Theorem

Lecture Topic: Quantum Computing

Quantum computing

In quantum computing, instead of symbols from finite alphabets (e.g., bits), one works with vectors in suitable complex vector spaces.

An extension of BPP to this setting is known as BQP.

Quantum computing

In quantum computing, instead of symbols from finite alphabets (e.g., bits), one works with vectors in suitable complex vector spaces.

An extension of BPP to this setting is known as BQP.

States as vectors

One of the main differences between classical and quantum physics is the fact that quantum states are described by vectors in a complex vector space, rather than binary strings. Abstractly, we use the Dirac, or bra-ket, notation to denote a state vector. If the system is in some state, let us call it ψ, we denote this as

$$|\psi\rangle.$$

This is called a ket. The ψ is just a label of the state while the encasing $|\cdot\rangle$ is there to remind us that this is a vector.

Quick recap of vector spaces

The ket vectors satisfy the ordinary axioms of a vector space. Under addition, the vector space is closed, associative and commutative. There is a unique zero element, which we denote simply by 0, such that

$$|\psi\rangle + 0 = |\psi\rangle. \tag{1.1}$$

The reason why we do not use $|0\rangle$ to denote the zero vector is because we want to reserve that notation for something completely different, as we will see in a short while. There is also a unique vector $(-|\psi\rangle)$ such that

$$|\psi\rangle + (-|\psi\rangle) = 0. \tag{1.2}$$

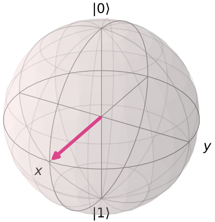
The vector space is linear and distributive under scalar multiplication. This means that for some complex numbers $z, z_1, z_2 \in \mathbb{C}$,

$$|(z_1 + z_2)\psi\rangle = z_1|\psi\rangle + z_2|\psi\rangle, \quad z(|\psi\rangle + |\varphi\rangle) = z|\psi\rangle + z|\varphi\rangle. \tag{1.3}$$

Bloch Sphere

Formally, single qubit can be seen as a two-dimensional complex space, $\mathbb{C}^2$, associated with an inner product and a basis. The standard complex inner product is $v_i{}^\dagger w_i$. The standard basis is $\{|0\rangle, |1\rangle\}$. Together with the inner product, we call $\mathbb{C}^2$ a Hilbert space, sometimes denoted $\mathcal{H}_2$.

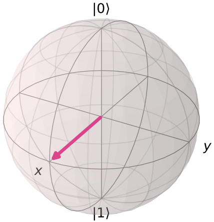
The usual representation of the state of a qubit is that of unit vector $\mathbb{R}^3$ on the so-called Bloch sphere, see Fig. 1.1, which is isomorphic to the complex projective plane $\mathbb{CP}^1$. As such, a qubit's state can be completely characterized as the unit vector on the unit sphere. A quantum state $|\psi\rangle$ and a quantum state $c\,|\psi\rangle$, $c \in \mathbb{C}$ are indistinguishable.



Bloch Sphere

Formally, single qubit can be seen as a two-dimensional complex space, $\mathbb{C}^2$, associated with an inner product and a basis. The standard complex inner product is $v_i{}^\dagger w_i$. The standard basis is $\{|0\rangle, |1\rangle\}$. Together with the inner product, we call $\mathbb{C}^2$ a Hilbert space, sometimes denoted $\mathcal{H}_2$.

The usual representation of the state of a qubit is that of unit vector $\mathbb{R}^3$ on the so-called Bloch sphere, see Fig. 1.1, which is isomorphic to the complex projective plane $\mathbb{CP}^1$. As such, a qubit's state can be completely characterized as the unit vector on the unit sphere. A quantum state $|\psi\rangle$ and a quantum state $c\,|\psi\rangle$, $c \in \mathbb{C}$ are indistinguishable.



Superposition

Formally, just as in any other vector space, we can represent vectors as combinations of basis states. Let the arbitrary state of a qubit be denoted as $|\psi\rangle \in \mathcal{H}_2$. How do we describe this state in terms of the two basis states of $\mathcal{H}_2$?

Let us have two complex numbers $c_x \in \mathbb{C}$ with $x \in \{|0\rangle, |1\rangle\}$, which we will call *amplitudes*. These satisfy $\sum_{x\in\{|0\rangle,|1\rangle\}} |c_x|^2 = 1$.

