Quantum Algorithms
Lecture 15
Definition of Quantum Computation II

Zhejiang University

A universal quantum circuit

Quantum advantage

The second of the examples mentioned in the Introduction was simulation of a quantum mechanical system. This is a vaguely posed problem since the choice of particular systems and distinguishing “essential” degrees of freedom play an important role. The problem has been actually solved in several settings.

Simulation of quantum system

With high confidence, we may claim that every physical quantum system can be efficiently simulated on a quantum computer, but we can never prove this statement. The situation resembles that of Turing's thesis:

Any algorithm can be realized by a Turing machine. Cannot be proved, but it is supported by empirical evidence.

Quantum simulation

a 0	a 00	a 000
b 1	b 01	b 001
	c 10	c 010
	d 11	d 011
		e 100
		f 101
		g 110
		h 111

Molecule of Penicillin

2^{286}

source: https://pxhere.com/en/photo/181611

Universality

Recall that the validity of Turing's thesis is partially justified by the existence of the universal Turing machine. In this vein, we may examine universality of our quantum computation model by purely mathematical means. Let us try to simulate many circuits by one.

General quantum circuits

We will not limit the type of gates we use to any particular basis. General quantum circuits have manageable description if the gates are specified as matrices with entries given by binary fractions to certain precision δ_1. Then the inaccuracy of an r-qubit gate (in the operator norm) does not exceed $\delta = M\delta_1$, where $M = 2^r$ is the size of the matrix.

Simulating all circuits

Suppose we have a description Z of a quantum circuit of size $\leq L$ and precision δ. Each gate of the circuit acts on at most r qubits, so that the total length of the description does not exceed $poly(L2^r \log(1/\delta))$. The operator realized by the circuit will be denoted by $Op(Z)$. We will try to simulate all circuits with the given parameters L, r, δ.

Reducing to standard basis

Using the algorithm from the proof of Theorem 8.1, we reduce the problem to the case $r = 2$. Then we apply Theorem 8.3. Thus we can realize each operator in Z by a circuit of size $poly(2^r \log(1/\delta))$ over the standard basis using $O(r)$ ancillas.

T8.1: The basis consisting of all one-qubit and two-qubit unitary operators allows the realization of an arbitrary unitary operator.

T8.3: Any unitary operator U on a fixed number of qubits can be realized with precision δ by a $poly(\log(1/\delta))$-size, $poly(\log\log(1/\delta))$-depth circuit over the standard basis, using ancillas. There is a polynomial algorithm that constructs this circuit on the description of U.

$$Q = \{H, K, K^{-1}, \Lambda(\sigma^x), \Lambda^2(\sigma^x)\}$$

Circuit description

This yields (a description of) a circuit $R(Z)$ over the standard basis, which has size $S = poly(L2^r \log(1/\delta))$, operates on $N = L + O(r)$ qubits, and approximates $Op(Z)$ with precision $O(L\delta)$. The transformation $Z \to R(Z)$ is performed by a Boolean circuit of size $poly(L2^r \log(1/\delta))$. Hence simulating a general circuit is not much harder than simulating circuits over the standard basis.

Universal quantum circuit

The result is as follows. There is a universal quantum circuit U of size $poly(L2^r \log(1/\delta))$ that simulates the work of an arbitrary quantum circuit in the following way: for any circuit description Z and input vector $|\xi\rangle$, U satisfies the condition

$$\left\| U(|Z\rangle \otimes |\xi\rangle \otimes |0^k\rangle) - |Z\rangle \otimes (Op(Z)|\xi\rangle) \otimes |0^k\rangle \right\| = O(L\delta)$$

That is, U works as a “programmable quantum computer”, with Z being the “program”.

Control operators

The qubits of the circuit U include N "controlled" qubits that correspond to the qubits of $R(Z)$. Another subset of qubits holds $|Z\rangle$. There is also a number of auxiliary qubits, some of which are called "controlling". The key component of the circuit U is a circuit V, the product of the operators $V_j = \Lambda(X)[j, k_j]$ (or $V_j = \Lambda(X)[j, k_j, l_j]$, or $V_j = \Lambda(X)[j, k_j, l_j, m_j]$), with X from the standard basis, applied to each one (or pair, or triple) of the controlled qubits in an arbitrary order.

