|i\rangle$ (the new state of the system) with probability $|\alpha_i|^2$.

Recall: $\sum_{i=0}^{d-1} |\alpha_i|^2 = 1$.

The observer gets the knowledge to which basis state $|i\rangle$ the quantum state collapses.

Principles of quantum computing: measurements

More generally, a projective measurement is given by a collection of projectors $\{\Pi_1, \ldots, \Pi_k\}$ ($\Pi_i^2 = \Pi_i$ and $\Pi_i^\dagger = \Pi_i$) such that

- $\Pi_i \Pi_j = 0$ (the zero matrix) for $i \neq j$ and
- $\sum_{i=1}^k \Pi_i = \mathrm{Id}_d$ (the $(d \times d)$ identity matrix).

Applying this projective measurement to the state $|x\rangle$ results with probability $\|\Pi_i |x\rangle\|^2 = \langle x| \Pi_i |x\rangle$ in the post-measurement state

$$\frac{\Pi_i |x\rangle}{\|\Pi_i |x\rangle\|}.$$

This is a unit vector from the subspace $S_i = \{\Pi_i |x\rangle : |x\rangle \in \mathbb{C}^d\}$ onto which Π_i projects. The observer gets the knowledge of i.

The probabilities $\langle x| \Pi_i |x\rangle$ sum to 1:

$$\sum_{i=1}^k \langle x| \Pi_i |x\rangle = \langle x| \left(\sum_{i=1}^k \Pi_i \right) |x\rangle = \langle x|x\rangle = 1.$$

Principles of quantum computing: measurements

Remarks:

- If $i \neq j$ and $|x\rangle, |y\rangle \in \mathbb{C}^d$ then $\langle x| \Pi_i^\dagger \Pi_j |y\rangle = \langle x| \Pi_i \Pi_j |y\rangle = 0$. Thus, S_i and S_j are orthogonal for $i \neq j$.
- $\mathbb{C}^d = S_1 \oplus S_2 \oplus \cdots \oplus S_k$, since $|x\rangle = \sum_{i=1}^k \Pi_i |x\rangle$ for all $|x\rangle \in \mathbb{C}^d$.
- In a full projective measurement in the standard base, the S_i are the vector spaces spanned by the basis vectors $|i\rangle$ (and we have $k = d$).
- A Hermitian matrix H yields a projective measurement as follows:

  Let $\lambda_1, \ldots, \lambda_k \in \mathbb{R}$ be the different eigenvalues of H and let S_i be the eigenspace of λ_i.

  Then $\{\Pi_{S_1}, \ldots, \Pi_{S_k}\}$ defines a projective measurement!
- There is an even more general notion of measurement in quantum information theory: POVM – positive operator-valued measurement.

  We won't need it in this lecture.

Principles of quantum computing: combining systems

Classical computing: If we have two systems with state spaces $S = \{x_1, \ldots, x_d\}$ and $T = \{y_1, \ldots, y_e\}$, then the two systems can be viewed as a single system with state space $S \times T$.

Quantum computing: Two quantum systems with standard bases

- $\{|0\rangle, \ldots, |d-1\rangle\}$ (yielding space $\mathbb{C}^d$) and
- $\{|0\rangle, \ldots, |e-1\rangle\}$ (yielding space $\mathbb{C}^e$)

can be combined into a single system with standard base

- $\{|i\,j\rangle : 0 \leq i < d, 0 \leq j < e\}$ yielding space $\mathbb{C}^d \otimes \mathbb{C}^e \cong \mathbb{C}^{de}$.

If the two systems are in states $|x\rangle = \sum_{i=0}^{d-1} \alpha_i |i\rangle$ and $|y\rangle = \sum_{j=0}^{e-1} \beta_j |j\rangle$ then the combined system is in state

$$|x\rangle \otimes |y\rangle = \sum_{i,j} \alpha_i \beta_j |i\rangle \otimes |j\rangle = \sum_{i,j} \alpha_i \beta_j |i\,j\rangle .$$

Principles of quantum computing: combining systems

Important remark: Not every state of $\mathbb{C}^d \otimes \mathbb{C}^e$ can be written as $|x\rangle \otimes |y\rangle$ for states $|x\rangle \in \mathbb{C}^d$ and $|y\rangle \in \mathbb{C}^e$.

A state $|z\rangle \in \mathbb{C}^d \otimes \mathbb{C}^e$ is called entangled if it is not of the form $|x\rangle \otimes |y\rangle$ for states $|x\rangle \in \mathbb{C}^d$ and $|y\rangle \in \mathbb{C}^e$.

Example: The Bell state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ is entangled.
To see this, assume that

$$\begin{aligned} \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) &= (\alpha\,|0\rangle + \beta\,|1\rangle) \otimes (\gamma\,|0\rangle + \delta\,|1\rangle) \\ &= \alpha\gamma\,|00\rangle + \alpha\delta\,|01\rangle + \beta\gamma\,|10\rangle + \beta\delta\,|11\rangle\,. \end{aligned}$$

This implies $\alpha\gamma = \beta\delta = \frac{1}{\sqrt{2}}$ and $\alpha\delta = \beta\gamma = 0$, from which we get $\alpha\beta\gamma\delta = \frac{1}{2}$ and $\alpha\beta\gamma\delta = 0$, a contradiction.

Principles of quantum computing: combining systems

If two systems evolve according to the unitary transformations U and V then the combined system evolves according to the unitary $U \otimes V$.

This makes sense, since $(U \otimes V)(|x\rangle \otimes |y\rangle) = U|x\rangle \otimes V|y\rangle$.

Measuring the

- 1st system using $\{\Pi_1, \ldots, \Pi_k\}$ and independently the
- 2nd system using $\{\Phi_1, \ldots, \Phi_\ell\}$

is the same as measuring the combined system using

- $\{\Pi_i \otimes \Phi_j : 1 \leq i \leq k, 1 \leq j \leq \ell\}$.

This yields the intuitively correct probabilities!

Principles of quantum computing: combining systems

Assume that $|x\rangle$ and $|y\rangle$ are the states of the two systems and measure them independently from each other.

With $\pi_i = \|\Pi_i |x\rangle\|$ and $\rho_j = \|\Phi_j |y\rangle\|$ we get

- Prob$\big[$ post measurement state of system 1 $= \frac{\Pi_i|x\rangle}{\pi_i}\big] = \pi_i^2$
- Prob$\big[$ post measurement state of system 2 $= \frac{\Phi_j|y\rangle}{\rho_j}\big] = \rho_j^2$

Hence, with probability $\pi_i^2\rho_j^2$ the combined system is in state

$$\frac{\Pi_i |x\rangle}{\pi_i} \otimes \frac{\Phi_j |y\rangle}{\rho_j} = \frac{\Pi_i |x\rangle \otimes \Phi_j |y\rangle}{\pi_i\rho_j} = \frac{\Pi_i |x\rangle \otimes \Phi_j |y\rangle}{\|\Pi_i |x\rangle \otimes \Phi_j |y\rangle\|}$$

after the measurements.

With the combined measurement $\{\Pi_i \otimes \Phi_j : 1 \leq i \leq k, 1 \leq j \leq \ell\}$ the same result is obtained!

Global phase and the density matrix

Consider quantum states $|x\rangle, |y\rangle \in \mathbb{C}^d$ (so $\langle x|x\rangle = \langle y|y\rangle = 1$) and assume there is $\alpha \in \mathbb{C}$ such that $|x\rangle = \alpha |y\rangle$.

We must have $|\alpha| = 1$, i.e., $\alpha = e^{i\phi}$ for some $\phi \in [0, 2\pi)$.

α is called a global phase factor.

It has no physical meaning in the following sense: For every measurement $\{\Pi_1, \ldots, \Pi_k\}$ and every outcome j we have $\langle x| \Pi_j |x\rangle = \langle y| \Pi_j |y\rangle$.

Hence, $|x\rangle$ and $|y\rangle$ cannot be distinguished by measurements.

Exercise: Show that: $|x\rangle = e^{i\phi} |y\rangle$ for some $\phi \iff |x\rangle\langle x| = |y\rangle\langle y|$.

The matrix $|x\rangle\langle x|$ is called the density matrix of the quantum state $|x\rangle$.

Mixed quantum states

Let $|x\rangle \in \mathbb{C}^d$ be a quantum state and $\{\Pi_1, \ldots, \Pi_k\}$ a measurement.

How should we describe the state resulting from the measurement (before actually doing the measurement and the resulting collapse)?

It is the probability distribution on the states

$$|x_1\rangle := \frac{\Pi_1 |x\rangle}{\|\Pi_1 |x\rangle\|}, \ldots, |x_k\rangle := \frac{\Pi_k |x\rangle}{\|\Pi_k |x\rangle\|},$$

where $|x_i\rangle$ arises with probability $p_i := \|\Pi_i |x\rangle\|^2 = \langle x| \Pi_i |x\rangle$.

A set $\{(p_1, |x_1\rangle), \ldots, (p_k, |x_k\rangle)\}$ is an ensemble of pure quantum states or a mixed state if:

- $(p_1, \ldots, p_k)$ is a probability distribution, i.e., $p_i \in \mathbb{R}_{\geq 0}$, $\sum_{i=1}^k p_i = 1$,
- $|x_1\rangle, \ldots, |x_k\rangle$ are quantum states.

Mixed quantum states

The quantum states we considered so far are also called pure states.

Unitary evolution and measurements can be easily defined for mixed states.

Let $\mathcal{E} = \{(p_1, |x_1\rangle), \ldots, (p_k, |x_k\rangle)\}$ be a mixed state.

- The unitary transformation U transforms $\mathcal{E}$ into the mixed state $\{(p_1, U|x_1\rangle), \ldots, (p_k, U|x_k\rangle)\}$.
- The measurement $\{\Pi_1, \ldots, \Pi_\ell\}$ transforms $\mathcal{E}$ into the mixed state

$$\left\{ \left(p_j \left\| \Pi_i |x_j\rangle \right\|^2, \frac{\Pi_i |x_j\rangle}{\left\| \Pi_i |x_j\rangle \right\|} \right) : 1 \leq i \leq \ell, 1 \leq j \leq k \right\}.$$

The probability for outcome $i \in [1, \ell]$ is $\sum_{j=1}^{k} p_j \left\| \Pi_i |x_j\rangle \right\|^2$.

Density matrices

As for pure states, two mixed states can be physically indistinguishable in the sense that for every measurement they yields the same probability for every outcome.

Example: $\{(0.7, |x\rangle), (0.3, |x\rangle)\}$ and $\{(0.5, |x\rangle), (0.5, e^{i\phi} |x\rangle)\}$ are physically indistinguishable from $\{(1, |x\rangle)\}$ (a pure state).

The mixed state $\{(p_1, |x_1\rangle), \ldots, (p_k, |x_k\rangle)\}$ can be described by its density matrix

$$\rho = \sum_{i=1}^{k} p_i |x_i\rangle\langle x_i| . \tag{2}$$

It turns out that mixed states with the same density matrix cannot be distinguished by measurements.

Density matrices

Lemma 8

A matrix ρ is a density matrix (i.e., ρ has the form (2) for a probability distribution $(p_1, \ldots, p_k)$ and pure states $|x_1\rangle, \ldots, |x_k\rangle$) if and only if

- ρ is positive semi-definite and
- $\mathrm{Tr}(\rho) = 1$.

Let ρ be a density matrix.

- A unitary U transforms the density matrix ρ into $U\rho U^\dagger$.
- The measurement $\{\Pi_1, \ldots, \Pi_k\}$ transforms ρ into $\sum_{i=1}^{k} \Pi_i \rho \Pi_i$.

  Outcome $i \in [1, k]$ occurs with probability $\mathrm{Tr}(\Pi_i \rho)$ and leads to the collapsed density matrix $\frac{\Pi_i \rho \Pi_i}{\mathrm{Tr}(\Pi_i \rho)}$ (note that $\mathrm{Tr}(\Pi_i \rho) = \mathrm{Tr}(\Pi_i \rho \Pi_i)$).

Quantum bits (qubits)

Unit vectors of the form $\alpha\,|0\rangle + \beta\,|1\rangle \in \mathbb{C}^2$ are the states of a single qubit.

Physically, one could realize a qubit by a quantum mechanical system with two distinguished states (e.g. the spin of an electron can be $\uparrow$ and $\downarrow$).

An *n*-qubit quantum register is the combination of n qubits.

Its quantum states are unit vectors from the vector space

$$\underbrace{\mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2}_{n \text{ times}} \cong \mathbb{C}^{2^n}.$$

with the computational basis $\{|u\rangle : u \in \{0,1\}^n\}$.

Note: if $u = a_1 a_2 \cdots a_n$ then $|u\rangle = |a_1\rangle \otimes |a_2\rangle \otimes \cdots \otimes |a_n\rangle$.

In the following let us write $\mathcal{U}_d$ for the set of unitary $(d \times d)$-matrices.

Quantum circuits

A quantum circuit is the quantum analog of a classical Boolean circuit.

An n-qubit quantum circuit operates on the state space of an n-qubit quantum register.

The circuit consists of a sequence of unitary matrices $U_1, \ldots, U_m \in \mathcal{U}_{2^n}$.

Each unitary matrix U_i should be simple but also powerful enough in the same sense as a single Boolean gate is simple.

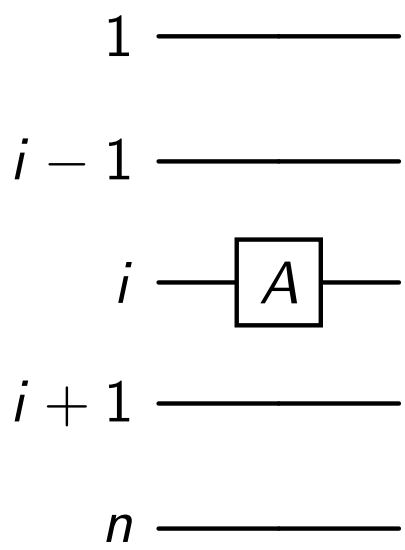
It is common to restrict the U_i to unitary transformations that operate locally on only one or two qubits (analogous to the boolean gates $\neg$, $\vee$ and $\wedge$ that operate on one or two bits).

1-qubit quantum gates

1-qubit quantum gates: $\mathrm{Id}_{2^{i-1}} \otimes A \otimes \mathrm{Id}_{2^{n-i}}$ where $A \in \mathcal{U}_2$ and $1 \leq i \leq n$.

Intuition: A operates on qubit i and does not touch the j-th qubit for every $j \in \{1, \ldots, n\} \setminus \{i\}$.

In pictures:



CNOT gate

It turns out that only one type of two-qubit quantum gates is needed: CNOT-gate (controlled not-gates).

$\mathrm{CNOT}_{i,j}$ for $1 \leq i, j \leq n$ with $i \neq j$ is the unique linear transformation that operates on the computational basis as follows, where $a_1, \ldots, a_n \in \{0, 1\}$

$$\mathrm{CNOT}_{i,j}(|a_1 \cdots a_n\rangle) = |a_1 \cdots a_{j-1}(a_j \oplus a_i)a_j \cdots a_n\rangle \,,$$

where $\oplus$ is the boolean XOR.