The general state $|\psi\rangle$ of the qubit, can be seen as:

$$|\psi\rangle = \sum_{x\in\{|0\rangle,|1\rangle\}} c_x |x\rangle \,. \tag{1.4}$$

The squares of the amplitudes can be thought as the probabilities of finding the qubit in a particular basis state.

Superposition

Formally, just as in any other vector space, we can represent vectors as combinations of basis states. Let the arbitrary state of a qubit be denoted as $|\psi\rangle \in \mathcal{H}_2$. How do we describe this state in terms of the two basis states of $\mathcal{H}_2$?

Let us have two complex numbers $c_x \in \mathbb{C}$ with $x \in \{|0\rangle, |1\rangle\}$, which we will call *amplitudes*. These satisfy $\sum_{x\in\{|0\rangle,|1\rangle\}} |c_x|^2 = 1$.

The general state $|\psi\rangle$ of the qubit, can be seen as:

$$|\psi\rangle = \sum_{x\in\{|0\rangle,|1\rangle\}} c_x |x\rangle . \tag{1.4}$$

The squares of the amplitudes can be thought as the probabilities of finding the qubit in a particular basis state.

Superposition

Formally, just as in any other vector space, we can represent vectors as combinations of basis states. Let the arbitrary state of a qubit be denoted as $|\psi\rangle \in \mathcal{H}_2$. How do we describe this state in terms of the two basis states of $\mathcal{H}_2$?

Let us have two complex numbers $c_x \in \mathbb{C}$ with $x \in \{|0\rangle, |1\rangle\}$, which we will call *amplitudes*. These satisfy $\sum_{x\in\{|0\rangle,|1\rangle\}} |c_x|^2 = 1$.

The general state $|\psi\rangle$ of the qubit, can be seen as:

$$|\psi\rangle = \sum_{x\in\{|0\rangle,|1\rangle\}} c_x |x\rangle . \tag{1.4}$$

The squares of the amplitudes can be thought as the probabilities of finding the qubit in a particular basis state.

Superposition

Formally, just as in any other vector space, we can represent vectors as combinations of basis states. Let the arbitrary state of a qubit be denoted as $|\psi\rangle \in \mathcal{H}_2$. How do we describe this state in terms of the two basis states of $\mathcal{H}_2$?

Let us have two complex numbers $c_x \in \mathbb{C}$ with $x \in \{|0\rangle, |1\rangle\}$, which we will call *amplitudes*. These satisfy $\sum_{x\in\{|0\rangle,|1\rangle\}} |c_x|^2 = 1$.

The general state $|\psi\rangle$ of the qubit, can be seen as:

$$|\psi\rangle = \sum_{x\in\{|0\rangle,|1\rangle\}} c_x |x\rangle . \tag{1.4}$$

The squares of the amplitudes can be thought as the probabilities of finding the qubit in a particular basis state.

Observables

In quantum mechanics, observable quantities always are Hermitian operators. Often, one can think of them as Hermitian matrices, that is, complex matrices H with the property $H = H^\dagger$. As a map, an observable simply corresponds to an endomorphism in the Hilbert space, $H : \mathcal{H} \to \mathcal{H}$. An observable H is measured by considering its expectation value when acting on a state $|\psi\rangle$.

$$\langle H \rangle = \langle \psi | H | \psi \rangle = \langle \psi | H\psi \rangle \, . \tag{1.5}$$

Hamiltonian

One of the most important observables in quantum mechanics is the Hamiltonian of a quantum system. When acting on a state, the Hamiltonian provides the energy of the state. The Hamiltonians play a fundamental role in many quantum simulation algorithms.

However, as described above, the Hamiltonian seems to be a very physical concept. Within quantum computation, the role of the Hamiltonian can essentially be assumed by any Hermitian operator. It is customary to call operators that act on qubits as (quantum) *gates* which are usually discussed in the *circuit model of quantum computation*.

Hamiltonian

One of the most important observables in quantum mechanics is the Hamiltonian of a quantum system. When acting on a state, the Hamiltonian provides the energy of the state. The Hamiltonians play a fundamental role in many quantum simulation algorithms.

However, as described above, the Hamiltonian seems to be a very physical concept. Within quantum computation, the role of the Hamiltonian can essentially be assumed by any Hermitian operator. It is customary to call operators that act on qubits as (quantum) *gates* which are usually discussed in the *circuit model of quantum computation*.