Controlling qubits

The controlling qubits j are all different. If we set one controlling qubit to 1 and all the others to 0, then the circuit V realizes an operator of the form $X[k]$ (or $X[k,l]$, or $X[k,l,m]$) on the controlled qubits. Hence the composition of S copies of V with different controlling qubits can simulate an arbitrary circuit of size S over the standard basis, provided that the controlling qubits are set appropriately.

Controlling qubits

To set the controlling qubits, we need to compute $R(Z)$ by a reversible circuit (with garbage) and arrange the output in a certain way. This computation should be reversed at the end.

Overall description

Our Universal Quantum Circuit (QC) U receives:

- a state $|Z\rangle$, where Z is a description of the circuit, that we are simulating;
- input vector $|\xi\rangle$, on which circuit Z has to be launched.

So, we can assume, that U just simulates $Z|\xi\rangle$. Universal circuit U is designed for specific parameters:

- L – maximal length (number of operations) of circuit Z;
- δ – precision of each operation of Z;
- r – maximal number of qubits on which operators of Z can act.

Effect of parameters

The parameters L, r, δ affect the initial step that approximates Z over a standard basis. This is the classical step and is described in Chapter 8. So we classically translate Z into approximation $R(Z)$, with the following result: size $S = poly(L2^r \log(1/\delta))$, operates on $N = L + O(r)$ qubits, and approximates $Op(Z)$ with precision $O(L\delta)$. So these maximal values can be fixed for our Universal QC.

Next step

Our Universal QC can be considered as consisting of two parts:

- Classical part, that translated $|Z\rangle$ into $R(Z)$ according to Chapter 8.
- Quantum part, that launches operations of $R(Z)$ on a provided input vector $|\xi\rangle$.

Classical part can be implemented as quantum circuit in reversible way (also described in Chapter 7).

Quantum step

We have a finite basis of quantum operators and $R(Z)$ is sequence of these operators. We have N qubits, on which $|\xi\rangle$ is located. Suppose that maximal number of operations of our circuit is S_{max}. The main part of Universal QC will consist of control qubits and N operational qubits. Control will be our programmable part – it will affect which operations on our N qubits should be performed.

Quantum step

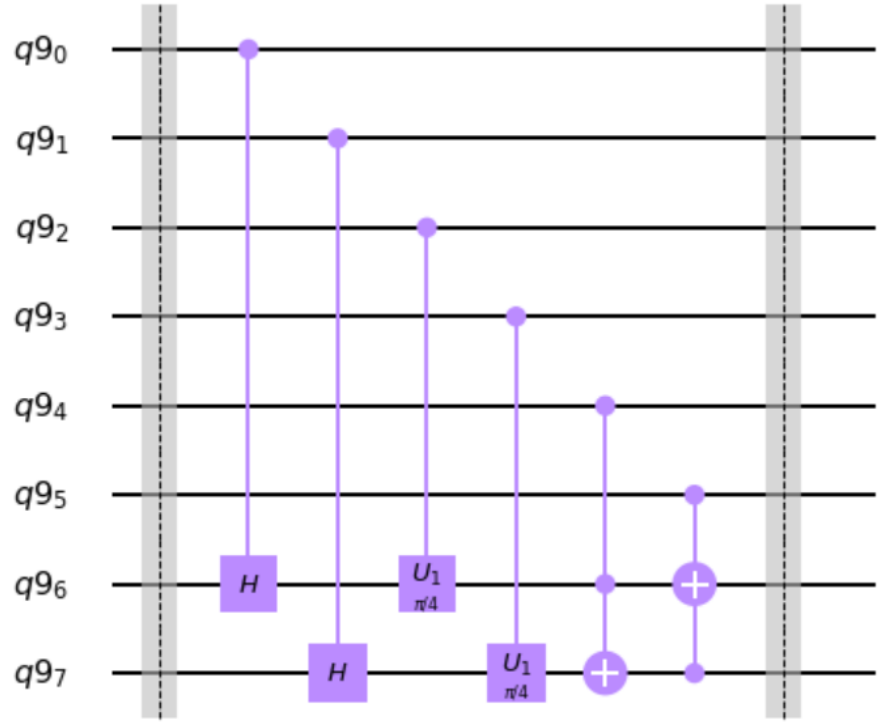
Universal QC operates on a standard basis $Q = \{H, K, K^{-1}, \Lambda(\sigma^x), \Lambda^2(\sigma^x)\}$. Therefore, our circuit on N operational qubits will consist of S_{max} blocks of these operations on arbitrary qubits. Each block will have H, K, K^{-1} for each of N qubits, $\Lambda(\sigma^x)$ for each pair of qubits, and $\Lambda^2(\sigma^x)$ for each combination of 3 qubits. Each such qubit, pair, and combination of 3 qubits will have a corresponding control qubit, and control qubits will determine which one operation in the block will be performed. This is simulation of one computational step. And we will have up to S_{max} such computational steps.