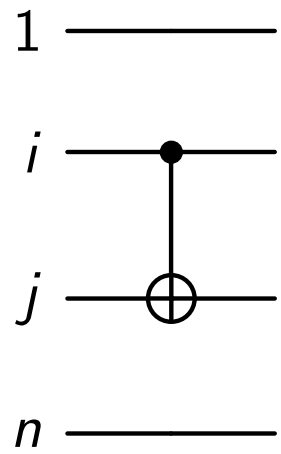
Intuition: Flip the j-th bit if the i-th bit is 1, otherwise do nothing.

Since $\mathrm{CNOT}_{i,j}$ permutes the computational basis, the columns (and rows) of $\mathrm{CNOT}_{i,j}$ form again the computational basis.

Therefore $\mathrm{CNOT}_{i,j}$ is indeed unitary.

CNOT gate

In pictures:



The matrix for $\mathrm{CNOT}_{1,2}$ is:

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

CNOT and classical copying

CNOT can be used to copy classical information: for all $a \in \{0,1\}$ we have:

$$\mathrm{CNOT}_{1,2}\,|a\,0\rangle = |a\,(0 \oplus a)\rangle = |a\,a\rangle$$

This does not contradict the following famous result of quantum computing:

No-cloning theorem

There is no unitary transformation U on $2n$ qubits such that for every n-qubit quantum state $|x\rangle \in \mathbb{C}^{2^n}$ we have

$$U(|x\rangle \otimes |0^n\rangle) = |x\rangle \otimes |x\rangle\,. \tag{3}$$

The above comment on CNOT only implies that the identity (3) can be achieved if we restrict to computational basis states $|u\rangle$ for $u \in \{0,1\}^n$.

Proof of the no-cloning theorem

Assume there is a unitary transformation U of the form excluded in the no-cloning theorem.

Take two differet quantum states $|x\rangle, |y\rangle$ (on n qubits). We get:

$$\begin{aligned}
\langle x|y\rangle &= \langle x|y\rangle \langle 0^n|0^n\rangle \\
&= (\langle x| \otimes \langle 0^n|)(|y\rangle \otimes |0^n\rangle) \\
&= (\langle x| \otimes \langle 0^n|) U^\dagger U (|y\rangle \otimes |0^n\rangle) \\
&= (\langle x| \otimes \langle x|)(|y\rangle \otimes |y\rangle) \\
&= \langle x|y\rangle \langle x|y\rangle
\end{aligned}$$

The equation $a^2 = a$ has only two solutions in $\mathbb{C}$: 0 and 1.

Hence, we have $\langle x|y\rangle \in \{0, 1\}$.

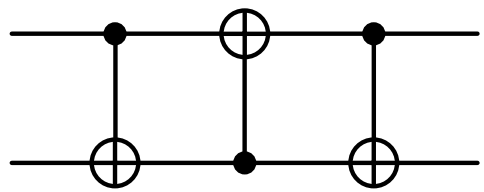
We get a contradiction if we choose x, y such that $0 \neq \langle x|y\rangle \neq 1$. □

CNOT and SWAP

CNOT can be used to swap two qubits, i.e., to compute a unitary operation SWAP (on 2 qubits) such that for all $a, b \in \{0, 1\}$:

$$\text{SWAP} \, |ab\rangle = |ba\rangle \, .$$

We need 3 CNOT-gates for this:



Controlled operations

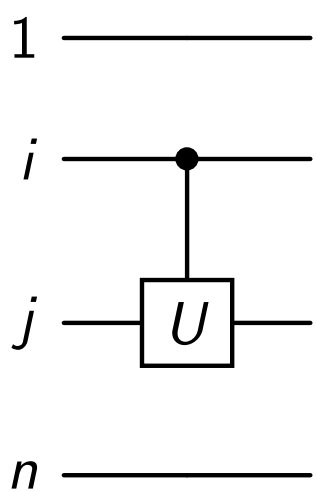
CNOT is a particular controlled operation.

Let $U \in \mathcal{U}_2$ be unitary.

$\mathsf{C}_{i,j}(U)$ for $1 \leq i, j \leq n$ with $i \neq j$ is the unique unitary transformation that operates on the computational basis as follows, where $a_1, \ldots, a_n \in \{0, 1\}$:

$$\mathsf{C}_{i,j}(U)(|a_1 \cdots a_n\rangle) = |a_1 \cdots a_{j-1}\rangle \otimes U^{a_i} |a_j\rangle \otimes |a_j \cdots a_n\rangle$$

In pictures:



Controlled operations

If the matrix for U in the computational basis is

$$\begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix}$$

then the matrix for $\mathrm{C}_{1,2}(U)$ in the computational basis is

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & a_1 & a_2 \\ 0 & 0 & a_3 & a_4 \end{pmatrix}.$$

Note that $\mathrm{CNOT}_{i,j} = \mathrm{C}_{i,j}(X)$ where X is the Pauli-X gate.

CNOT + 1-qubit quantum gates are universal

Theorem 9 (see Nielsen, Chuang, Section 4.5.2)

Every unitary transformation on n qubits can be composed from at most $\mathcal{O}(n^2 4^n)$ 1-qubit quantum gates and CNOT-gates.

The bound $\mathcal{O}(n^2 4^n)$ cannot be significantly improved (see Nielsen, Chuang, Section 4.5.4)!

There is still a problem with using arbitrary 1-qubit quantum gates:

There are uncountably many 1-qubit quantum gates and we cannot expect to find a physical implementation of every 1-qubit quantum gate.

Our goal is to approximate arbitrary 1-qubit quantum gates with high precision using a fixed finite set of 1-qubit quantum gates.

Spectral norm

For a matrix $A \in \mathbb{C}^{d \times d}$ we define its spectral norm

$$\|A\| = \max\{\|A\,|x\rangle\| : |x\rangle \in \mathbb{C}^d, \|x\| = 1\}.$$

For $A, B \in \mathbb{C}^{d \times d}$ we define their distance as $d(A, B) = \|A - B\|$.

Some facts about the spectral norm, where $A, B \in \mathbb{C}^{d \times d}$, $\alpha \in \mathbb{C}$, and $U, V \in \mathcal{U}_d$:

- $\|A + B\| \leq \|A\| + \|B\|$ (triangle inequality)
- $\|\alpha A\| = |\alpha| \cdot \|A\|$
- $A = 0$ if and only if $\|A\| = 0$
- $\|AB\| \leq \|A\| \cdot \|B\|$ (submultiplicativity)
- $\|UAV\| = \|A\|$ (unitary invariance)

Spectral norm

Proof of unitary invariance: Let $U, V \in \mathcal{U}_d$.

The mapping $|x\rangle \mapsto V|x\rangle$ induces a bijection on the set of unit vectors.

Hence, we have:

$$\begin{aligned} \|AV\| &= \max\{\|A(V|x\rangle)\| : |x\rangle \in \mathbb{C}^d, \||x\rangle\| = 1\} \\ &= \max\{\|A|y\rangle\| : |y\rangle \in \mathbb{C}^d, \||y\rangle\| = 1\} \\ &= \|A\| \end{aligned}$$

Since $\|U|y\rangle\| = \||y\rangle\|$ for all vectors $|y\rangle$, we have:

$$\begin{aligned} \|UA\| &= \max\{\|UA|x\rangle\| : |x\rangle \in \mathbb{C}^d, \||x\rangle\| = 1\} \\ &= \max\{\|A|x\rangle\| : |x\rangle \in \mathbb{C}^d, \||x\rangle\| = 1\} \\ &= \|A\| \end{aligned}$$

Spectral norm and eigenvalues

Theorem 10

For every $A \in \mathbb{C}^{d\times d}$ we have $\|A\| = \sqrt{\rho}$, where ρ be the largest eigenvalue of $A^\dagger A$ (a positive semi-definite matrix).

Proof: Since $A^\dagger A$ is positive semi-definite, there is a unitary matrix U such that

$$D := U^\dagger A^\dagger A U = \begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_d \end{pmatrix}$$

where the λ_i are the (non-negative real) eigenvalues of $A^\dagger A$.

Spectral norm and eigenvalues

We obtain the following, where w.l.o.g. $\lambda_d = \rho$ is the largest eigenvalue of the matrix $A^\dagger A$:

$$\begin{aligned}
\|A\|^2 &= \max\{\langle y| A^\dagger A |y\rangle : |y\rangle \in \mathbb{C}^d, \||y\rangle\| = 1\} \\
&= \max\{\langle x| U^\dagger A^\dagger A U |x\rangle : |x\rangle \in \mathbb{C}^d, \||x\rangle\| = 1\} \\
&= \max\{\langle x| D |x\rangle : |x\rangle \in \mathbb{C}^d, \||x\rangle\| = 1\} \\
&= \max\left\{\sum_{i=1}^d |\alpha_i|^2 \lambda_i : \alpha_1, \ldots, \alpha_d \in \mathbb{C}, \sum_{i=1}^d |\alpha_i|^2 = 1\right\} \\
&= \max\left\{\sum_{i=1}^d p_i \lambda_i : 0 \leq p_1, \ldots, p_d \leq 1, \sum_{i=1}^d p_i = 1\right\} \\
&= \lambda_d
\end{aligned}$$

Spectral norm and eigenvalues

Corollary 11

$\|U\| = 1$ for a unitary matrix U.

Corollary 12

$\|H\| = \max\{|\lambda| : \lambda \text{ is an eigenvalue of } H\}$ for a Hermitian matrix H.

Corollary 13

$\|P\| = \max\{\lambda : \lambda \text{ is an eigenvalue of } P\}$ for a positive-semi-definite P.

Spectral norm and tensor products

Lemma 14

$\|A \otimes B\| = \|A\| \cdot \|B\|$ for every $A \in \mathbb{C}^{d \times d}$, $B \in \mathbb{C}^{e \times e}$.

Proof: We have $(A \otimes B)^\dagger (A \otimes B) = (A^\dagger \otimes B^\dagger) \cdot (A \otimes B) = A^\dagger A \otimes B^\dagger B$.

The matrix $A^\dagger A \otimes B^\dagger B$ is positive semi-definite
(since $A^\dagger A$ and $B^\dagger B$ are positive semi-definite).

Choose orthonormal bases

- $|x_1\rangle, \ldots, |x_d\rangle$ of $\mathbb{C}^d$ consisting of eigenvectors of $A^\dagger A$ and
- $|y_1\rangle, \ldots, |y_e\rangle$ of $\mathbb{C}^e$ consisting of eigenvectors of $B^\dagger B$ and let
- λ_i be the eigenvalue of $A^\dagger A$ for $|x_i\rangle$, where $0 \leq \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_d$,
- μ_i be the eigenvalue of $B^\dagger B$ for $|y_i\rangle$, where $0 \leq \mu_1 \leq \mu_2 \leq \cdots \leq \mu_e$.

Spectral norm and tensor products

Then $\{|x_i\rangle \otimes |y_j\rangle : 1 \leq i \leq d, 1 \leq j \leq e\}$ is an orthonormal basis of eigenvectors for $A^\dagger A \otimes B^\dagger B$.

The eigenvalue for $|x_i\rangle \otimes |y_j\rangle$ is $\lambda_i \mu_j$.

These must be all eigenvalues! There is no more space for further eigenvalues.

Hence, the largest eigenvalue of $(A \otimes B)^\dagger (A \otimes B)$ is

$$\lambda_d \cdot \mu_e = \|A\|^2 \cdot \|B\|^2 .$$

We get $\|A \otimes B\| = \sqrt{\lambda_d \cdot \mu_e} = \|A\| \cdot \|B\|$. □

The Solovay-Kitaev Theorem

A subset $S \subseteq \mathcal{U}_d$ of d-dimensional unitary matrices is dense if for all $V \in \mathcal{U}_d$ and $\epsilon > 0$ there is product $U = U_1 U_2 \cdots U_k$ such that

- $U_1, U_2, \ldots, U_k \in S$ and
- $d(U, V) \leq \epsilon$.

S is inverse-closed if $U^{-1} \in S$ for every $U \in S$.

Theorem 15 (Solovay 1995, Kitaev 1997)

Fix a dimension d and let $S \subseteq \mathcal{U}_d$ be finite, dense and inverse-closed. Then for all $V \in \mathcal{U}_d$ and $\epsilon > 0$ there is product $U = U_1 U_2 \cdots U_k$ with

- $U_1, U_2, \ldots, U_k \in S$,
- $d(U, V) \leq \epsilon$ and
- $k \leq \mathcal{O}(\log^c(1/\epsilon))$ for some constant c.

The Solovay-Kitaev Theorem

Remarks:

- In the first proofs of the Solovay-Kitaev theorem, the constant c was $3 + \delta$, where $\delta > 0$ can be chosen arbitrarily small.
- Recently, Greg Kuperberg gave a new proof showing that one can take $c = 1.44042\cdots + \delta$ for $\delta > 0$ arbitrarily small.
- Given a good approximation of V, one can compute the sequence $U_1 U_2 \cdots U_k$ also efficiently.

We are mainly interested in the case $d = 2$ (unitary transformations on 1 qubit).

Fortunately, there is a finite and dense subset of $\mathcal{U}_2$.

A universal gate set

Theorem 16 (see Nielsen and Chuang, Section 4.5.3)

The set consisting of

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \text{ and } T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$$

is a dense subset of $\mathcal{U}_2$.

H is called the Hadamard matrix (or the Hadamard gate), T is called the $\pi/8$ gate.

Hence, by the Solovay-Kitaev Theorem every $U \in \mathcal{U}_2$ can be approximated using H and T up to error $\epsilon > 0$ using a sequence of length $\mathcal{O}(\log^c(1/\epsilon))$.

Getting the Pauli matrices from H and T

Recall the Pauli matrices X, Y, Z from Slide 39.

- $Z = T^4$
- $X = HZH = HT^4H$
- $iY = XZ = HT^4HT^4$

In the last line, the factor i is a physically irrelevant global factor.

Error approximation in quantum circuits

Assume we have an n-qubit quantum circuit $U_1, U_2, \ldots, U_m$.

Every U_i is either a

- CNOT-gates or an
- arbitrary 1-qubit quantum gate $\mathrm{Id}_{2^{i-1}} \otimes A \otimes \mathrm{Id}_{2^{n-i}}$ where $A \in \mathcal{U}_2$.

Assume we have approximated (using H and T) every A in the 2nd point by a unitary $B \in \mathcal{U}_2$ with $d(A, B) \leq \epsilon$ (using Solovay-Kitaev).