Product states

If we imagine that we have several quantum systems, each in some state represented by some state vector, we can combine the separate system into a combined system using the tensor product of vector spaces, $\otimes$. If we imagine that we have one system where the state is given by $|\psi\rangle$ and another where the state is given by $|\varphi\rangle$, the state of the composite system is given by

$$|\psi\rangle \otimes |\varphi\rangle. \tag{1.6}$$

States that can be written in this simple way are called *product states*. Otherwise, we call states entangled. Note that the tensor product does not commute in general.

A Model of n Qubits

A system of n qubits (also known as a quantum register) has a state space $\mathbb{C}^{2^n}$, which can be seen as a tensor product $\mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2$ of the 2-dimensional single-qubit Hilbert spaces, which we denote $(\mathbb{C}^2)^{\otimes n}$. There, each factor corresponds to one qubit.

A system of n qubits is associated with the complex inner product $\langle v|w\rangle = \sum_i v_i^* w_i$ and the standard basis $\{|x_1 x_2 \dots x_n\rangle : x_j \in 0, 1\}$.

We denote the tensor product of N spaces $\mathbb{C}^2$, together with the inner products and the standard basis, by $\mathcal{B}^{\otimes N}$.

A Model of n Qubits

A system of n qubits (also known as a quantum register) has a state space $\mathbb{C}^{2^n}$, which can be seen as a tensor product $\mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2$ of the 2-dimensional single-qubit Hilbert spaces, which we denote $(\mathbb{C}^2)^{\otimes n}$. There, each factor corresponds to one qubit.

A system of n qubits is associated with the complex inner product $\langle v|w\rangle = \sum_i v_i^* w_i$ and the standard basis $\{|x_1 x_2 \dots x_n\rangle : x_j \in 0, 1\}$.

We denote the tensor product of N spaces $\mathbb{C}^2$, together with the inner products and the standard basis, by $\mathcal{B}^{\otimes N}$.

A Model of n Qubits

A system of n qubits (also known as a quantum register) has a state space $\mathbb{C}^{2^n}$, which can be seen as a tensor product $\mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2$ of the 2-dimensional single-qubit Hilbert spaces, which we denote $(\mathbb{C}^2)^{\otimes n}$. There, each factor corresponds to one qubit.

A system of n qubits is associated with the complex inner product $\langle v|w\rangle = \sum_i v_i^* w_i$ and the standard basis $\{|x_1 x_2 \dots x_n\rangle : x_j \in 0, 1\}$.

We denote the tensor product of N spaces $\mathbb{C}^2$, together with the inner products and the standard basis, by $\mathcal{B}^{\otimes N}$.

Entangled states

Entanglement is a quantum mechanical phenomenon where the properties of two or more quantum states become correlated. When entangled, the properties of the qubits are linked in such a way that the state of one qubit cannot be described independently of the other(s). Multi-qubit states that cannot be written as separable states are called *entangled states*. Measuring one qubit of an entangled state will instantaneously affect the properties of the other qubits, regardless of the distance between them. This is known as “spooky action at a distance” and is one of the most mysterious and intriguing aspects of quantum mechanics. We will see that entanglement is necessary but not sufficient for quantum speed-up.

Models of Quantum Computation

Several models of quantum computation have been devised.

Crucially, they do not allow for deciding any problems that are not decidable on a classical computer.

Models of Quantum Computation

Several models of quantum computation have been devised.

Crucially, they do not allow for deciding any problems that are not decidable on a classical computer.

An Alternative Definition of BPP/BQP, due to Arora and Barak

Let a probability threshold be a constant strictly larger than $1/2$. A language $L \subset \{0,1\}^n$ is in BPP or BQP, respectively, if and only if its corresponding indicator function $F(x) : \{0,1\}^n \to \{0,1\}$ can be computed probabilistically in polynomial time such that:

1. one starts with register $v \in [0,1]^{2^N}$ or $\mathbb{C}^{2^N}$, for some $N \geq n$ dependent on F, with an initial state $|x, 0^{N-n}\rangle$ consisting of the input padded to length N by zeros;
2. applies a linear stochastic function $U : \mathbb{R}^{2^N} \to \mathbb{R}^{2^N}$ or $U : \mathbb{C}^{2^N} \to \mathbb{C}^{2^N}$ to v, whose matrix representation can be computed in a sparse format by a Turing machine from all-ones input in time polynomial in n
3. obtains a random variable Y, wherein $F(x)$, i.e., a single 0 or 1, is followed by $N - 1$ arbitrary subsequent symbols with probability at least as high as the probability threshold, wherein the random variable Y has value y with probability v_y or with probability $|v_y|^2$, for the value v of register.