Let's simplify with example

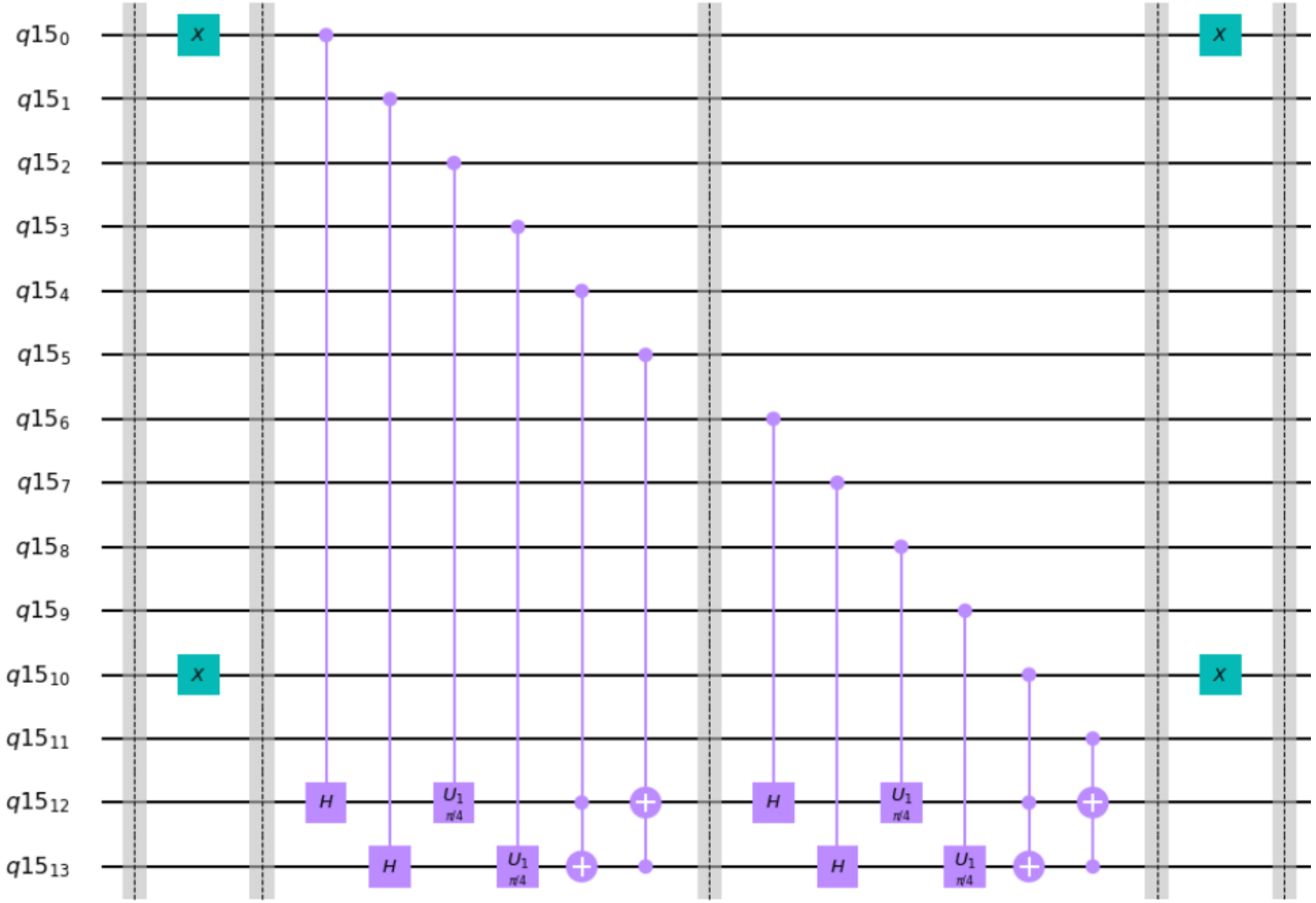
Suppose that we picked smallest universal basis $\{H, T, CNOT\}$ and that $N = 2$. Then one block will have H for both qubits, T for 2 qubits, $CNOT$ from qubit 0 to qubit 1, and $CNOT$ from qubit 1 to qubit 0. In total we have 6 different possibilities for one operation on our circuit. For each of this 6 possibilities we have 1 control qubit – if it is initialized in state $|1\rangle$, operation will be performed. So the values of control qubits encode operations on our circuit.