By Lemma 14 we have:

$$\begin{aligned} & d(\mathrm{Id}_{2^{i-1}} \otimes A \otimes \mathrm{Id}_{2^{n-i}}, \mathrm{Id}_{2^{i-1}} \otimes B \otimes \mathrm{Id}_{2^{n-i}}) \\ = \; & \|\mathrm{Id}_{2^{i-1}} \otimes A \otimes \mathrm{Id}_{2^{n-i}} - \mathrm{Id}_{2^{i-1}} \otimes B \otimes \mathrm{Id}_{2^{n-i}}\| \\ = \; & \|\mathrm{Id}_{2^{i-1}} \otimes (A - B) \otimes \mathrm{Id}_{2^{n-i}}\| \\ = \; & \|\mathrm{Id}_{2^{i-1}}\| \cdot \|A - B\| \cdot \|\mathrm{Id}_{2^{n-i}}\| \\ = \; & \|A - B\| = d(A, B) \leq \epsilon. \end{aligned}$$

Error approximation in quantum circuits

Lemma 17

Let $U_1, V_1, \ldots, U_m, V_m \in \mathcal{U}_d$ with $d(U_i, V_i) \leq \epsilon$ for all $1 \leq i \leq m$.
Then we have $d(U_1 U_2 \cdots U_m, V_1 V_2 \cdots V_m) \leq m\epsilon$.

Proof: Induction over m.

The case $m = 1$ is clear.

Assume now that $m > 1$.

Define $A = U_1 U_2 \cdots U_{m-1}$ and $B = V_1 V_2 \cdots V_{m-1}$.

The induction hypothesis tells us that $d(A, B) \leq (m-1)\epsilon$.

We obtain:

Error approximation in quantum circuits

$$
\begin{aligned}
d(AU_m, BV_m) &= \|AU_m - BV_m\| \\
&= \|AU_m - AV_m + AV_m - BV_m\| \\
&= \|A(U_m - V_m) + (A - B)V_m\| \\
&\leq \|A(U_m - V_m)\| + \|(A - B)V_m\| \\
&\leq \epsilon + (m-1)\epsilon \;=\; m\epsilon
\end{aligned}
$$

These are good news: If we want to approximate the quantum circuit up to precision δ, we have to approximate every 1-qubit quantum gate with an error bounded by δ/m.

By the Solovay-Kitaev theorem, this can be achieved by replacing every 1-qubit quantum gate by a sequence of H-gates and T-gates of length $\mathcal{O}(\log^c(m/\delta)) = \mathcal{O}((\log m - \log \delta)^c)$.

Measurements in quantum circuits

We will usually use a quantum circuit $U_1, U_2, \ldots, U_m$ working on n qubits in the following way:

1. We prepare a quantum state $|x\rangle \in \mathbb{C}^{2^n}$ (usually, it is obtained from some classical input string $u \in \{0,1\}^*$).
2. We apply the quantum circuit (first U_1, then U_2, etc) and obtain the quantum state $|y\rangle = U_m \cdots U_2 U_1 |x\rangle$.
3. We extract classical information from $|y\rangle$ by doing a projective measurement.

By the lemma on the next slide, a good approximation of the quantum circuit yields a good approximation of the measurement probabilities.

Measurements in quantum circuits

Lemma 18

Let $|x\rangle \in \mathbb{C}^d$ be a quantum state (a unit vector), $U, V \in \mathcal{U}_d$ such that $d(U,V) \leq \epsilon$ and Π a projector of dimension d.

Then we have: $|\langle x| U^\dagger \Pi U |x\rangle - \langle x| V^\dagger \Pi V |x\rangle| \leq 2\epsilon$.

Proof: With $|y\rangle = (U - V)|x\rangle$ we have

$$
\begin{aligned}
& |\langle x| U^\dagger \Pi U |x\rangle - \langle x| V^\dagger \Pi V |x\rangle| \\
= \; & |\langle x| U^\dagger \Pi U |x\rangle - \langle x| U^\dagger \Pi V |x\rangle + \langle x| U^\dagger \Pi V |x\rangle - \langle x| V^\dagger \Pi V |x\rangle| \\
= \; & |\langle x| U^\dagger \Pi |y\rangle + \langle y| \Pi V |x\rangle| \\
\leq \; & |\langle x| U^\dagger \Pi |y\rangle| + |\langle y| \Pi V |x\rangle| \\
\leq \; & \||y\rangle\| + \||y\rangle\| \quad \text{(here we apply Cauchy-Schwarz, see slide 28)} \\
\leq \; & 2d(U,V) \leq 2\epsilon.
\end{aligned}
$$

Measurements in quantum circuits

Similar to the unitary operations in a quantum circuit we should also restrict the allowed measurements at the end.

Most of the time, we measure only a single qubit, say the 1st of the n qubits on which the quantum circuit works.

This means that the measurement is given by the projectors Π_0, Π_1, where

$$\Pi_0 = |0\rangle\langle 0| \otimes \mathrm{Id}_{2^{n-1}} \quad \text{and} \quad \Pi_1 = |1\rangle\langle 1| \otimes \mathrm{Id}_{2^{n-1}}.$$

Π_0 (Π_1) projects on the subspace spanned by all basis vectors $|0y\rangle$ ($|1y\rangle$) for $y \in \{0,1\}^{n-1}$.

Note that we have $\Pi_0\Pi_1 = \Pi_1\Pi_0 = 0$ and

$$\begin{aligned} \Pi_0 + \Pi_1 &= |0\rangle\langle 0| \otimes \mathrm{Id}_{2^{n-1}} + |1\rangle\langle 1| \otimes \mathrm{Id}_{2^{n-1}} \\ &= (|0\rangle\langle 0| + |1\rangle\langle 1|) \otimes \mathrm{Id}_{2^{n-1}} = \mathrm{Id}_{2^n}. \end{aligned}$$

Measurements in quantum circuits

If the current quantum state is $|x\rangle \in \mathbb{C}^{2^n}$ then the probability of measuring 1 in the first qubit is

$$\langle x| \Pi_1 |x\rangle = \langle x| \left(|1\rangle\langle 1| \otimes \mathrm{Id}_{n-1} \right) |x\rangle .$$

Assume that $|x\rangle = \sum_{u \in \{0,1\}^n} \alpha_u |u\rangle$ and write

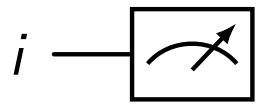
$$\begin{aligned} |x\rangle &= \sum_{v \in \{0,1\}^{n-1}} \Big(\alpha_{1v}(|1\rangle \otimes |v\rangle) + \alpha_{0v}(|0\rangle \otimes |v\rangle) \Big) \\ &= |1\rangle \otimes \underbrace{\left(\sum_{v \in \{0,1\}^{n-1}} \alpha_{1v} |v\rangle \right)}_{|x_1\rangle} + |0\rangle \otimes \underbrace{\left(\sum_{v \in \{0,1\}^{n-1}} \alpha_{0v} |v\rangle \right)}_{|x_0\rangle} \end{aligned}$$

This yields $\langle x| \left(|1\rangle\langle 1| \otimes \mathrm{Id}_{2^{n-1}} \right) |x\rangle = \langle x_1 | x_1 \rangle = \sum_{v \in \{0,1\}^{n-1}} |\alpha_{1v}|^2$.

Measurements in quantum circuits

Measuring another qubit is done analogously.

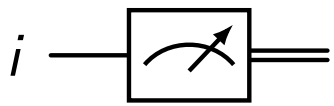
The symbol for measuring the i-th qubit in a quantum circuit is:



In quantum algorithms, measurements are often performed in the middle of a computation.

The measured qubit is thereby transformed into a classical bit: it is either 0 or 1.

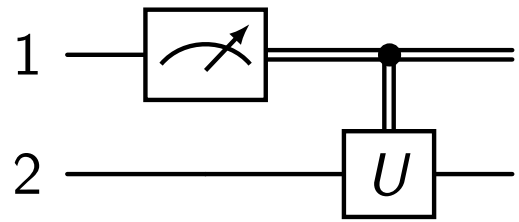
Classical bits are shown as double lines in quantum circuits:



The value of such a classical bit can determine which quantum gates are performed after the measurement.

The principle of deferred measurement

Example: In the following circuit, U is only performed on the 2nd qubit if the result of measuring the 1st qubit is 1.



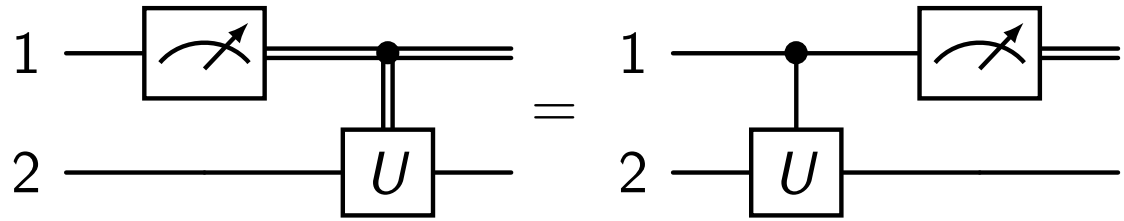
This is called a classical-controlled quantum gate.

Principle of deferred measurement

Measurements can always be moved to the end of the computation. Classical-controlled quantum gates are then replaced by (quantum)-controlled quantum gates.

The principle of deferred measurement

Example:



This can be shown as follows:

Let $|x\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$ be the initial quantum state.

Left circuit: Let $p_0 = |\alpha_{00}|^2 + |\alpha_{01}|^2$ and $p_1 = |\alpha_{10}|^2 + |\alpha_{11}|^2$.

The measurement yields the mixed state

$$\left\{ \left(p_0, \frac{\alpha_{00}|00\rangle + \alpha_{01}|01\rangle}{\sqrt{p_0}} \right), \left(p_1, \frac{\alpha_{10}|10\rangle + \alpha_{11}|11\rangle}{\sqrt{p_1}} \right) \right\}$$

The principle of deferred measurement

The classical-controlled U-gate finally yields the mixed state

$$\left\{\left(p_0, \frac{\alpha_{00}\,|00\rangle + \alpha_{01}\,|01\rangle}{\sqrt{p_0}}\right), \left(p_1, \frac{\alpha_{10}\,|1\rangle \otimes U\,|0\rangle + \alpha_{11}\,|1\rangle \otimes U\,|1\rangle}{\sqrt{p_1}}\right)\right\}$$

Right circuit: The controlled U gate yields the pure state

$$\alpha_{00}\,|00\rangle + \alpha_{01}\,|01\rangle + \alpha_{10}\,|1\rangle \otimes U\,|0\rangle + \alpha_{11}\,|1\rangle \otimes U\,|1\rangle\,.$$

The measurement finally yields the mixed state

$$\left\{\left(p_0, \frac{\alpha_{00}\,|00\rangle + \alpha_{01}\,|01\rangle}{\sqrt{p_0}}\right), \left(p_1, \frac{\alpha_{10}\,|1\rangle \otimes U\,|0\rangle + \alpha_{11}\,|1\rangle \otimes U\,|1\rangle}{\sqrt{p_1}}\right)\right\}.$$

Acceptance probability of a quantum circuit

For a quantum circuit $Q = U_1, \ldots, U_m$ working on n quibts and a quantum state $|x\rangle \in \mathbb{C}^{2^n}$ we define its acceptance probability:

$$\text{Prob}[Q \text{ accepts } |x\rangle] = \langle x| U_1^\dagger \cdots U_m^\dagger \Pi_1 U_m \cdots U_1 |x\rangle .$$

In other words:

- We first apply the quantum circuit and obtain $|y\rangle = U_m \cdots U_1 |x\rangle$.
- We then measure the first qubit in $|y\rangle$.
- $|x\rangle$ is accepted if the 1st qubit is 1 in the postmeasurement state, i.e., the latter is $|1\rangle \otimes |z\rangle$ for some $(n-1)$-qubit state $|z\rangle$.

  This happens with probability $\langle y| \Pi_1 |y\rangle$.

Often, we will also measure another qubit in $|y\rangle$ instead of the 1st one; this makes no real difference.

If $u \in \{0,1\}^n$ then we define: $\text{Prob}[Q \text{ accepts } u] = \text{Prob}[Q \text{ accepts } |u\rangle]$.

P-uniform families of quantum circuits

In the following we only consider quantum circuits that are composed of CNOT-gates, H-gates and T-gates, and where the first qubit is measured at the end (unless otherwise stated).

For a quantum circuit $Q = U_1, U_2, \ldots, U_m$ we define its size $|Q| = m$.

For a quantum state $|x\rangle$ we also write $Q|x\rangle$ for $U_m \cdots U_2 U_1 |x\rangle$ in the following.

Consider a family $(Q_n)_{n\geq 0}$ of such quantum circuits, where every Q_n works on $p(n)$ qubits for a polynomial $p(n) \geq n$.

$(Q_n)_{n\geq 0}$ is called **P**-uniform if there is a PTM M that produces on input 1^n (n 1-bits) a binary encoding of Q_n.

Note: this implies that there exists a polynomial $q(n)$ with $|Q_n| \leq q(n)$.

The class **promiseBQP**

The class **promiseBQP** consists of all promise problems (L_0, L_1) for which there exists a **P**-uniform quantum circuit family $(Q_n)_{n\geq 0}$ and a polynomial $a(n)$ such that Q_n works on $n + a(n)$ qubits and for every $u \in \{0,1\}^n$ the following hold:

- If $u \in L_1$ then $\text{Prob}[Q_n \text{ accepts } u0^{a(n)}] \geq \frac{2}{3}$.
- If $u \in L_0$ then $\text{Prob}[Q_n \text{ accepts } u0^{a(n)}] \leq \frac{1}{3}$.

Remark:

- In the beginning, the first n qubits are set to the classical bits from the input $u \in \{0,1\}^n$.
- The other $a(n)$ qubits are called ancilla qubits and initialized to 0.
- Ancilla qubits are needed to make the whole computation reversible.
- In the following we will simply say **BQP** instead of **promiseBQP**; it stands for bounded-error quantum polynomial time.

$\mathbf{BPP} \subseteq \mathbf{BQP}$

BQP is usually identified with the class of all (promise) problems that can be solved efficiently using a quantum computer.

In the following, we also refer implicitly for classical complexity classes (like **P**, **BPP** or **PSPACE**) always to the corresponding promise classes.

Theorem 19

$\mathbf{BPP} \subseteq \mathbf{BQP}$

Proof: We first show that $\mathbf{P} \subseteq \mathbf{BQP}$.

By Theorem 1 (slide 21) it suffices to simulate a classical Boolean circuit by a quantum circuit.

BPP $\subseteq$ BQP

What does it mean to simulate a Boolean circuit by a quantum circuit?

Boolean circuits are in general irreversible (i.e., one cannot reverse the computation).

For instance the binary AND-function $(a, b) \mapsto a \wedge b$ is irreversible.

Quantum circuits are always reversible (if we omit the final measurement).

Solution: simulate the Boolean function $f : \{0,1\}^n \to \{0,1\}$ by the bijection $\tilde{f} : \{0,1\}^{n+1} \to \{0,1\}^{n+1}$ with:

$$\forall u \in \{0,1\}^n \; \forall a \in \{0,1\} : \tilde{f}(u\, a) \;=\; u\,(a \oplus f(u)),$$

where $\oplus$ denotes the XOR (addition modulo two).

BPP $\subseteq$ BQP

In particular: $\tilde{f}(\tilde{f}(w)) = w$ for all $w \in \{0,1\}^{n+1}$ and $\tilde{f}(u\,0) = u\,f(u)$.