Relationship to Classical Complexity Classes

We know that P $\subseteq$ BPP $\subseteq$ BQP, although the proof is quite non-trivial: one has to establish the power of reversible (classical) circuits and then of the restriction thereof to (classical) permutations.

We know that BQP $\subseteq$ PP. The proof is rather simple.

Because PP $\subseteq$ PSPACE, i.e., the class of languages that can be recognised by a (classical) Turing machine with a polynomial amount of space, we also know BQP $\subseteq$ PSPACE.

Interestingly, there is no material difference between what can be done by a Turing machine with a polynomial amount of space and a quantum Turing machine with a polynomial amount of space.

Unfortunately, we do not know much about the relationship between BQP and non-deterministc Turing machines (NP), other than some relativised results.

Relationship to Classical Complexity Classes

We know that P ⊆ BPP ⊆ BQP, although the proof is quite non-trivial: one has to establish the power of reversible (classical) circuits and then of the restriction thereof to (classical) permutations.

We know that BQP ⊆ PP. The proof is rather simple.

Because PP ⊆ PSPACE, i.e., the class of languages that can be recognised by a (classical) Turing machine with a polynomial amount of space, we also know BQP ⊆ PSPACE.

Interestingly, there is no material difference between what can be done by a Turing machine with a polynomial amount of space and a quantum Turing machine with a polynomial amount of space.

Unfortunately, we do not know much about the relationship between BQP and non-deterministc Turing machines (NP), other than some relativised results.

Relationship to Classical Complexity Classes

We know that P $\subseteq$ BPP $\subseteq$ BQP, although the proof is quite non-trivial: one has to establish the power of reversible (classical) circuits and then of the restriction thereof to (classical) permutations.

We know that BQP $\subseteq$ PP. The proof is rather simple.

Because PP $\subseteq$ PSPACE, i.e., the class of languages that can be recognised by a (classical) Turing machine with a polynomial amount of space, we also know BQP $\subseteq$ PSPACE.

Interestingly, there is no material difference between what can be done by a Turing machine with a polynomial amount of space and a quantum Turing machine with a polynomial amount of space.

Unfortunately, we do not know much about the relationship between BQP and non-deterministc Turing machines (NP), other than some relativised results.

Relationship to Classical Complexity Classes

We know that P ⊆ BPP ⊆ BQP, although the proof is quite non-trivial: one has to establish the power of reversible (classical) circuits and then of the restriction thereof to (classical) permutations.

We know that BQP ⊆ PP. The proof is rather simple.

Because PP ⊆ PSPACE, i.e., the class of languages that can be recognised by a (classical) Turing machine with a polynomial amount of space, we also know BQP ⊆ PSPACE.

Interestingly, there is no material difference between what can be done by a Turing machine with a polynomial amount of space and a quantum Turing machine with a polynomial amount of space.

Unfortunately, we do not know much about the relationship between BQP and non-deterministc Turing machines (NP), other than some relativised results.

Relationship to Classical Complexity Classes

We know that P $\subseteq$ BPP $\subseteq$ BQP, although the proof is quite non-trivial: one has to establish the power of reversible (classical) circuits and then of the restriction thereof to (classical) permutations.

We know that BQP $\subseteq$ PP. The proof is rather simple.

Because PP $\subseteq$ PSPACE, i.e., the class of languages that can be recognised by a (classical) Turing machine with a polynomial amount of space, we also know BQP $\subseteq$ PSPACE.

Interestingly, there is no material difference between what can be done by a Turing machine with a polynomial amount of space and a quantum Turing machine with a polynomial amount of space.

Unfortunately, we do not know much about the relationship between BQP and non-deterministc Turing machines (NP), other than some relativised results.

An Alternative Definition of BPP, due to Fortnow

Let us consider a k-tape extension of a Turing machine:

- a finite, non-empty set Q of objects, representing states
- a subset F of Q, corresponding to "accepting" states, where computation halts
- $q_0 \in Q$, the initial state
- a finite, non-empty set Γ of objects, representing the symbols to be used on any tape
- a partial function $\delta : (Q \setminus F) \times \Gamma^k \to Q \times \Gamma^k \times \{-1, 0, 1\}^k$, where for a combination of a state and k symbols read from the tape, we get the next state, the symbol to write onto the k tapes, and an instruction to shift the k tapes left (-1), right (+1), or keep in its position (0).