Example for one block

Here qubits with indexes 0-5 are control qubits, 6 and 7 are our circuit operational qubits. U_1 denotes T operation. Corresponding qubit in state $|1\rangle$ (with index 0-5) will affect which operation will be performed.

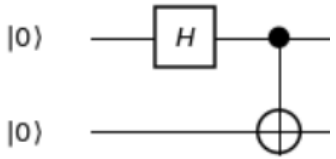


Example for two operations



X changes qubit to state $|1\rangle$,
at the end we reverse it.

Result:



Universality of the circuit

Blocks are designed in a way that they can accept arbitrary circuit $R(Z)$ for the simulation. Here just sketch for universal classical step, that precomputes values of control qubits after receiving description of $R(Z)$.

In our example description has the following alphabet: $\{H[1], H[2], T[1], T[2], CNOT[1,2], CNOT[2,1]\}$, and classical algorithm can process the input description:

```
Set all classical_bits[] = 0;
Set control_num=0;
For each symbol of description:
    if symbol='H[1]', then classical_bits[control_num]=1
    if symbol='H[2]', then classical_bits[control_num+1]=1
    if symbol='T[1]', then classical_bits[control_num+2]=1
    if symbol='T[2]', then classical_bits[control_num+3]=1
    if symbol='CNOT[1,2]', then classical_bits[control_num+4]=1
    if symbol='CNOT[2,1]', then classical_bits[control_num+5]=1
    control_num=control_num+6
```

After processing description, resulting classical bits can be passed to our quantum circuit as basis states.

Summary

This part of the circuit looks classically programmable, where we just classically set controlling qubits to basis states. Depending on this, corresponding sequence of quantum gates is being applied to our operational qubits.

Technical notes:

- Classical reversible circuit can translate initial circuit to circuit over standard basis.
- As you can see, there are many control qubits, in simplified example 6 per operation in simulation.

Quantum algorithms and the class BQP

Nonuniform quantum computation

Up until now we have been studying nonuniform quantum computation (i.e., computation of Boolean functions). Algorithms compute functions on words of arbitrary length. A definition of a quantum algorithm can be given using quantum circuits that have been already introduced. Roughly speaking, a classical Turing machine builds a quantum circuit that computes the value of the function on one or many inputs.

“nonuniform” - a circuit, is used to perform computation with input strings of each individual length.

Definitions

Actually, there are several equivalent definitions, the following being the standard one. Let F: $B^* \rightarrow B^*$ be a function such that the length of the output is polynomial in the length of the input. It is composed of a sequence of Boolean functions F_n: $B^n \rightarrow B^{m(n)}$ (restrictions of F to inputs of length $n = 0,1,2, \ldots$).

Uniform computation

A quantum algorithm for the computation of F is a uniform sequence of quantum circuits that compute F_n. Uniform means that the description Z_n of the corresponding circuit is constructed by a classical Turing machine which takes n as the input. We will say that the quantum algorithm computes F in time $T(n)$ if building the circuit takes at most $T(n)$ steps. The size of the circuit L is obviously not greater than $T(n)$.

Choice of basis

A subtle point in this definition is what basis to use. It is safe to stick to the standard basis. Alternatively, the basis may consist of all unitary operators. In this case, each r-qubit gate should be specified as a list of all its matrix elements with precision $c2^{-r}L^{-1}$, so that the precision of the matrix (in the operator norm) is cL^{-1}, where c is a small constant.

Error probability

If $\varepsilon + 2c < 1/2$, then the approximate circuit works fine. Using the algorithm of Theorems 8.1 and 8.3, this circuit can be transformed to an equivalent circuit of size $poly(T(n))$ over the standard basis (note that $T(n)$ includes the factor 2^r). The converse is obvious.

Suppose that each gate of the circuit U_k is approximated by $\widetilde{U}_k$ with precision δ. The resulting circuit $\widetilde{U} = \widetilde{U}_L \cdots \widetilde{U}_2 \widetilde{U}_1$ satisfies the inequality

$$\sum_z |\langle F(x), z|U|x, 0^{N-n}\rangle|^2 \geq 1 - \varepsilon$$

with ε replaced by $\tilde{\varepsilon} = \varepsilon + 2L\delta$.