We say that a quantum circuit Q on $(n+1+k)$ qubits computes the boolean function $f : \{0,1\}^n \to \{0,1\}$ if for all

$$\forall u \in \{0,1\}^n \; \forall a \in \{0,1\} : Q\,|u\,a\,0^k\rangle = |u\,(f(u) \oplus a)\;0^k\rangle\,.$$

Consider now a **P**-uniform boolean circuit family $(C_n(x_1,\ldots,x_n))_{n\geq 1}$.

Goal: compute $f_n := f_{C_n} : \{0,1\}^n \to \{0,1\}$ by a quantum circuit Q_n.

We can assume that C_n is built up from NAND-gates.

NAND stands for not-and and it works as follows:

$$\mathrm{NAND}(0,0) = \mathrm{NAND}(0,1) = \mathrm{NAND}(1,0) = 1, \quad \mathrm{NAND}(1,1) = 0$$

BPP $\subseteq$ BQP

NAND is universal; it can simulate AND, OR and NOT.

First consider the so-called Toffoli-gate, also called CCNOT:

$$\forall a, b, c \in \{0,1\} : \mathrm{CCNOT}\,|a\,b\,c\rangle = |a\,b\,(a \wedge b) \oplus c\rangle\,.$$

Flip the 3rd qubit if the 1st and 2nd qubit are 1, otherwise do nothing.

We have $\mathrm{CCNOT}\,|a\,b\,1\rangle = |a\,b\,\mathrm{NAND}(a,b)\rangle$.

The ancilla 1-qubit can be obtained from an ancilla 0-qubit using a Pauli-X.

Using SWAP, CCNOT and copying of classical bits (see Slide 70) we can build for every $n \geq 1$ a quantum circuit Q'_n such that

$$\forall u \in \{0,1\}^n : Q'_n\,|u\,0^{k_n+1}\rangle \mapsto |u\,f_n(u)\,g_n(u)\rangle\,,$$

where $g_n(u) \in \{0,1\}^{k_n}$ is garbage produced during the computation.

BPP $\subseteq$ BQP

We can get rid of the garbage $g_n(u)$ as follows
(we write $k = k_n$, $f = f_n$, $g = g_n$ below):

Add an additional qubit a (that will be an ancilla-0 qubit at the end):

Starting from a computational basis state $|u\,0\,0^k\,a\rangle$ with $u \in \{0,1\}^n$ and $a \in \{0,1\}$, our final quantum circuit Q_n behaves as follows:

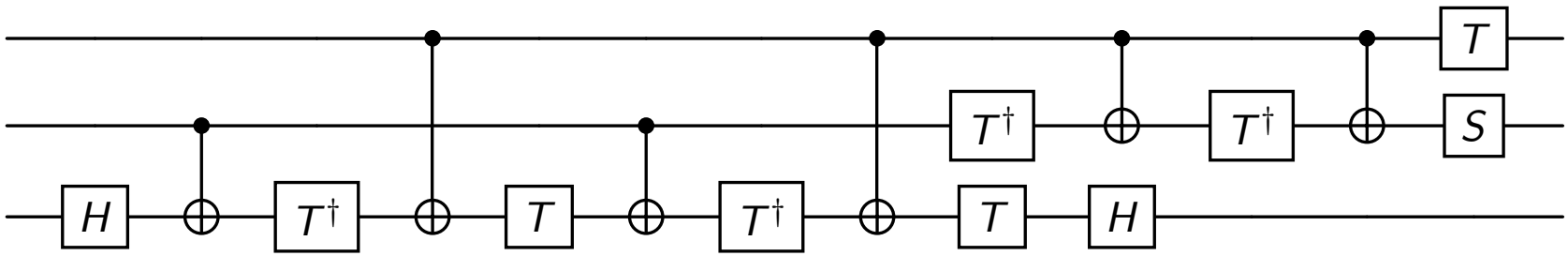
- Apply Q'_n on the first $n+1+k$ qubits to get $|u\,f(u)\,g(u)\,a\rangle$
- With a CNOT we get $|u\,f(u)\,g(u)\,(f(u) \oplus a)\rangle$
- Reversing the first step (apply $(Q'_n)^\dagger$) yields $|u\,0\,0^k\,(f(u) \oplus a)\rangle$

This is called the compute-uncompute trick.

Note: If $Q'_n = U_1, U_2, \ldots, U_m$ then $(Q'_n)^\dagger = U_m^\dagger, \ldots, U_2^\dagger, U_1^\dagger$.

$\mathbf{BPP} \subseteq \mathbf{BQP}$

CCNOT can be obtained from H, T and CNOT as follows, where $S = T^2$ (note that $T^\dagger = T^{-1} = T^7$):



(see Nielsen, Chuang, page 182).

This finally yields the desired quantum circuit Q_n.

Q_n is error-free: Applying Q_n to the computational basis state $|u\,0^{k+2}\rangle$ results in the computational basis state $|u\,0^{k+1}\,f_n(u)\rangle$ and measuring the last qubit gives the output bit $f_n(u)$ with probability 1.

Finally note that since the Boolean circuit family $(C_n)_{n\geq 0}$ is **P**-uniform, the same holds for the quantum circuit family $(Q_n)_{n\geq 0}$.

$\mathbf{BPP} \subseteq \mathbf{BQP}$

This shows $\mathbf{P} \subseteq \mathbf{BQP}$.

It remains to show $\mathbf{BPP} \subseteq \mathbf{BQP}$. Let $L \in \mathbf{BPP}$.

Recall: In **BPP** we have $r(n)$ random bits available (where n is the length of the input). These are randomly set to 0 or 1.

Let C_n be a Boolean circuit with $n + r(n)$ inputs such that for all $u \in \{0,1\}^n$ we have (with $f_n = f_{C_n}$):

- If $u \in L_1$ then $f_n(u, v) = 1$ for $\geq \frac{2}{3} \cdot 2^{r(n)}$ many $v \in \{0,1\}^{r(n)}$.
- If $u \in L_0$ then $f_n(u, v) = 1$ for $\leq \frac{1}{3} \cdot 2^{r(n)}$ many $v \in \{0,1\}^{r(n)}$.

BPP $\subseteq$ BQP

In the quantum circuit Q_n (for input length n) the $r(n)$ random bits from C_n are $r(n)$ many additional ancilla qubits that are initially set to 0.

To these ancilla qubits we apply $H^{\otimes r(n)} = \underbrace{H \otimes H \otimes \cdots \otimes H}_{r(n) \text{ many}}$.

Then we apply a quantum circuit that computes the Boolean function f_n (using k additional ancilla bits):

$$|u\, 0^{r(n)}\, 0^k\rangle \xrightarrow{H^{\otimes r(n)}} \frac{1}{2^{r(n)/2}} \cdot \sum_{v \in \{0,1\}^{r(n)}} |u\, v\, 0^k\rangle$$

$$\xrightarrow{\text{quantum circuit for } f_n} \frac{1}{2^{r(n)/2}} \cdot \sum_{v \in \{0,1\}^{r(n)}} |u\, v\, 0^{k-1}\, f_n(u,v)\rangle$$

Measuring now the last qubit yields the same acceptance probability as the original boolean circuit. $\square$

Error reduction for **BQP**

Theorem 20

The error probability $1/3$ in the definition of **BQP** can be replaced by $2^{-r(n)}$ where $n =$ input length and $r(n)$ is any polynomial.

Proof sketch: Assume we have a promise problem $(L_0, L_1) \in \mathbf{BQP}$.

Let $(Q_n)_{n \geq 1}$ be the corresponding quantum circuit family with error probability $\leq 1/3$ and let $Q_n = U_1, U_2, \ldots, U_{p(n)}$.

Fix an input $u \in \{0,1\}^n$ of length n. We ignore ancilla qubits below.

Then we can write $Q_n |u\rangle = |0\rangle \otimes |x_0\rangle + |1\rangle \otimes |x_1\rangle$ for $|x_0\rangle, |x_1\rangle \in \mathbb{C}^{2^{n-1}}$ (see slide 93) and we have:

- If $u \in L_1$ then $\langle x_0 | x_0 \rangle \leq 1/3$ (and $\langle x_1 | x_1 \rangle = 1 - \langle x_0 | x_0 \rangle \geq 2/3$).
- If $u \in L_0$ then $\langle x_1 | x_1 \rangle \leq 1/3$ (and $\langle x_0 | x_0 \rangle = 1 - \langle x_1 | x_1 \rangle \geq 2/3$).

Error reduction for **BQP**

The idea of the proof is the same as for **BPP**: execute $k = \Theta(r(n))$ independent copies of Q_n on input u and make a majority vote.

Formally we define $Q_n^{\otimes k} = U_1^{\otimes k}, U_2^{\otimes k}, \ldots, U_{p(n)}^{\otimes k}$.
Note that $|Q_n^{\otimes k}| = k \cdot p(n)$.

Step 1: Produce from the classical input u the state $|u\rangle^{\otimes k}$ using CNOT gates (see slide 70).

Step 2: Apply $Q_n^{\otimes k}$. This results in the state $\big(|0\rangle \otimes |x_0\rangle + |1\rangle \otimes |x_1\rangle \big)^{\otimes k}$ or, after applying SWAP operations, ($v[i]$ is the i-th bit in the bit string v):

$$\sum_{v \in \{0,1\}^k} |v\rangle \otimes \bigotimes_{i=1}^{k} |x_{v[i]}\rangle \, .$$

Error reduction for **BQP**

W.l.o.g. we assume $k = 2^\ell - 1$ in the following.

Step 3: One could now measure the first k bits and make a majority vote (i.e., accept iff at least $k/2$ of the measured bits are 1).

Alternatively one can also apply to the first k qubits plus ℓ many ancilla qubits w the transformation $|v\rangle \otimes |w\rangle \to |v\rangle \otimes |w \oplus \mathrm{bin}(|v|_1)\rangle$.

Here, $|v|_1$ is the number of 1's in v (a number in $[0, 2^\ell - 1]$) and $\mathrm{bin}(|v|_1) \in \{0,1\}^\ell$ its binary representation.

Assuming $w = 0^\ell$ in the beginning, we get the state

$$\sum_{v \in \{0,1\}^k} |v\rangle \otimes \bigotimes_{i=1}^{k} |x_{v[i]}\rangle \otimes |\mathrm{bin}(|v|_1)\rangle \,.$$

Error reduction for **BQP**

Step 4: We then measure the 1st qubit of $|\text{bin}(|v|_1)\rangle$.

Note: the first bit of $\text{bin}(|v|_1)$ is 1 if and only if $|v|_1 \geq k/2$.

Therefore, the probability that in the post-measurement state the measured qubit is 1 is

$$\sum_{v \in \{0,1\}^k, |v|_1 \geq k/2} \prod_{i=1}^{k} \langle x_{v[i]} | x_{v[i]} \rangle \,. \tag{4}$$

Assume $u \in L_0$ (the case $u \in L_1$ is analogous). Then (4) is the error probability and we have $\langle x_1 | x_1 \rangle \leq 1/3$ and $\langle x_0 | x_0 \rangle \geq 2/3$.

The probability (4) can be obtained also by taking k independent Bernoulli random variables $X_i \in \{0, 1\}$ with $\text{Prob}[X_i = 1] = \langle x_1 | x_1 \rangle \leq 1/3$ for all i.

Then (4) $= \text{Prob}[\sum_{i=1}^{k} X_i \geq k/2] \leq e^{-\Theta(k)}$ by the Chernoff bound. □

Error reduction for **BQP**

The ability to do error reduction is important for **BQP**.

Assume that we solve a problem with a **P**-uniform family $(Q_n)_{n\geq 0}$ of quantum circuits using only d-qubit quantum gates for a constant d.

Let $p(n)$ be the number of gates in Q_n (a polynomial) and let $1/3$ be the error probability.

- Theorem 9 (slide 75) $\rightarrow$ quantum circuit of size $\mathcal{O}(p(n))$ consisting of CNOT and 1-qubit gates.
- Error reduction $\rightarrow$ error probability can be reduced to $1/6$.

  Thereby the number of gates only increases by a constant.

  Solovay-Kitaev (slide 83) & Lemma 18 (slide 91) with $\epsilon = 1/12 \rightarrow$ quantum circuit with $\mathcal{O}(p(n)\log^c p(n)) = \mathcal{O}(p(n)\log^c n)$ CNOT, H, and T gates and error of $1/6 + 2/12 = 1/3$.

BQP-complete problems

There are promise problems that are **BQP**-complete with respect to Karp polynomial time reductions, but their formulation is usually a bit technical.

Example: MI (matrix inversion) is the following problem (see next slide for further definitions):

INPUT: a succinctly specified sparse Hermitian invertible matrix $A \in \mathbb{C}^{2^n \times 2^n}$ with $\kappa(A) \leq \text{poly}(n)$.

OUTPUT: Let $|x\rangle$ be a unit vector such that $|x\rangle = \alpha \cdot A^{-1} |0^n\rangle$ for a normalization factor $\alpha \in \mathbb{C}$. Then the output is

- 1 if $\langle x| \Pi_1 |x\rangle \geq 2/3$,
- 0 if $\langle x| \Pi_1 |x\rangle \leq 1/3$.

BQP-complete problems

- Sparse means that A has at most c entries in every row, where c is some fixed constant (it should be not too small in order to get **BQP**-hardness).
- Succinctly specified means that A is given by a Boolean circuit that computes from the binary encoding of a row index $i \in [0, 2^n - 1]$ the at most c entries in row i and their positions.
- $\kappa(A)$ is the condition number of the Hermitian matrix A:

$$\kappa(A) = \frac{\lambda_{\mathsf{max}}}{\lambda_{\mathsf{min}}}$$

  with λ_{min} (λ_{max}) the smallest (largest) eigenvalue of A.

Membership of **MI** in **BQP** is a direct consequence of the famous HHL-algorithm (Harrow, Hassidim, Lloyd).

This paper contains also the idea for **BQP**-hardness.

BQP-complete problems

Other **BQP**-complete problems deal for instance with

- matrix powering,
- computing an additive approximation of the Jones polynomial of a knot at a root of unity.

The class **MA**

MA stands for Merlin-Arthur (we should call it **promiseMA**, but we omit the **promise**) and is defined as follows:

A promise problem (L_0, L_1) belongs to **MA** if there is a PTM M and polynomials $p(n), r(n)$ such that for every $u \in \{0,1\}^n$:

- If $u \in L_1$ then $\exists v \in \{0,1\}^{p(n)}$: $\langle u, v, w\rangle \in L(M)$ for $\geq \frac{2}{3} \cdot 2^{r(n)}$ many $w \in \{0,1\}^{r(n)}$
- If $u \in L_0$ then $\forall v \in \{0,1\}^{p(n)}$: $\langle u, v, w\rangle \in L(M)$ for $\leq \frac{1}{3} \cdot 2^{r(n)}$ many $w \in \{0,1\}^{r(n)}$.

MA can be seen as a randomized version of **NP** (inparticular **NP** $\subseteq$ **MA**):

- Merlin proposes to Arthur a "proof" v for $u \in L_1$.
- Arthur can verify in polynomial with high probability whether it is really a proof.

The class **QMA**

We now define a quantum analog of the class **NP** (actually a quantum analog of **MA**).