An Alternative Definition of BPP, due to Fortnow

In the “Computation as Matrix Multiplication” view of Fortnow, we consider:

- one-step binary version of the transition function: $\delta' : Q \times \Gamma^k \times Q \times \Gamma^k \to \{0, 1\}$, which indicates whether the transition from a configuration c_a to c_b is permitted $c_a, c_b \in C \subseteq (Q \times \Gamma^k)$.
- one-step transition matrix T representing δ' as a $|C| \times |C|$ binary matrix.
- multi-step transition matrix T^r representing the r-step transition function as a $|C| \times |C|$ binary matrix, where $T^r(c_a, c_b) = 1$ if and only if M starting in configuration c_a will be in configuration c_b when run for r steps. $T^r(c_a, c_b)$ is the number of computation paths from c_a to c_b of length r and M accepts if and only if $T^r(c_a, c_b) \geq 1$. For polynomial-time machines, we can obtain the definition of $\#P$ this way.

An Alternative Definition of BPP, due to Fortnow

One can extend this view to probabilistic machines:

- one-step $[0,1]$ version of the transition function: $\delta'' : Q \times \Gamma^k \times Q \times \Gamma^k \rightarrow [0,1]$.
- probabilistic machines use the δ'' with the additional restriction that for any initial state and symbols on the tapes, the values of δ'' for all other arguments sum up to one.
- corresponding one-step transition matrix T and multi-step transition matrices T^r are row and column stochastic.
- Entries of $T^r(c_I, c_A)$ are the probabilities of acceptance by the probabilistic machine.

An Alternative Definition of BPP, due to Fortnow

Let a $0 < \epsilon < 1/2$ be a constant. A language $L \subset \{0, 1\}^n$ is in BPP, if and only if there exists a probabilistic machine as above and a polynomial p such that

- For x in L, we have $T^p(c_I, c_A) \geq 1/2 + \epsilon$.
- For x not in L, we have $T^p(c_I, c_A) \leq 1/2 - \epsilon$.

An Alternative Definition of BPP, due to Fortnow

One can extend this view further to weird machines:

- one-step $[-1, 1]$ version of the transition function: $\delta''' : Q \times \Gamma^k \times Q \times \Gamma^k \rightarrow [-1, 1]$, where the negative values can be intersected with rational numbers.
- weird machines use the δ''' with the additional restriction that the corresponding one-step transition matrix T and multi-step transition matrices T^r are unitary.
- Squared entries of $T^r(c_I, c_A)$ are the probabilities of acceptance by the weird machine.

An Alternative Definition of BQP, due to Fortnow

Let a $0 < \epsilon < 1/2$ be a constant. A language $L \subset \{0,1\}^n$ is in BQP, if and only if there exists a weird machine as above and a polynomial p such that

- For x in L, we have $(T^p(c_I, c_A))^2 \geq 1/2 + \epsilon$.
- For x not in L, we have $(T^p(c_I, c_A))^2 \leq 1/2 - \epsilon$.

Quantum Turing Machine

Deutsch defined the quantum Turing machine with one tape for input and output and one tape for intermediate results using:

- still finite set Σ of symbols used for the inputs and outputs
- Hilbert space instead of a finite set Q of objects, representing states, with an accepting subspace
- Hilbert space instead of a finite set Γ representing symbols to be used on the intermediate result tape, with zero-vector instead of a blank symbol,
- partial function δ is now $\delta : \Sigma \times Q \otimes \Gamma \to \Sigma \times Q \otimes \Gamma \times \{L, R\}$, where each automorphism of the Hilbert space is given by a unitary matrix.

The probabilistic element comes in the form of a measurement, which translates the state to the output upon an accepting subspace is reached.

Quantum Turing machines and quantum circuits were shown to be equivalent in the sense that they can simulate each other in some distributional sense.

Quantum Turing Machine

Deutsch defined the quantum Turing machine with one tape for input and output and one tape for intermediate results using:

- still finite set Σ of symbols used for the inputs and outputs
- Hilbert space instead of a finite set Q of objects, representing states, with an accepting subspace
- Hilbert space instead of a finite set Γ representing symbols to be used on the intermediate result tape, with zero-vector instead of a blank symbol,
- partial function δ is now $\delta : \Sigma \times Q \otimes \Gamma \to \Sigma \times Q \otimes \Gamma \times \{L, R\}$, where each automorphism of the Hilbert space is given by a unitary matrix.

The probabilistic element comes in the form of a measurement, which translates the state to the output upon an accepting subspace is reached.

Quantum Turing machines and quantum circuits were shown to be equivalent in the sense that they can simulate each other in some distributional sense.