Noncomputable – remark

The use of an arbitrary complete basis could lead to “pathologies”. For example, let the basis contain the gate

$$X = \begin{pmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{pmatrix}$$

where θ is a noncomputable number, e.g., the n-th digit of θ says whether the universal Turing machine terminates at input n (Halting problem). Then $p = \sin^2\theta$ is also noncomputable.

Noncomputable – remark

If we apply X to the state $|0\rangle$ and measure the qubit, we will get 1 with probability p and 0 with probability $1-p$. Repeating this procedure $\exp(\Theta(n))$ times and counting the number of 0s and 1s, we can find the n-th digit of p with very small error probability. Thus the gate X enables us to solve the halting problem!

$$X = \begin{pmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{pmatrix}$$

Noncomputable – remark

This argument has nothing to do with quantum mechanics. A classical probabilistic computer could also get extra power if random numbers with arbitrary p were allowed.

Noncomputable – remark

Of course, we want to avoid such things in our theory, so we must be careful about the choice of basis. However, in the real world “superpowerful” gates might exist. Experimentalists measure dimensionless physical constants (such as the fine structure constant) with increasingly high precision, getting new digits of the number theoreticians cannot compute.

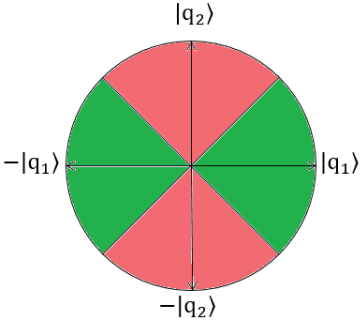
Noncomputable – remark

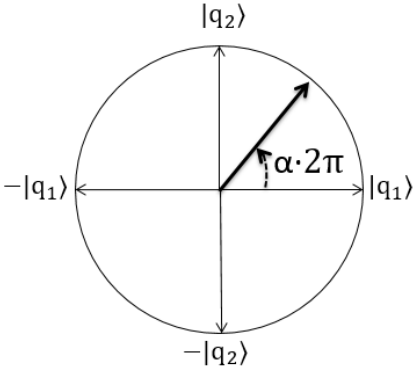
If we ever learn where the fundamental physical constants come from, we will probably know whether they are computable, and if they are not, whether they carry some mathematically meaningful information, e.g., allow one to solve the halting problem.

Noncomputable – research

$\alpha = 0.00\alpha_3\alpha_4\alpha_5 \ldots \alpha_j \ldots$ and $\beta = 0.00\beta_3\beta_4\beta_5 \ldots \beta_j \ldots$.

There exists minimal $j > 2$ such that $\alpha_j \neq \beta_j$. The values of α_{j-2} and α_{j-1} determine the number of a quadrant. For any given two irrational numbers α and β in $(0, 1/4)$, the QFAs M_α and M_β recognize different languages with cutpoint $\frac{1}{2}$.

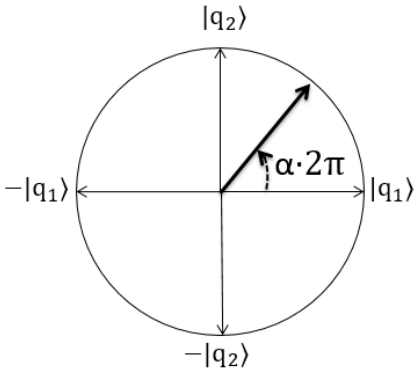




α_{j-2}	α_{j-1}	$quadrant$	$f_{M_\alpha}(x_j)$	$f_{M_\beta}(x_j)$
0	0	I	$< 1/2$	$> 1/2$
0	1	II	$> 1/2$	$< 1/2$
1	0	III	$< 1/2$	$> 1/2$
1	1	IV	$> 1/2$	$< 1/2$

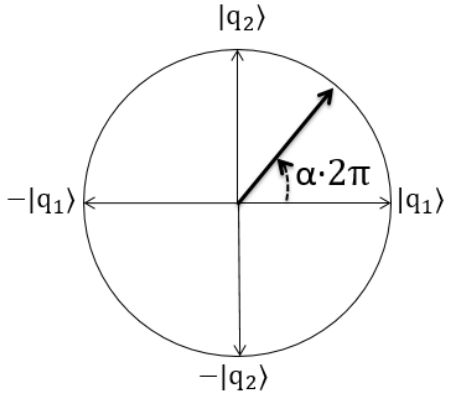
Noncomputable – research

The result allows us to conclude, that noncomputable amplitudes affect the minimal possible quantum computing devices, allowing them to exceed the capabilities of Turing machines. For probabilistic case this affects a bit stronger devices.