A promise problem (L_0, L_1) belongs to **QMA** if there are polynomials $p(n)$, $a(n)$ and a **P**-uniform quantum circuit family $(Q_n)_{n\geq 0}$ working on $n + a(n) + p(n)$ qubits such that for every $u \in \{0,1\}^n$:

- If $u \in L_1$ then there is a quantum state $|x\rangle \in \mathbb{C}^{2^{p(n)}}$:

$$\text{Prob}[Q_n \text{ accepts } |u\, 0^{a(n)}\rangle \otimes |x\rangle] \geq \frac{2}{3}. \tag{5}$$

- If $u \in L_0$ then for every quantum state $|x\rangle \in \mathbb{C}^{2^{p(n)}}$:

$$\text{Prob}[Q_n \text{ accepts } |u\, 0^{a(n)}\rangle \otimes |x\rangle] \leq \frac{1}{3}. \tag{6}$$

The class **QMA** and its relatives

Remarks:

- the "quantum proof" $|x\rangle$ is an arbitrary $p(n)$-qubit quantum state and not necessarily a computational basis state.
- If one requires that $|x\rangle$ is a computational basis state $|v\rangle$ with $v \in \{0,1\}^{p(n)}$ then one obtains the class **QCMA**.
- **MA** $\subseteq$ **QCMA**: basically the same proof as for **BPP** $\subseteq$ **BQP**.
- **QCMA** $\subseteq$ **QMA**: almost obvious, one only has to show that if (6) holds for all classical proofs $|x\rangle = |v\rangle$ with $v \in \{0,1\}^{p(n)}$ then (6) holds for all quantum proofs $|x\rangle$ (Exercise).
- **BQP** $\subseteq$ **QCMA**: the quantum proof can be ignored.

Error reduction for **QMA**

Similar to **BQP** we would like to reduce the error $1/3$ in our definition of **QMA** to $2^{-r(n)}$ for a polynomial $r(n)$.

There are two ways to do this:

- error reduction by parallel repetition: Similarly to error reduction for **BQP** one runs $k = \Theta(r(n))$ copies of the proof verification quantum circuit.

  If $|x\rangle$ is the proof in (136), then one can choose the proof $|x\rangle^{\otimes k}$.

  Note: In (6) one has to consider all $kp(n)$-qubit quantum states $|y\rangle$ and not only quantum states of the form $|x\rangle^{\otimes k}$ (but one can argue that this is not a problem).

- witness-preserving error reduction: avoids duplicating the proof.

Witness-preserving error reduction for **QMA**

Fix a quantum circuit family $(Q_n)_{n\geq 0}$ as on slide 118 and an input $u \in \{0,1\}^n$.

In the following, it is useful to rewrite the acceptance probability $\mathrm{Prob}[Q_n \text{ accepts } |u\,0^{a(n)}\rangle \otimes |x\rangle]$ for a quantum proof $|x\rangle$.

We use the following short-hand notations:

- $\tilde{u} = u\,0^{a(n)}$ for $u \in \{0,1\}^n$.
- $A_n = Q_n^\dagger \Pi_1 Q_n$.
- $P_u = \left(\langle \tilde{u}| \otimes \mathrm{Id}_{2^{p(n)}} \right) \cdot A_n \cdot \left(|\tilde{u}\rangle \otimes \mathrm{Id}_{2^{p(n)}} \right)$

Lemma 21

For every $p(n)$-qubit quantum state $|x\rangle$ we have

$$\mathrm{Prob}[Q_n \text{ accepts } |\tilde{u}\rangle \otimes |x\rangle] = \mathrm{Tr}(P_u\,|x\rangle\langle x|) = \langle x|\,P_u\,|x\rangle\,.$$

Witness-preserving error reduction for **QMA**

Proof: In the following, we write Id for $\mathrm{Id}_{2^{p(n)}}$.

$$\begin{aligned}
& \mathrm{Prob}[Q_n \text{ accepts } |\tilde{u}\rangle \otimes |x\rangle] \\
= \; & (\langle\tilde{u}| \otimes \langle x|) \cdot A_n \cdot (|\tilde{u}\rangle \otimes |x\rangle) \\
= \; & (\mathrm{Id}_1 \otimes \langle x|) \cdot (\langle\tilde{u}| \otimes \mathrm{Id}) \cdot A_n \cdot (|\tilde{u}\rangle \otimes \mathrm{Id}) \cdot (\mathrm{Id}_1 \otimes |x\rangle) \\
= \; & \langle x| P_u |x\rangle \\
= \; & \mathrm{Tr}(\langle x| P_u |x\rangle) \\
= \; & \mathrm{Tr}(P_u |x\rangle\langle x|)
\end{aligned}$$

where the last equality follows from the fact that $\mathrm{Tr}(AB) = \mathrm{Tr}(BA)$ for all rectangular matrices (for which the products AB and BA are defined). □

Error reduction for **QMA**

Lemma 22

P_u is positive semi-definite (and hence Hermitian).

Proof: We show that all eigenvalues of P_u are in $[0,1]_{\mathbb{R}}$.

Let λ be an eigenvalue of P_u and let $|x\rangle$ an eigenvector of norm 1 for λ.

We then have

$$\lambda = \langle x| P_u |x\rangle = \mathrm{Tr}(P_u |x\rangle\langle x|) = \mathrm{Prob}[Q_n \text{ accepts } |\tilde{u}\rangle\otimes|x\rangle] \in [0,1]_{\mathbb{R}}. \quad \square$$

By Lemma 21, a quantum proof $|x\rangle$ maximizes the acceptance probability of the input u iff it maximizes the value $\langle x| P_u |x\rangle = \mathrm{Tr}(P_u |x\rangle\langle x|)$.

This is related to the largest eigenvalue of P_u.

For a Hermitian matrix H (all its eigenvalues are real) let $\lambda_{\max}(H)$ be its largest eigenvalue and $\lambda_{\min}(H)$ be its smallest eigenvalue.

Interlude: maximal eigenvalues of Hermitian matrices

Lemma 23

For every Hermitian matrix $H \in \mathbb{C}^{d \times d}$ we have

$$\lambda_{\max}(H) = \max\{\langle x| H |x\rangle : |x\rangle \in \mathbb{C}^d, \langle x|x\rangle = 1\}.$$

Proof: Let $\{|y_1\rangle, \ldots, |y_d\rangle\}$ be an orthonormal basis of eigenvectors of H and let $\lambda_i \in \mathbb{R}$ be the eigenvalue corresponding to $|y_i\rangle$.

Let $|x\rangle = \sum_{i=1}^{d} \alpha_i |y_i\rangle$ be an arbitrary unit vector, where $\sum_{i=1}^{d} |\alpha_i|^2 = 1$.

We obtain $\langle x| H |x\rangle = \sum_{i=1}^{d} |\alpha_i|^2 \lambda_i \leq \sum_{i=1}^{d} |\alpha_i|^2 \lambda_{\max}(H) = \lambda_{\max}(H)$.

On the other hand, if $|y\rangle$ is a unit eigenvector for the eigenvalue $\lambda_{\max}(H)$ then $\langle y| H |y\rangle = \lambda_{\max}(H)$. □

Error reduction for **QMA**

Remarks:

- An analogous argument shows that

$$\lambda_{\min}(H) = \min\{\langle x| H |x\rangle : |x\rangle \in \mathbb{C}^d, \langle x|x\rangle = 1\}. \tag{7}$$

- Implicitly, we have shown that the maximum of the set

$$\{\langle x| H |x\rangle : |x\rangle \in \mathbb{C}^d, \langle x|x\rangle = 1\} \subseteq \mathbb{R}$$

  exists. This can be also shown using the compactness of this set.

Our new quantum circuit will use two different measurements:

- $\{\Pi_0, \Pi_1\}$
- $\{\Theta_0, \Theta_1\}$ where $\Theta_1 := |\tilde{u}\rangle\langle\tilde{u}| \otimes \mathrm{Id}$ and $\Theta_0 = \mathrm{Id}_{2^{n+a(n)+p(n)}} - \Theta_1$.

Consider the algorithm on the next slide, where k is sufficiently large.

Witness-preserving error reduction for **QMA**

The initial quantum state is $|\tilde{u}\rangle \otimes |x\rangle$.

1. $i := 1$; $a_0 := 1$;
2. while $i \leq k - 1$ do
3. apply Q_n to the current quantum state;
4. measure with respect to $\{\Pi_0, \Pi_1\}$;
5. if the outcome is $a \in \{0, 1\}$ then $a_i := a$;
6. $i := i + 1$;
7. apply $Q_n^\dagger$ to the current quantum state;
8. measure with respect to $\{\Theta_0, \Theta_1\}$;
9. if the outcome is $a \in \{0, 1\}$ then $a_i := a$;
10. $i := i + 1$;
11. endwhile
12. accept if and only if the number of i with $a_i = a_{i+1}$ is at least $k/2$.

Witness-preserving error reduction for **QMA**

Lemma 24

If $|x\rangle$ is a unit eigenvector of P_u and $\lambda = \mathrm{Tr}(P_u |x\rangle\langle x|) \in [0,1]_{\mathbb{R}}$ is the corresponding eigenvalue, then $\mathrm{Prob}[a_i = a_{i+1}] = \lambda$ for all $0 \leq i < k$.

Proof: First assume $\lambda = 1$ and consider the 1st iteration of the while-loop:

- $Q_n(|\tilde{u}\rangle \otimes |x\rangle)$ is in the subspace $\mathrm{im}(\Pi_1)$ onto which Π_1 projects.
- Hence, with probability 1 we have $a_1 = 1$ and the state after line 4 is $\Pi_1 Q_n(|\tilde{u}\rangle \otimes |x\rangle) = Q_n(|\tilde{u}\rangle \otimes |x\rangle)$.
- Hence, with probability 1 the state after line 7 is $|\tilde{u}\rangle \otimes |x\rangle$, which belongs to the subspace $\mathrm{im}(\Theta_1)$ onto which Θ_1 projects.
- Hence, with probability 1 we have $a_2 = 1$ and the state at the end of the first iteration is $|\tilde{u}\rangle \otimes |x\rangle$.

We obtain $a_0 = a_1 = \cdots = a_k = 1$ with probability 1.

Witness-preserving error reduction for **QMA**

If $\lambda = 0$, then $Q_n(|\tilde{u}\rangle \otimes |x\rangle)$ belongs to $\mathrm{im}(\Pi_0)$ and we can argue analogously in this case.

We obtain $1 = a_0 \neq a_1 \neq a_2 \neq \cdots \neq a_k$ with probability 1.

Assume that $0 < \lambda < 1$ in the following.

With $\Gamma := \Theta_1 \, Q_n^\dagger \, \Pi_1 \, Q_n$ we have

$$\begin{aligned}
\Gamma(|\tilde{u}\rangle \otimes |x\rangle) &= \Theta_1 \, Q_n^\dagger \, \Pi_1 \, Q_n \, (|\tilde{u}\rangle \otimes |x\rangle) \\
&= (|\tilde{u}\rangle\langle\tilde{u}| \otimes \mathrm{Id}) \, Q_n^\dagger \, \Pi_1 \, Q_n \, (|\tilde{u}\rangle \otimes |x\rangle) \\
&= (|\tilde{u}\rangle \otimes \mathrm{Id}) \, (\langle\tilde{u}| \otimes \mathrm{Id}) \, Q_n^\dagger \, \Pi_1 \, Q_n \, (|\tilde{u}\rangle \otimes |x\rangle) \\
&= (|\tilde{u}\rangle \otimes \mathrm{Id}) \, (\langle\tilde{u}| \otimes \mathrm{Id}) \, Q_n^\dagger \, \Pi_1 \, Q_n \, (|\tilde{u}\rangle \otimes \mathrm{Id}) \, |x\rangle \\
&= (|\tilde{u}\rangle \otimes \mathrm{Id}) \, P_u \, |x\rangle \\
&= (|\tilde{u}\rangle \otimes \mathrm{Id}) \, \lambda \, |x\rangle \\
&= \lambda(|\tilde{u}\rangle \otimes |x\rangle).
\end{aligned}$$

Witness-preserving error reduction for **QMA**

Let us define the two quantum states

$$|y_1\rangle = |\tilde{u}\rangle \otimes |x\rangle \qquad \text{and} \qquad |y_0\rangle = \frac{\Theta_0\, Q_n^\dagger\, \Pi_1\, Q_n\, |y_1\rangle}{\left\| \Theta_0\, Q_n^\dagger\, \Pi_1\, Q_n\, |y_1\rangle \right\|}.$$

Since $|y_1\rangle \in \mathrm{im}(\Theta_1)$ and $|y_0\rangle \in \mathrm{im}(\Theta_0)$, we have $\langle y_0|y_1\rangle = 0$.

Moreover, we have

$$\begin{aligned}
\lambda &= \|\Pi_1 Q_n\, |y_1\rangle\|^2 \\
&= \left\| Q_n^\dagger \Pi_1 Q_n\, |y_1\rangle \right\|^2 \\
&= \left\| \Theta_0 Q_n^\dagger \Pi_1 Q_n\, |y_1\rangle \right\|^2 + \left\| \Theta_1 Q_n^\dagger \Pi_1 Q_n\, |y_1\rangle \right\|^2 \\
&= \left\| \Theta_0 Q_n^\dagger \Pi_1 Q_n\, |y_1\rangle \right\|^2 + \|\Gamma\, |y_1\rangle\|^2 \\
&= \left\| \Theta_0 Q_n^\dagger \Pi_1 Q_n\, |y_1\rangle \right\|^2 + \lambda^2.
\end{aligned}$$

Witness-preserving error reduction for **QMA**

Therefore, we have $\left\| \Theta_0 Q_n^\dagger \Pi_1 Q_n |y_1\rangle \right\|^2 = \lambda - \lambda^2 = \lambda(1-\lambda)$ and

$$|y_0\rangle = \frac{\Theta_0 \, Q_n^\dagger \, \Pi_1 \, Q_n \, |y_1\rangle}{\sqrt{\lambda}\sqrt{1-\lambda}}.$$

The state $|y_1\rangle$ is the initial state and after the first execution of line 3 we obtain the state

$$Q_n |y_1\rangle = \Pi_0 Q_n |y_1\rangle + \Pi_1 Q_n |y_1\rangle .$$

In line 4, the state collapses to

$$|z_1\rangle := \frac{\Pi_1 Q_n |y_1\rangle}{\|\Pi_1 Q_n |y_1\rangle\|} = \frac{\Pi_1 Q_n |y_1\rangle}{\sqrt{\lambda}} \quad \text{with probability } \lambda \text{ and to}$$

$$|z_0\rangle := \frac{\Pi_0 Q_n |y_1\rangle}{\|\Pi_0 Q_n |y_1\rangle\|} = \frac{\Pi_0 Q_n |y_1\rangle}{\sqrt{1-\lambda}} \quad \text{with probability } 1-\lambda$$

Witness-preserving error reduction for **QMA**

Note that $|z_1\rangle \in \mathrm{im}(\Pi_1)$, $|z_0\rangle \in \mathrm{im}(\Pi_0)$ and therefore $\langle z_0|z_1\rangle = 0$.

We have:

$$
\begin{aligned}
Q_n\,|y_1\rangle &= \Pi_0 Q_n\,|y_1\rangle + \Pi_1 Q_n\,|y_1\rangle = \sqrt{\lambda}\,|z_1\rangle + \sqrt{1-\lambda}\,|z_0\rangle\,, \\
Q_n\,|y_0\rangle &= \frac{Q_n\,\Theta_0\,Q_n^\dagger\,\Pi_1\,Q_n\,|y_1\rangle}{\sqrt{\lambda}\sqrt{1-\lambda}} \\
&= \frac{Q_n\,Q_n^\dagger\,\Pi_1\,Q_n\,|y_1\rangle}{\sqrt{\lambda}\sqrt{1-\lambda}} - \frac{Q_n\,\Theta_1\,Q_n^\dagger\,\Pi_1\,Q_n\,|y_1\rangle}{\sqrt{\lambda}\sqrt{1-\lambda}} \\
&= \frac{\Pi_1\,Q_n\,|y_1\rangle}{\sqrt{\lambda}\sqrt{1-\lambda}} - \frac{Q_n\,\Gamma\,|y_1\rangle}{\sqrt{\lambda}\sqrt{1-\lambda}} \\
&= \frac{\sqrt{\lambda}\,|z_1\rangle}{\sqrt{\lambda}\sqrt{1-\lambda}} - \frac{\lambda(\sqrt{\lambda}\,|z_1\rangle + \sqrt{1-\lambda}\,|z_0\rangle)}{\sqrt{\lambda}\sqrt{1-\lambda}} \\
&= -\sqrt{\lambda}\,|z_0\rangle + \sqrt{1-\lambda}\,|z_1\rangle\,.
\end{aligned}
$$

Witness-preserving error reduction for **QMA**

After line 4 the state is either $|z_0\rangle$ or $|z_1\rangle$.

Hence, after line 7 the state is either

$$\begin{aligned} Q_n^\dagger |z_0\rangle &= \Theta_0 Q_n^\dagger |z_0\rangle + \Theta_1 Q_n^\dagger |z_0\rangle \text{ or} \\ Q_n^\dagger |z_1\rangle &= \Theta_0 Q_n^\dagger |z_1\rangle + \Theta_1 Q_n^\dagger |z_1\rangle . \end{aligned}$$

We have:

$$\begin{aligned} \Theta_1 Q_n^\dagger |z_1\rangle &= \frac{\Theta_1 Q_n^\dagger \Pi_1 Q_n |y_1\rangle}{\sqrt{\lambda}} = \frac{\Gamma |y_1\rangle}{\sqrt{\lambda}} = \frac{\lambda |y_1\rangle}{\sqrt{\lambda}} = \sqrt{\lambda} |y_1\rangle \\ \Theta_0 Q_n^\dagger |z_1\rangle &= \frac{\Theta_0 Q_n^\dagger \Pi_1 Q_n |y_1\rangle}{\sqrt{\lambda}} = \sqrt{1-\lambda} |y_0\rangle \end{aligned}$$

Witness-preserving error reduction for **QMA**

$$
\begin{aligned}
\Theta_1 Q_n^\dagger |z_0\rangle &= \frac{\Theta_1 Q_n^\dagger \Pi_0 Q_n |y_1\rangle}{\sqrt{1-\lambda}} \\
&= \frac{\Theta_1 Q_n^\dagger Q_n |y_1\rangle}{\sqrt{1-\lambda}} - \frac{\Theta_1 Q_n^\dagger \Pi_1 Q_n |y_1\rangle}{\sqrt{1-\lambda}} \\
&= \frac{\Theta_1 |y_1\rangle}{\sqrt{1-\lambda}} - \frac{\Gamma |y_1\rangle}{\sqrt{1-\lambda}} \\
&= \frac{|y_1\rangle}{\sqrt{1-\lambda}} - \frac{\lambda |y_1\rangle}{\sqrt{1-\lambda}} = \sqrt{1-\lambda}\,|y_1\rangle
\end{aligned}
$$

$$
\begin{aligned}
\Theta_0 Q_n^\dagger |z_0\rangle &= \frac{\Theta_0 Q_n^\dagger \Pi_0 Q_n |y_1\rangle}{\sqrt{1-\lambda}} \\
&= \frac{\Theta_0 Q_n^\dagger Q_n |y_1\rangle}{\sqrt{1-\lambda}} - \frac{\Theta_0 Q_n^\dagger \Pi_1 Q_n |y_1\rangle}{\sqrt{1-\lambda}} = -\sqrt{\lambda}\,|y_0\rangle
\end{aligned}
$$

Witness-preserving error reduction for **QMA**

Let's sum up everything:

$$Q_n |y_0\rangle = -\sqrt{\lambda}\,|z_0\rangle + \sqrt{1-\lambda}\,|z_1\rangle \quad (8)$$

$$Q_n |y_1\rangle = \sqrt{1-\lambda}\,|z_0\rangle + \sqrt{\lambda}\,|z_1\rangle \quad (9)$$

$$Q_n^\dagger |z_0\rangle = -\sqrt{\lambda}\,|y_0\rangle + \sqrt{1-\lambda}\,|y_1\rangle \quad (10)$$

$$Q_n^\dagger |z_1\rangle = \sqrt{1-\lambda}\,|y_0\rangle + \sqrt{\lambda}\,|y_1\rangle \quad (11)$$

This shows that after the measurement in line 4 (resp., line 8) the quantum state is either $|z_0\rangle$ or $|z_1\rangle$ (resp., either $|y_0\rangle$ or $|y_1\rangle$).

If the state is $|z_0\rangle$ (resp., $|y_0\rangle$) then the measurement outcome is 0 and if the state is $|z_1\rangle$ (resp., $|y_1\rangle$) then the measurement outcome is 1.

The probability that the current measurement outcome is the same as in the previous measurement is exactly λ. □

We now come to main result:

Witness-preserving error reduction for **QMA**

Theorem 25

Let (L_0, L_1) be a promise problem in **QMA** and let $(Q_n)_{n\geq 0}$ be a **P**-uniform quantum circuit family as in the definition of **QMA** on slide 118 ($p(n)$ is the number of qubits of the quantum proof).

For every polynomial $r(n)$ there is a **P**-uniform quantum circuit family $(R_n)_{n\geq 0}$, where R_n works on $n + b(n) + p(n)$ qubits for a polynomial $b(n)$ such that for every $u \in \{0,1\}^n$:

- if $u \in L_1$ then there is a quantum state $|x\rangle \in \mathbb{C}^{2^{p(n)}}$:

$$\text{Prob}[R_n \text{ accepts } |u\, 0^{b(n)}\rangle \otimes |x\rangle] \geq 1 - 2^{-r(n)}.$$

- If $u \in L_0$ then for every quantum state $|x\rangle \in \mathbb{C}^{2^{p(n)}}$:

$$\text{Prob}[R_n \text{ accepts } |u\, 0^{b(n)}\rangle \otimes |x\rangle] \leq 2^{-r(n)}.$$

Witness-preserving error reduction for **QMA**

Proof: The quantum circuits R_n implements the algorithm on slide 126 with $k = \Theta(r(n))$.

By the principle of deferred measurement this is possible.

Let $u \in \{0,1\}^n$ be a classical input.

Case 1. $u \in L_1$. Let $|x\rangle \in \mathbb{C}^{2^{p(n)}}$ be a quantum proof such that

$$\text{Prob}[Q_n \text{ accepts } |u\,0^{a(n)}\rangle \otimes |x\rangle] \geq \frac{2}{3}.$$

By Lemma 21 and 23 we can assume that $|x\rangle$ is a unit eigenvector of P_u.

By Lemma 24, Prob[R_n accepts $|u\,0^{b(n)}\rangle \otimes |x\rangle$] = prob. for getting $\geq k/2$ heads if you toss k times a coin that yields head with prob. $\geq 2/3$.

By the Chernoff bound this probability is $1 - e^{-\Theta(k)} \geq 1 - 2^{-r(n)}$ if you choose the constant factor in $k = \Theta(r(n))$ large enough.

Witness-preserving error reduction for **QMA**

Case 2. $u \in L_0$.

Here, we must be careful: we have to consider all quantum proofs $|x\rangle \in \mathbb{C}^{2^{p(n)}}$ and not only eigenvectors of P_u.

So, let $|x\rangle \in \mathbb{C}^{2^{p(n)}}$ be a quantum state.

Let $|x_1\rangle, \ldots, |x_{2^{p(n)}}\rangle$ be an orthonormal basis of eigenvectors of P_u (exists since P_u is positive semidefinite and hence normal) and let $\lambda_i \leq 1/3$ be the eigenvalue for $|x_i\rangle$.

We can write $|x\rangle$ as $|x\rangle = \sum_{i=1}^{2^{p(n)}} \alpha_i |x_i\rangle$.

For each i with $\lambda_i > 0$ we get states $|y_{0,i}\rangle$, $|y_{1,i}\rangle = |\tilde{u}\rangle \otimes |x_i\rangle$, $|z_{0,i}\rangle$ and $|z_{1,i}\rangle$ such that the equations (8)–(11) hold with the index i added:

Witness-preserving error reduction for **QMA**

$$
\begin{aligned}
Q_n \left|y_{0,i}\right\rangle &= -\sqrt{\lambda_i}\left|z_{0,i}\right\rangle + \sqrt{1-\lambda_i}\left|z_{1,i}\right\rangle && (12)\\
Q_n \left|y_{1,i}\right\rangle &= \sqrt{1-\lambda_i}\left|z_{0,i}\right\rangle + \sqrt{\lambda_i}\left|z_{1,i}\right\rangle && (13)\\
Q_n^\dagger \left|z_{0,i}\right\rangle &= -\sqrt{\lambda_i}\left|y_{0,i}\right\rangle + \sqrt{1-\lambda_i}\left|y_{1,i}\right\rangle && (14)\\
Q_n^\dagger \left|z_{1,i}\right\rangle &= \sqrt{1-\lambda_i}\left|y_{0,i}\right\rangle + \sqrt{\lambda_i}\left|y_{1,i}\right\rangle && (15)
\end{aligned}
$$

We can extend these equations to all i by setting in case $\lambda_i = 0$:

$$\left|y_{0,i}\right\rangle = \left|z_{1,i}\right\rangle = 0, \left|z_{0,i}\right\rangle = Q_n \left|y_{1,i}\right\rangle$$

From the equations (12)–(15) one can deduce that for all $i \neq j$:

$$\langle y_{0,i}|y_{0,j}\rangle = \langle y_{1,i}|y_{1,j}\rangle = \langle z_{0,i}|z_{0,j}\rangle = \langle z_{1,i}|z_{1,j}\rangle = 0.$$

Witness-preserving error reduction for **QMA**

We can now argue as follows: Assume that we start the j-th iteration of the algorithm (where we set a_{2j-1} and a_{2j}) from slide 126 in a state

- $\sum_{i=1}^{2^{p(n)}} \beta_i |y_{0,i}\rangle$ (and hence $a_{2j-2} = 0$) or
- $\sum_{i=1}^{2^{p(n)}} \beta_i |y_{1,i}\rangle$ (and hence $a_{2j-2} = 1$).

Initially, the 2nd case holds with $\beta_i = \alpha_i$ for all i.

After applying Q_n we obtain

1. $\sum_{i=1}^{2^{p(n)}} \beta_i Q_n |y_{0,i}\rangle = \sum_{i=1}^{2^{p(n)}} \beta_i(-\sqrt{\lambda_i} |z_{0,i}\rangle + \sqrt{1-\lambda_i} |z_{1,i}\rangle)$ or
2. $\sum_{i=1}^{2^{p(n)}} \beta_i Q_n |y_{1,i}\rangle = \sum_{i=1}^{2^{p(n)}} \beta_i(\sqrt{1-\lambda_i} |z_{0,i}\rangle + \sqrt{\lambda_i} |z_{1,i}\rangle)$.

Now consider the measurement in line 4.

Witness-preserving error reduction for **QMA**

In the 1st case the quantum state is of the form

a $\sum_{i=1}^{2^{p(n)}} \beta'_i |z_{0,i}\rangle$ $(a_{2j-1} = 0)$ with probability $\sum_{i=1}^{2^{p(n)}} |\beta_i|^2 \lambda_i \leq 1/3$ and

b $\sum_{i=1}^{2^{p(n)}} \beta'_i |z_{1,i}\rangle$ $(a_{2j-1} = 1)$ with probability $\sum_{i=1}^{2^{p(n)}} |\beta_i|^2 (1 - \lambda_i) \geq 2/3$.

Similarly, in the 2nd case the quantum state is of the form

c $\sum_{i=1}^{2^{p(n)}} \beta'_i |z_{0,i}\rangle$ with probability $\sum_{i=1}^{2^{p(n)}} |\beta_i|^2 (1 - \lambda_i) \geq 2/3$ and

d $\sum_{i=1}^{2^{p(n)}} \beta'_i |z_{1,i}\rangle$ with probability $\sum_{i=1}^{2^{p(n)}} |\beta_i|^2 \lambda_i \leq 1/3$.

It follows that $\Pr[a_{2j-2} \neq a_{2j-1}] \geq 2/3$.

Similarly, we obtain $\Pr[a_{2j-1} \neq a_{2j}] \geq 2/3$.

We can now conclude in the same as in the case where $|x\rangle$ is an eigenvector of P_u. □

QMA $\subseteq$ PSPACE

Theorem 26

QMA $\subseteq$ PSPACE

Proof: Fix a promise problem $(L_0, L_1) \in$ **QMA** and an input $u \in \{0,1\}^n$.

Let $(Q_n)_{n \geq 0}$ be a **P**-uniform quantum circuit family as on slide 118 and built up from CNOT-, H- and T-gates.

By Theorem 25, the probabilities $1/3$ and $2/3$ can be replaced by $2^{-p(n)-1}$ and $1 - 2^{-p(n)-1}$ with $p(n) =$ number of qubits in the quantum proof and w.l.o.g. $p(n) \geq 1$.

Important: the number of qubits $p(n)$ in the quantum proof is not increased in Theorem 25.

Let $|x_1\rangle, \ldots, |x_{2^{p(n)}}\rangle$ be an orthonormal basis of $\mathbb{C}^{2^{p(n)}}$ consisting of eigenvectors of the positive semidefinite matrix P_u; see slide 121.

QMA $\subseteq$ PSPACE

By Lemma 21 and Lemma 23 we have:

- if $u \in L_0$ then $\forall i : \mathrm{Tr}(P_u \left|x_i\right\rangle\!\left\langle x_i\right|) \leq 2^{-p(n)-1}$ and
- if $u \in L_1$ then $\exists i : \mathrm{Tr}(P_u \left|x_i\right\rangle\!\left\langle x_i\right|) \geq 1 - 2^{-p(n)-1}$.