Noncomputable – research

Published: Naumovs Aleksejs, Dimitrijevs Maksims, and Yakaryılmaz Abuzer - The minimal probabilistic and quantum finite automata recognizing uncountably many languages with fixed cutpoints. dmtcs:5450 - Discrete Mathematics & Theoretical Computer Science, April 30, 2020, vol. 22 no. 1 - https://doi.org/10.23638/DMTCS-22-1-13



Remark – another definition

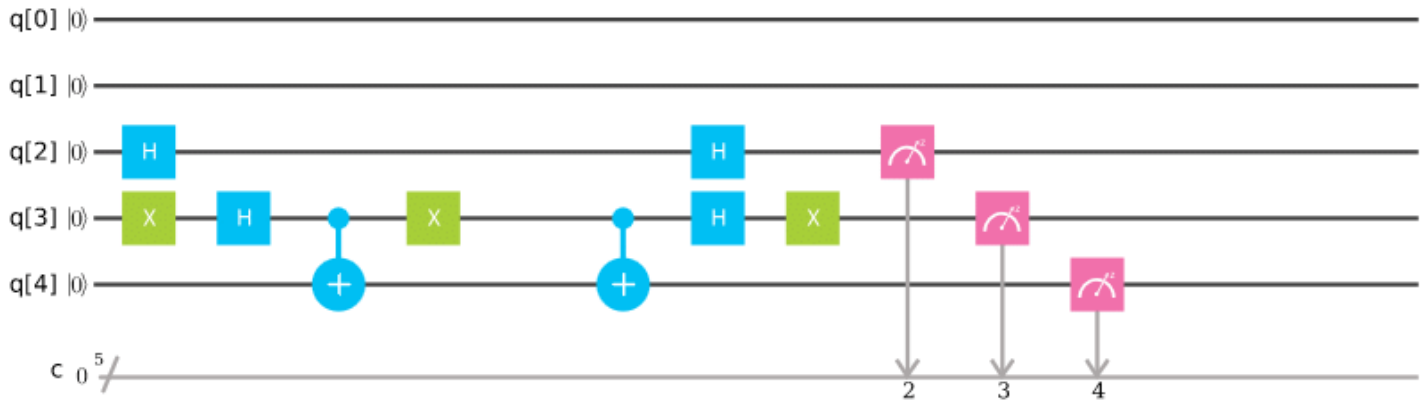
It is possible to define a quantum Turing machine directly, through superpositions of various states of a classical Turing machine (the original definition of D. Deutsch was just like this). The standard definition turns out to be equivalent but more convenient.

Simplifying definition

Using the universal quantum circuit, we can simplify the standard definition even further. It suffices to have a classical Turing machine that generates a description of a quantum circuit $Z(x)$ which is only good to compute $F(x)$ for a single value of x. In this case, x is the input word for the TM whereas the circuit does not have input data at all (i.e., it operates on supplementary qubits initialized by the state $|0^N\rangle$).

Remark – real computers

This definition coincides quite well with current situation how do circuit-model based quantum computers work. Circuit is initialized in state $|0^N\rangle$, input is programmed into circuit, classical description is used to implement a quantum circuit.



Simplifying definition

Indeed, if we have such a machine M, then we can build a machine M' which receives n and constructs a Boolean circuit which computes $Z(x)$ for all values of x, $|x| = n$. By a certain polynomial algorithm, the Boolean circuit can be transformed into a reversible circuit (with garbage) and combined with the universal quantum circuit, so that the output of the former (i.e., $Z(x)$) becomes the “program” for the latter. This yields a quantum circuit that computes F_n.

Quantum algorithm

A quantum algorithm for the computation of a function F: $B^* \to B^*$ is a classical algorithm (i.e., a Turing machine) that computes a function of the form $x \to Z(x)$, where $Z(x)$ is a description of a quantum circuit which computes $F(x)$ on empty input. The function F is said to belong to class BQP if there is a quantum algorithm that computes F in time $poly(n)$.