We obtain:

$$\begin{aligned}
\mathrm{Tr}(P_u) &= \mathrm{Tr}(P_u \cdot \mathrm{Id}_{2^{p(n)}}) \\
&= \mathrm{Tr}\left(P_u \sum_{i=1}^{2^{p(n)}} \left|x_i\right\rangle\!\left\langle x_i\right| \right) \\
&= \sum_{i=1}^{2^{p(n)}} \mathrm{Tr}(P_u \left|x_i\right\rangle\!\left\langle x_i\right|) \quad \begin{cases} \leq 2^{-p(n)-1} 2^{p(n)} = \frac{1}{2} & \text{if } u \in L_0 \\ \geq 1 - 2^{-p(n)-1} \geq \frac{3}{4} & \text{if } u \in L_1. \end{cases}
\end{aligned}$$

Our new goal is to approximate $\mathrm{Tr}(P_u) \in \mathbb{R}_{\geq 0}$ in **PSPACE** with a precision that allows to distinguish between $1/2$ and $3/4$.

QMA $\subseteq$ PSPACE

Recall that $P_u = \left(\langle \tilde{u}| \otimes \mathrm{Id}_{2^{p(n)}} \right) \cdot Q_n^\dagger \cdot \Pi_1 \cdot Q_n \cdot \left(|\tilde{u}\rangle \otimes \mathrm{Id}_{2^{p(n)}} \right)$.

Moreover, Q_n can be written as $U_1 U_2 \cdots U_{t(n)}$ for a polynomial $t(n)$. Every U_i is either a CNOT, H or T gate.

Let us write $M_0, \ldots, M_{2t(n)+2}$ for the sequence of matrices

$$\left(\langle \tilde{u}| \otimes \mathrm{Id}_{2^{p(n)}} \right), U_1^\dagger, U_2^\dagger, \ldots, U_{t(n)}^\dagger \Pi_1, U_{t(n)}, \ldots, U_2, U_1, \left(|\tilde{u}\rangle \otimes \mathrm{Id}_{2^{p(n)}} \right).$$

Let $A = 2^{p(n)}$, $B = 2^{n+a(n)+p(n)}$, and $s = 2t(n) + 2$.

We have $M_0 \in \mathbb{C}^{A \times B}$, $M_1, \ldots, M_{2t(n)+1} \in \mathbb{C}^{B \times B}$ and $M_0 \in \mathbb{C}^{B \times A}$.

We can compute $\mathrm{Tr}(M_0 M_1 \cdots M_s)$ as follows:

QMA $\subseteq$ PSPACE

$$
\begin{aligned}
& \operatorname{Tr}(M_0 M_1 \cdots M_s) \\
= & \sum_{i_0=0}^{A-1} \langle i_0 | M_0 M_1 \cdots M_s | i_0 \rangle \\
= & \sum_{i_0=0}^{A-1} \langle i_0 | M_0 \cdot \mathrm{Id}_B \cdot M_1 \cdot \mathrm{Id}_B \cdot M_2 \cdots \mathrm{Id}_B \cdot M_s | i_0 \rangle \\
= & \sum_{i_0=0}^{A-1} \langle i_0 | M_0 \cdot \sum_{i_1=0}^{B-1} |i_1\rangle\langle i_1| \cdot M_1 \cdot \sum_{i_2=0}^{B-1} |i_2\rangle\langle i_2| \cdot M_2 \cdots \sum_{i_s=0}^{B-1} |i_s\rangle\langle i_s| \cdot M_s | i_0 \rangle \\
= & \sum_{i_0=0}^{A-1} \sum_{i_1=0}^{B-1} \sum_{i_2=0}^{B-1} \cdots \sum_{i_s=0}^{B-1} \langle i_0 | M_0 | i_1 \rangle \langle i_1 | M_1 | i_2 \rangle \langle i_2 | M_2 \cdots | i_s \rangle \langle i_s | M_s | i_0 \rangle \\
= & \sum_{i_0=0}^{A-1} \sum_{i_1=0}^{B-1} \sum_{i_2=0}^{B-1} \cdots \sum_{i_s=0}^{B-1} (M_0)_{i_0,i_1} (M_1)_{i_1,i_2} (M_2)_{i_2,i_3} \cdots (M_s)_{i_s,i_0} \qquad (16)
\end{aligned}
$$

QMA $\subseteq$ PSPACE

Let $|z|_1 = |\Re(z)| + |\Im(z)|$ be the L1-norm of a complex number z.

Our polynomial space algorithm works as follows:

1. $T := 0$
2. **for all** $i_0 \in [0, A-1]$ **do**
3. **for all** $i_1, \ldots, i_s \in [0, B-1]$ **do**
4. compute approximation α of $(M_0)_{i_0,i_1}(M_1)_{i_1,i_2}(M_2)_{i_2,i_3}\cdots(M_s)_{i_s,i_0}$ such that $|\alpha - (M_0)_{i_0,i_1}(M_1)_{i_1,i_2}(M_2)_{i_2,i_3}\cdots(M_s)_{i_s,i_0}|_1 \leq \frac{1}{16AB^s}$
5. $T := T + \alpha$
6. **end**
7. **end**
8. **if** $\Re(T) \leq \frac{9}{16}$ then **reject**
9. **if** $\Re(T) \geq \frac{11}{16}$ then **accept**

QMA $\subseteq$ PSPACE

Let us first prove the correctness of the algorithm.

The sum in (16) consists of $A \cdot B^s$ products.

For each of these products $(M_0)_{i_0,i_1}(M_1)_{i_1,i_2}(M_2)_{i_2,i_3}\cdots(M_s)_{i_s,i_0}$ the algorithm computes an approximation $\alpha_{i_0,\ldots,i_s}$ such that

$$|\alpha_{i_0,\ldots,i_s} - (M_0)_{i_0,i_1}(M_1)_{i_1,i_2}(M_2)_{i_2,i_3}\cdots(M_s)_{i_s,i_0}|_1 \leq \frac{1}{16AB^s}$$

Hence, the final value T (the sum over all $\alpha_{i_0,\ldots,i_s}$) computed by the algorithm satisfies

$$\begin{aligned}
& |\mathrm{Tr}(M_0M_1\cdots M_s) - T|_1 \\
= & \left|\sum_{i_0=0}^{A-1}\sum_{i_1=0}^{B-1}\sum_{i_2=0}^{B-1}\cdots\sum_{i_s=0}^{B-1}(M_0)_{i_0,i_1}(M_1)_{i_1,i_2}(M_2)_{i_2,i_3}\cdots(M_s)_{i_s,i_0} - T\right|_1 \\
\leq & \sum_{i_0=0}^{A-1}\sum_{i_1=0}^{B-1}\sum_{i_2=0}^{B-1}\cdots\sum_{i_s=0}^{B-1}|(M_0)_{i_0,i_1}(M_1)_{i_1,i_2}(M_2)_{i_2,i_3}\cdots(M_s)_{i_s,i_0} - \alpha_{i_0,\ldots,i_s}|_1
\end{aligned}$$

QMA $\subseteq$ PSPACE

$$\leq \sum_{i=0}^{A-1} \sum_{i_1=0}^{B-1} \sum_{i_2=0}^{B-1} \cdots \sum_{i_s=0}^{B-1} \frac{1}{16AB^s} = \frac{1}{16}.$$

Recall that we only have to distinguish the two cases $\mathrm{Tr}(M_0 \cdots M_s) \leq 1/2$ (where the algorithm should reject) and $\mathrm{Tr}(M_0 \cdots M_s) \geq 3/4$.

If $\mathrm{Tr}(M_0 \cdots M_s) \leq 1/2$ then $\Re(T) \leq 1/2 + 1/16 = 9/16$.

If $\mathrm{Tr}(M_0 \cdots M_s) \geq 3/4$ then $\Re(T) \geq 3/4 - 1/16 = 11/16$.

Therefore, the algorithm gives the correct answer in lines 8 and 9 for all inputs in $L_0 \cup L_1$.

QMA $\subseteq$ PSPACE

It remains to show that the algorithm works in polynomial space.

Recall that $A = 2^{p(n)}$, $B = 2^{n+a(n)+p(n)}$ and $s = 2t(n) + 2$ for polynomials $p(n)$, $a(n)$ and $t(n)$.

Let $r(n) = p(n) + (2t(n) + 2)(n + a(n) + p(n))$ (a polynomial in n).

Then $r(n)$ bits are needed for the program variables $i_0, \ldots, i_s$.

Consider now the product $P := (M_0)_{i_0,i_1}(M_1)_{i_1,i_2}(M_2)_{i_2,i_3} \cdots (M_s)_{i_s,i_0}$

In order to get an $\frac{1}{16AB^s}$-approximation of P in the L1-norm, it suffices to compute $\frac{1}{32AB^s}$-approximations of $\Re(P)$ and $\Im(P)$.

Note that $\dfrac{1}{32AB^s} = 2^{-r(n)-5}$.

QMA $\subseteq$ PSPACE

We only consider $\Re(P)$. The approximation for $\Im(P)$ can be computed in the same way.

In order to get a $2^{-r(n)-5}$-approximation of $\Re(P)$, it suffices to compute the first $r(n)+5$ bits after the comma of $\Re(P)$.

From the indices $i_0, \ldots, i_s$ one can symbolically compute in polynomial time the matrix entries $(M_0)_{i_0,i_1}$, $(M_1)_{i_1,i_2}$, $(M_2)_{i_2,i_3}, \ldots, (M_s)_{i_s,i_0}$.

Each of them is either 0, 1, $e^{i\pi/4}$ or $\frac{\pm 1}{\sqrt{2}}$.

If a 0 appears then $P = 0$.

If no 0 appears, then one can compute $0 \leq a, b \leq s \leq \mathcal{O}(t(n))$ in polynomial time such that

$$P = \frac{\pm 1}{2^{a/2}} \cdot e^{b\pi i/4}$$

QMA $\subseteq$ PSPACE

Hence $\Re(P)$ can be symbolically computed in one of the forms

$$\frac{\pm 1}{2^c} \quad \text{or} \quad \frac{\pm\sqrt{2}}{2^c}$$

where $0 \leq c \leq \mathcal{O}(t(n))$ can be computed in polynomial time.

In order to compute the first $r(n) + 5$ bits after the comma of $\Re(P)$, one only needs to compute $r(n) + 5$ bits of $\sqrt{2}$ after the comma and then make a right shift for c positions.

Computing polynomially many bits of an algebraic number like $\sqrt{2}$ is know to be possible in polynomial time; see e.g. this paper

Finally, notice that since every α computed in line 4 of the algorithm has bit length $r(n) + 5$, the bit length of the program variable T is bounded by $2r(n) + 5$ (again a polynomial).

QMA $\subseteq$ PP

One can improve the upper bound for **QMA** from **PSPACE** to **PP**.

A promise problem (L_0, L_1) belongs to **PP** if there is a PTM M and a polynomial $r(n)$ such that the following hold for every $u \in \{0,1\}^n$:

- If $u \in L_1$ then $\langle u, v \rangle \in L(M)$ for $\geq \frac{1}{2} \cdot 2^{r(n)}$ many $v \in \{0,1\}^{r(n)}$.
- If $u \in L_0$ then $\langle u, v \rangle \in L(M)$ for $< \frac{1}{2} \cdot 2^{r(n)}$ many $v \in \{0,1\}^{r(n)}$.

Then one has **QMA** $\subseteq$ **PP** $\subseteq$ **PSPACE**.

The local Hamiltonian problem

The dynamics of the state $|\Psi(t)\rangle$ of a quantum system is determined by Schrödinger's equation:

$$i\frac{d\,|\Psi(t)\rangle}{dt} = H\,|\Psi(t)\rangle\,.$$

Here, H is a Hermitian operator, called the Hamiltonian of the quantum system.

Solution: $|\Psi(t)\rangle = e^{-iHt}\,|\Psi(0)\rangle$. Here, the exponential e^{iHt} is a unitary operator.

The eigenvalues (they are real numbers) of H are the energy levels of the quantum system.

The smallest eigenvalue $\lambda_{\min}(H)$ is the ground energy of the system; its eigenvectors are the ground states.

The local Hamiltonian problem

Assume we have a quantum system consisting of n qubits. Let $k \leq n$.

A Hermitian operator $H \in \mathbb{C}^{2^n \times 2^n}$ is a k-qubit Hamiltonian if it acts only on k of the n qubits.

More formally: there are $i_1, \ldots, i_k \in \{1, \ldots, n\}$ with $i_1 < i_2 < \cdots < i_k$ and a Hermitian matrix $L \in \mathbb{C}^{2^k \times 2^k}$ such that for all $u, v \in \{0,1\}^n$:

$$H_{u,v} = \begin{cases} L_{u[i_1]\cdots u[i_k], v[i_1]\cdots v[i_k]} & \text{if } u[j] = v[j] \text{ for all } j \notin \{i_1, \ldots, i_k\} \\ 0 & \text{else} \end{cases}$$

If $\{i_1, \ldots, i_k\} = \{i+1, \ldots, i+k\}$ then $H = \mathrm{Id}_2^{\otimes i} \otimes L \otimes \mathrm{Id}_2^{\otimes (n-i-k)}$.

A k-local Hamiltonian is a sum $H = H_1 + H_2 + \cdots + H_m$, where every H_i is a k-qubit Hamiltonian.

The local Hamiltonian problem

Many real Hamiltonian are k-local for small values of k.

Example: The Hamiltonian for the Ising model for a chain of length N is

$$H = -\sum_{i=1}^{N-1} \mathrm{Id}_2^{\otimes(i-1)} \otimes X \otimes X \otimes \mathrm{Id}_2^{\otimes(N-i-1)} + \sum_{i=1}^{N} \mathrm{Id}_2^{\otimes(i-1)} \otimes Z \otimes \mathrm{Id}_2^{\otimes(N-i)}$$

for the Pauli-X and Pauli-Z matrices.

The k-local Hamiltonian problem:

- **Input:** A k-local Hamiltonian H that is given as a sum of k-qubit Hamiltonians and two natural numbers $a > b$ in unary notation (a is given by a string of length a and similarly for b).
- **Output:** YES if $\lambda_{\min}(H) \leq 1/a$ and NO if $\lambda_{\min}(H) \geq 1/b$.

The local Hamiltonian problem

Theorem 27 (Kempe, Kitaev, Regev 2006)

The 2-local Hamiltonian problem is **QMA**-complete.

Remarks:

- Kitaev proved in 1999 that the 5-local Hamiltonian problem is **QMA**-complete; we will only proof this result.
- Aharonov, Gottesman, Irani and Kempe proved **QMA**-completeness for local Hamiltonians with a 1-dimensional structure (similar to the Ising model).
- The 2-local Hamiltonian problem is in **P**.
- Theorem 27 is often called the quantum analog of the Cook-Levin theorem.

The local Hamiltonian problem

The Cook-Levin theorem states that 3SAT is **NP**-complete, where 3SAT asks whether a given 3CNF-formula

$$\bigwedge_{i=1}^{k} (y_{i,1} \vee y_{i,2} \vee y_{i,3})$$

is satisfiable. Every $y_{i,j}$ is a boolean variable or a negated boolean variable:

The analogy is the following:

2-local Hamiltonian	3SAT
$\sum_{i=1}^{N}$	$\bigwedge_{i=1}^{N}$
2-qubit Hamiltonian	clause $y_{i,1} \wedge y_{i,2} \wedge y_{i,3}$
qubits	classical bits

The local Hamiltonian problem is in **QMA**

We now show that the k-local Hamiltonian problem belongs to **QMA** for every constant k.

Consider a k-local Hamiltonian $H = H_1 + H_2 + \cdots + H_m$ acting on n qubits, where every H_i is a k-qubit Hamiltonian that is defined via a Hermitian matrix $L_i \in \mathbb{C}^{2^k \times 2^k}$.

We restrict to the case where all eigenvalues of all L_i are between 0 and 1.

For every $1 \leq i \leq m$ define

$$H_i' = L_i \otimes \mathrm{Id}^{\otimes(n-k)}.$$

Then there exists a sequence of SWAP-operations S_i (a unitary matrix) such that $H_i = S_i^{-1} H_i' S_i$.

The local Hamiltonian problem is in **QMA**

Every L_i can be written in its spectral decomposition (see slide 41):

$$L_i = \sum_{j=1}^{2^k} \alpha_{i,j} \left|y_{i,j}\right\rangle \left\langle y_{i,j}\right| ,$$

where the $\left|y_{i,j}\right\rangle$ are pairwise orthogonal unit eigenvectors of L_i and the $\alpha_{i,j} \in [0,1]_{\mathbb{R}}$ $(1 \leq j \leq 2^k)$ are the corresponding eigenvalues of L_i.

Let $\left|x\right\rangle$ be an arbitrary quantum state on n qubits.

For every $1 \leq i \leq m$ there are quantum states $\left|z_{i,j}\right\rangle$ on $n-k$ qubits and complex numbers $\beta_{i,j}$ such

$$S_i \left|x\right\rangle = \sum_{j=1}^{2^k} \beta_{i,j} \left|y_{i,j}\right\rangle \otimes \left|z_{i,j}\right\rangle .$$

The local Hamiltonian problem is in **QMA**

Since $\langle x|x\rangle = 1$ and the $|y_{i,j}\rangle$ $(1 \leq j \leq 2^k)$ are orthonormal, we have

$$\sum_{j=1}^{2^k} |\beta_{i,j}|^2 = 1.$$

We now add an additional ancilla qubit that is initially set to 0 and define for every $1 \leq i \leq m$ the following unitary operation U_i on $n+1$ qubits:

$$U_i : |y_{i,j}\rangle \otimes |z\rangle \otimes |0\rangle \mapsto |y_{i,j}\rangle \otimes |z\rangle \otimes (\sqrt{\alpha_{i,j}}\,|0\rangle + \sqrt{1-\alpha_{i,j}}\,|1\rangle).$$

where $|z\rangle$ is an arbitrary quantum state on $n-k$ qubits.

U_i rotates the last qubit where the angle of the rotation depends on the first k qubits.

The local Hamiltonian problem is in **QMA**

We then have

$$\begin{aligned} & U_i(S_i \otimes \mathrm{Id}_2)(|x\rangle \otimes |0\rangle) \\ = \quad & U_i \sum_{j=1}^{2^k} \beta_{i,j} |y_{i,j}\rangle \otimes |z_{i,j}\rangle \otimes |0\rangle \\ = \quad & \sum_{j=1}^{2^k} \beta_{i,j} |y_{i,j}\rangle \otimes |z_{i,j}\rangle \otimes (\sqrt{\alpha_{i,j}} |0\rangle + \sqrt{1-\alpha_{i,j}} |1\rangle). \end{aligned}$$

Let $P_{i,x}$ be the probability of getting outcome 1 when measuring the last qubit in $U_i(S_i \otimes \mathrm{Id}_2)(|x\rangle \otimes |0\rangle)$. We get

$$P_{i,x} = \sum_{j=1}^{2^k} |\beta_{i,j}|^2 (1-\alpha_{i,j}) = 1 - \sum_{j=1}^{2^k} |\beta_{i,j}|^2 \alpha_{i,j}.$$

The local Hamiltonian problem is in **QMA**

Note that

$$
\begin{aligned}
\langle x| H |x\rangle &= \sum_{i=1}^{m} \langle x| S_i^\dagger H_i' S_i |x\rangle \\
&= \sum_{i=1}^{m} \sum_{j=1}^{2^k} \sum_{\ell=1}^{2^k} |\beta_{i,j}|^2 (\langle y_{i,j}| \otimes \langle z_{i,j}|) H_i' (|y_{i,\ell}\rangle \otimes |z_{i,\ell}\rangle) \\
&= \sum_{i=1}^{m} \sum_{j=1}^{2^k} \sum_{\ell=1}^{2^k} |\beta_{i,j}|^2 \langle y_{i,j}| L_i |y_{i,\ell}\rangle \langle z_{i,j}|z_{i,\ell}\rangle \\
&= \sum_{i=1}^{m} \sum_{j=1}^{2^k} \sum_{\ell=1}^{2^k} |\beta_{i,j}|^2 \alpha_{i,\ell} \langle y_{i,j}|y_{i,\ell}\rangle \langle z_{i,j}|z_{i,\ell}\rangle \\
&= \sum_{i=1}^{m} \sum_{j=1}^{2^k} |\beta_{i,j}|^2 \alpha_{i,j}.
\end{aligned}
$$

The local Hamiltonian problem is in **QMA**

We obtain

$$\frac{1}{m}\sum_{i=1}^{m} P_{i,x} = 1 - \frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{2^k} |\beta_{i,j}|^2 \alpha_{i,j} = 1 - \frac{1}{m}\langle x| H |x\rangle .$$

Consider now the following "quantum circuit", where $|x\rangle$ is an n-qubit quantum proof.

1. Randomly choose a number $i \in \{1, 2, \ldots, m\}$.
2. Apply to $|x\rangle \otimes |0\rangle$ the unitary transformation $U_i(S_i \otimes \mathrm{Id}_2)$.
3. Measure the last qubit.

By the above calculation, the measurement produces outcome 1 with probability $1 - \langle x| H |x\rangle / m$.

The local Hamiltonian problem is in **QMA**

Note:

- The probabilistic choice in the 1st step can be simulated by applying Hadamard gates to $\log_2 m$ additional ancilla qubits that are initially set to $|0\rangle$.
- S_i is a sequence of SWAP-gates and U_i is a unitary operation acting on $k+1$ (a constant) qubits.

Assume now that we want to distinguish the cases $\lambda_{min}(H) \leq 1/a$ and $\lambda_{min}(H) \geq 1/b$ for $a > b$ given in unary notation.

Case 1: $\lambda_{min}(H) \leq 1/a$.

By (7) there is a quantum proof $|x\rangle$ such that $\langle x| H |x\rangle \leq 1/a$.

Hence, our quantum circuit accepts with probability $1 - \langle x| H |x\rangle / m \geq 1 - 1/ma$.

The local Hamiltonian problem is in **QMA**

Case 2: $\lambda_{min}(H) \geq 1/b$.

By (7), for every quantum proof $|x\rangle$ we have $\langle x| H |x\rangle \geq 1/b$.

Hence, our quantum circuit accepts for every quantum proof $|x\rangle$ with probability $1 - \langle x| H |x\rangle / m \leq 1 - 1/mb$.

The gap between the two acceptance probabilities from Cases 1 and 2 is $\frac{1}{m}(\frac{1}{b} - \frac{1}{a})$, which is large enough.

For the proof of the **QMA**-hardness see the paper by Dorit Aharonov and Tomer Naveh.

Quantum Fourier transformation (QFT)

The d-dimensional QFT is the quantum transformation defined by

$$\forall a \in \{0, \ldots, a-1\} : |a\rangle \mapsto \frac{1}{\sqrt{d}} \sum_{b=0}^{d-1} \omega^{ab} \cdot |b\rangle$$

where $\omega = e^{2\pi i/d}$ is a primitive root of unity of order d.

It satisfies $\omega^d = 1$ and $\omega^k \neq 1$ for $1 \leq k \leq d-1$.

The corresponding $(d \times d)$-matrix is the matrix of the discrete Fourier transformation:

$$\mathrm{DFT}_d = \frac{1}{\sqrt{d}} \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega^1 & \omega^2 & \cdots & \omega^{d-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(d-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \omega^{d-1} & \omega^{2(d-1)} & \cdots & \omega^{(d-1)^2} \end{pmatrix}$$

Quantum Fourier transformation (QFT)

Lemma 28

DFT_d is unitary.

This will follow as a corollary soon.

Assume in the following that $d = 2^n$ and identify the basis state $|a\rangle$ ($0 \leq a \leq 2^n - 1$) with the n-bit binary representation $|a_1 a_2 \cdots a_n\rangle$ of a.

In other words: $a = \sum_{j=1}^{n} a_j 2^{n-j}$.

In the following, we make use of the rational numbers

$$0.a_j a_{j+1} \cdots a_n = \frac{a_j}{2} + \frac{a_{j+1}}{4} + \cdots + \frac{a_n}{2^{n-j+1}} = \frac{a}{2^j}.$$

Quantum Fourier transformation (QFT)

Lemma 29

DFT_{2^n} maps the computational basis state $|a\rangle = |a_1 a_2 \cdots a_n\rangle$ $(0 \leq a \leq 2^n - 1)$ to

$$\frac{1}{2^{n/2}} \bigotimes_{j=n}^{1} \left(|0\rangle + e^{2\pi i (0.a_j a_{j+1} \cdots a_n)} |1\rangle \right) \tag{17}$$

Proof: We have

$$\begin{aligned} \mathrm{DFT}_{2^n} |a\rangle &= \frac{1}{\sqrt{2^n}} \sum_{b=0}^{d-1} \omega^{ab} \cdot |b\rangle \\ &= \frac{1}{2^{n/2}} \sum_{b_1=0}^{1} \cdots \sum_{b_n=0}^{1} \exp(2\pi i \, a \sum_{j=1}^{n} b_j 2^{n-j}/2^n) \, |b_1 b_2 \cdots b_n\rangle \end{aligned}$$

Quantum Fourier transformation (QFT)

$$
\begin{aligned}
&= \frac{1}{2^{n/2}} \sum_{b_1=0}^{1} \cdots \sum_{b_n=0}^{1} \exp(2\pi i\, a \sum_{j=1}^{n} b_j 2^{-j}) \,|b_1 b_2 \cdots b_n\rangle \\
&= \frac{1}{2^{n/2}} \sum_{b_1=0}^{1} \cdots \sum_{b_n=0}^{1} \left(\prod_{j=1}^{n} \exp(2\pi i\, a\, b_j 2^{-j}) \right) |b_1\rangle \otimes |b_2\rangle \otimes \cdots |b_n\rangle \\
&= \frac{1}{2^{n/2}} \sum_{b_1=0}^{1} \cdots \sum_{b_n=0}^{1} \bigotimes_{j=1}^{n} \exp(2\pi i\, a\, b_j 2^{-j}) \,|b_j\rangle \\
&= \frac{1}{2^{n/2}} \bigotimes_{j=1}^{n} \sum_{b_j=0}^{1} \exp(2\pi i\, a\, b_j 2^{-j}) \,|b_j\rangle \\
&= \frac{1}{2^{n/2}} \bigotimes_{j=1}^{n} (|0\rangle + \exp(2\pi i\, a\, 2^{-j}) \,|1\rangle) \overset{(*)}{=} (17)
\end{aligned}
$$

For $(*)$ note that $\exp(2\pi i\, a_1 \cdots a_{k-1}.a_k \cdots a_n) = \exp(2\pi i\, 0.a_k \cdots a_n)$. □

Quantum Fourier transformation (QFT)

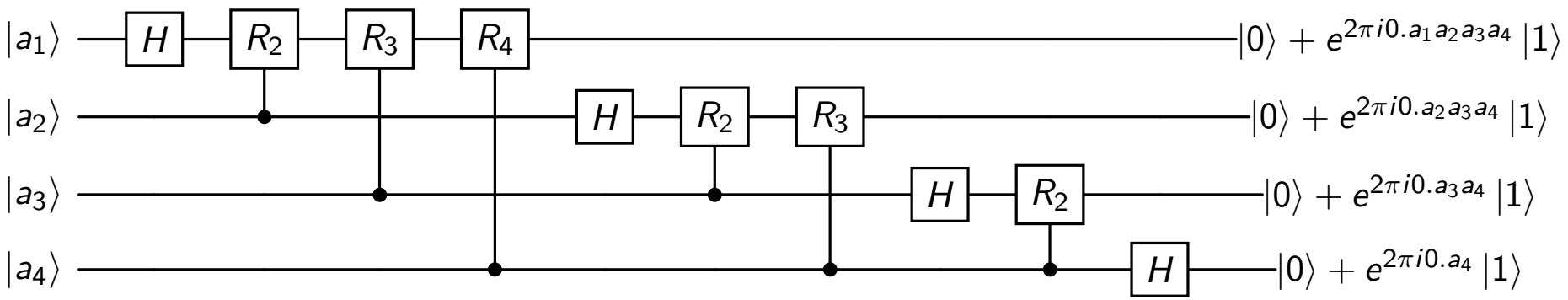
Using Lemma 29 we can obtain a quantum circuit for DFT_{2^n}.

Define the 1-qubit quantum gate

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

We will use the controlled 2-qubit gates $\mathrm{C}_{i,j}(R_k)$, see slide 74.

Then we obtain the following quantum circuit for DFT_{2^n} (here for $n = 4$):



At the end one has to do SWAP-operations in order to exchange the qubits i and $n+1-i$ for $1 \leq i \leq n$.

Quantum Fourier transformation (QFT)

To see the correctness of the above circuit, notice that:

- the first H-gate transforms $|a_1 a_2 a_3 a_4\rangle$ to

$$\frac{1}{2^{1/2}}(|0\rangle + e^{2\pi i\, 0.a_1}\,|1\rangle) \otimes |a_2 a_2 a_3\rangle \tag{18}$$

  Note that $e^{2\pi i\, 0.a_1} = 1$ if $a_1 = 0$ and $e^{2\pi i 0.a_1} = e^{\pi i} = -1$ if $a_1 = 1$.

- The first controlled-R2 gate transforms (18) into

$$\begin{aligned} & \frac{1}{2^{1/2}}(|0\rangle + e^{2\pi i\, 0.0a_2} \cdot e^{2\pi i\, 0.a_1}\,|1\rangle) \otimes |a_2 a_2 a_3\rangle \\ = \quad & \frac{1}{2^{1/2}}(|0\rangle + e^{2\pi i\, 0.a_1 a_2}\,|1\rangle) \otimes |a_2 a_2 a_3\rangle \end{aligned}$$

- We can continue similarly with the other gates.

Quantum Fourier transformation (QFT)

The number of gates for DFT_{2^n} is $\sum_{i=1}^{n} i = n(n+1)/2 = \Theta(n^2)$.

An a classical computer, the best algorithm for computing $\mathrm{DFT}_N |x\rangle$ for a given vector $|x\rangle$ is the fast Fourier transformation (FFT); see my lecture Algorithms II.

Its running time is $\Theta(N \log N)$, and hence $\Theta(2^n \cdot n)$ for $N = 2^n$.

This sounds like QFT is exponentially faster than FFT!

But there are two problems with QFT:

- We cannot prepare an arbitrary quantum state $|x\rangle$
- We cannot obtain the amplitudes of the quantum state $\mathrm{DFT}_{2^n} |x\rangle$.