BQP relations

BPP ⊆ BQP ⊆ PP ⊆ PSPACE

The class PP consists of predicates of the form

$$Q(x) = (|\{y: R_0(x,y)\}| < |\{y: R_1(x,y)\}|)$$

Where $R_0, R_1 \in P$, and y runs through all words of length bounded by some polynomial $q(x)$.

BQP vs BPP

BQP contains the factoring and discrete logarithm problems, the hidden Legendre symbol problem, the Pell's equation and principal ideal problems, and some other problems not thought to be in BPP.

$$BQP = BQP^{BQP}$$

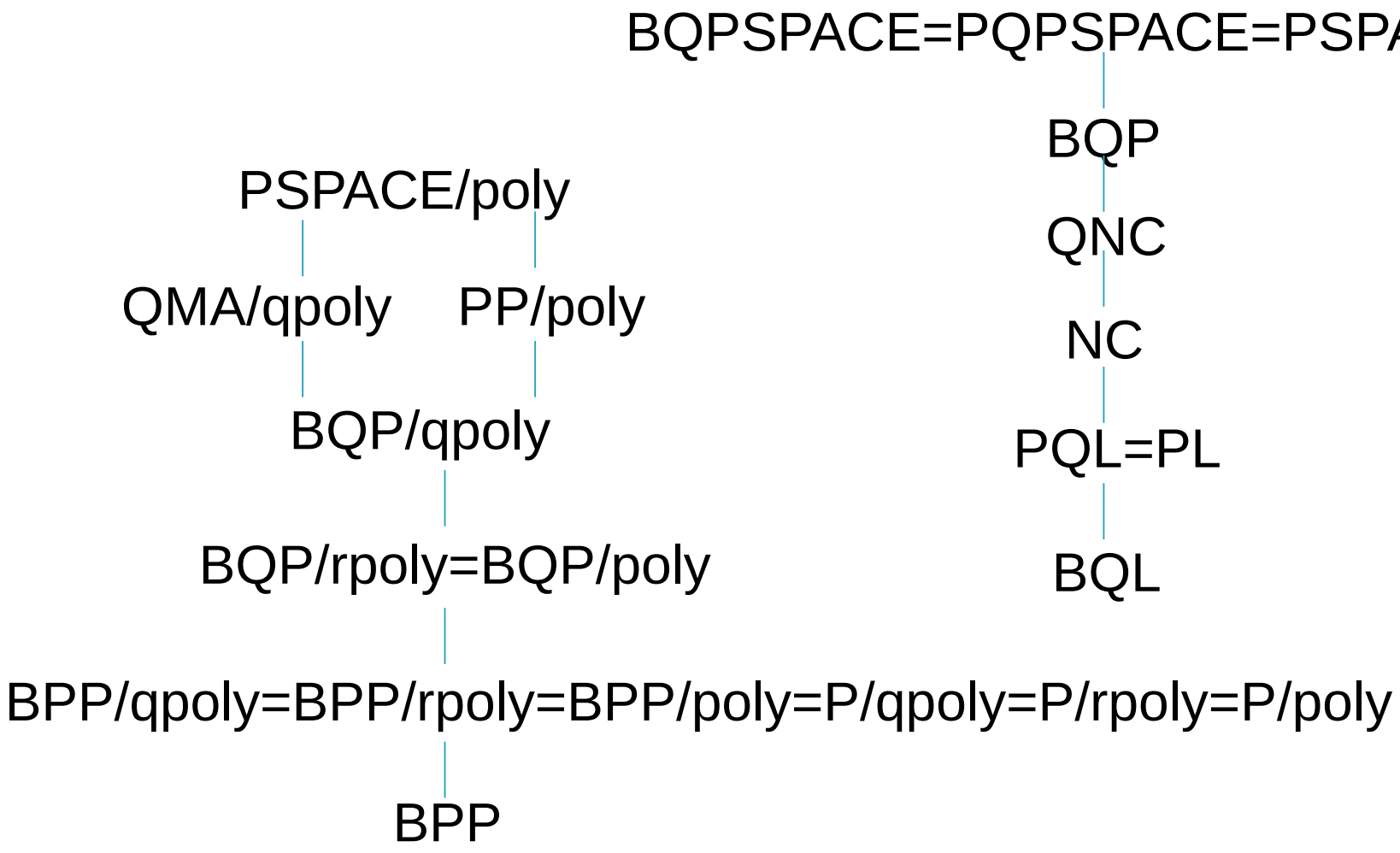
BQP relations

$BPP \subseteq BQP \subseteq PP \subseteq PSPACE$

This is almost all that is known about the correspondence between BQP and the other complexity classes. Indirect evidence in favor of the strict inclusion $BPP \subset BQP$ is given by the existence of effective quantum algorithms for some number-theoretic problems traditionally regarded as difficult.

Complexity classes ⊆

For curious participants



What's next? (A note to the impatient reader).

For other examples

We have spent four sections defining what quantum computation is, but have given only few nontrivial examples so far. The reader may want to see more examples right now. If so, you may skip some material and read Section 13.1 (Simon's algorithm).

Next notions

There will be some references to “mixed states”, but all calculations can be done with state vectors as well. However, most other results are based (not as much formally as conceptually) upon the general notion of quantum probability and measurement. We will proceed to these topics in next lectures.

“Quantum statics”

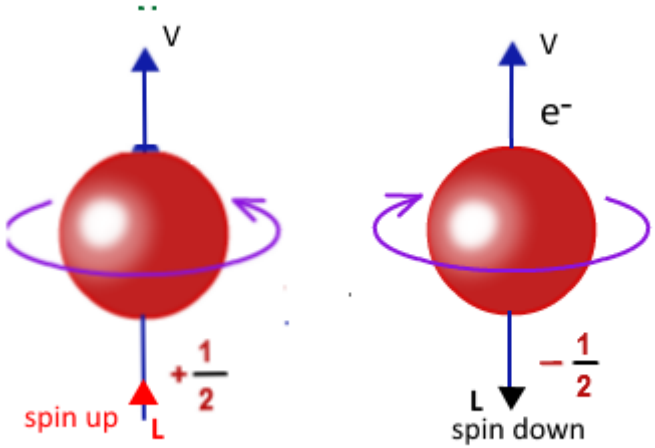
Set of postulates

1. For every physical system there is a corresponding (separable) complex Hilbert space.

Set of postulates

2. The number of “degrees of freedom” of the physical system is chosen as the dimension of the Hilbert space.

Examples: spin of a spin-1/2 particle, photon polarization – 2 degrees of freedom, two-dimensional Hilbert space.



Set of postulates

3. Conversely, for every complex Hilbert space of dimension d, there is a corresponding physical system with d degrees of freedom.

Set of postulates

4. For any “situation” in that physical system there is a unique operator ρ, called the “state”, on the corresponding Hilbert space, such that ρ is Hermitian, positive (+), semidefinite of unit trace.

$$\rho: H \to H, \rho^{\dagger} = \rho, \rho \geq 0, tr\rho = 1$$

Set of postulates

5. “Distinguishable” physical situations are represented in the mathematical formalism by orthogonal states, and vice versa.

ρ_1 and ρ_2 are orthogonal, if $\rho_1\rho_2 = 0$.

Alternatively, two vectors are orthogonal, if their inner product is zero.

Set of postulates

6. In any physical system with a Hilbert space H, for every operator $\rho: H \to H$ that is hermitian, positive semidefinite and of unit trace, there is a unique situation in the physical system that corresponds to ρ.

Some people think this is not true.

Set of postulates

7. Tensor product. If we have two systems corresponding to which the Hilbert spaces are H_1 and H_2, then the Hilbert space corresponding to the join system is $H_1 \otimes H_2$.

Spectral theorem

Normal matrix: $AA^H = A^H A$, unitary: $AA^H = A^H A = I$.

Hermitian matrix: $A^H = A$.

For an Hermitian matrix:

- all eigenvalues are real,
- eigenvectors corresponding to distinct eigenvalues are orthogonal,
- there exists an orthogonal basis of the whole space, consisting of eigenvectors.

Thus all Hermitian matrices are diagonalizable.

Spectral theorem

Normal matrix: $AA^H = A^H A$, unitary: $AA^H = A^H A = I$.

For a unitary matrix:

- all eigenvalues have absolute value 1.
- eigenvectors corresponding to distinct eigenvalues are orthogonal,
- there is an orthogonal basis of the whole space, consisting of eigenvectors.

Thus unitary matrices are diagonalizable. Moreover, for each unitary matrix A there exists a unitary matrix U such that $AU = U\Lambda$, where Λ is a diagonal matrix whose diagonal entries have absolute value 1. The columns of U are eigenvectors of A.

Thank you for your attention!