A Clarification

While elegant, the analogy with a Turing Machine may be somewhat confusing. It is important to stress that:

There is no branching based on the intermediate results or states. Measurement required by either would collapse the intermediate result or state. The addition of a probabilistic equivalent of branching, known as post-selection, leads to a different complexity class, PostBQP $=$ PP.

There is no computation in the traditional sense. The state $|\psi(nT)\rangle$ at nth time step is simply $U^n|\psi(0)\rangle$ for some constant unitary operator U. In some sense, one hence wishes to represent all possible solutions in the initial state already.

There is no notion a random access memory beyond the qubit register we work with.

A Clarification

While elegant, the analogy with a Turing Machine may be somewhat confusing. It is important to stress that:

There is no branching based on the intermediate results or states. Measurement required by either would collapse the intermediate result or state. The addition of a probabilistic equivalent of branching, known as post-selection, leads to a different complexity class, PostBQP = PP.

There is no computation in the traditional sense. The state $|\psi(nT)\rangle$ at nth time step is simply $U^n|\psi(0)\rangle$ for some constant unitary operator U. In some sense, one hence wishes to represent all possible solutions in the initial state already.

There is no notion a random access memory beyond the qubit register we work with.

A Clarification

While elegant, the analogy with a Turing Machine may be somewhat confusing. It is important to stress that:

There is no branching based on the intermediate results or states. Measurement required by either would collapse the intermediate result or state. The addition of a probabilistic equivalent of branching, known as post-selection, leads to a different complexity class, PostBQP = PP.

There is no computation in the traditional sense. The state $|\psi(nT)\rangle$ at nth time step is simply $U^n|\psi(0)\rangle$ for some constant unitary operator U. In some sense, one hence wishes to represent all possible solutions in the initial state already.

There is no notion a random access memory beyond the qubit register we work with.

A Clarification

While elegant, the analogy with a Turing Machine may be somewhat confusing. It is important to stress that:

There is no branching based on the intermediate results or states. Measurement required by either would collapse the intermediate result or state. The addition of a probabilistic equivalent of branching, known as post-selection, leads to a different complexity class, PostBQP = PP.

There is no computation in the traditional sense. The state $|\psi(nT)\rangle$ at nth time step is simply $U^n |\psi(0)\rangle$ for some constant unitary operator U. In some sense, one hence wishes to represent all possible solutions in the initial state already.

There is no notion a random access memory beyond the qubit register we work with.

Quantum Circuits

Last but not least, the standard model of quantum computing is known as the quantum circuit model of Deutsch, and it is not too different from the alternative definition of BQP, due to Arora and Barak.

Let a probability threshold be a constant strictly larger than 1/2. Consider $F : \{0,1\}^n \to \{0,1\}^m$ and $N \geq \max\{n, m\}$. There, one:

1. starts with an initial state $|x, 0^{N-n}\rangle$ padded to length N.
2. applies a unitary operator $U : \mathcal{B}^{\otimes N} \to \mathcal{B}^{\otimes N}$ (realised by a circuit), which is a composition of multiple unitary operators $U = U_L, U_{L-1}, \cdots U_2, U_1, U_i : \mathcal{B}^{\otimes N} \to \mathcal{B}^{\otimes N}$, where each U_i will be called a gate and L will be the known as the depth of the circuit.
3. obtains $F(x)$ followed by $N - m$ arbitrary subsequent symbols with probability at least as high as a probability threshold.

Quantum Circuits

Last but not least, the standard model of quantum computing is known as the quantum circuit model of Deutsch, and it is not too different from the alternative definition of BQP, due to Arora and Barak.

Let a probability threshold be a constant strictly larger than 1/2. Consider $F : \{0,1\}^n \to \{0,1\}^m$ and $N \geq \max\{n, m\}$. There, one:

1. starts with an initial state $|x, 0^{N-n}\rangle$ padded to length N.
2. applies a unitary operator $U : \mathcal{B}^{\otimes N} \to \mathcal{B}^{\otimes N}$ (realised by a circuit), which is a composition of multiple unitary operators $U = U_L, U_{L-1}, \cdots U_2, U_1, U_i : \mathcal{B}^{\otimes N} \to \mathcal{B}^{\otimes N}$, where each U_i will be called a gate and L will be the known as the depth of the circuit.
3. obtains $F(x)$ followed by $N - m$ arbitrary subsequent symbols with probability at least as high as a probability threshold.

Defining Computation of a Quantum Circuit

Let ϵ be a constant $0 < \epsilon < 1/2$. A circuit U computes $F : \{0,1\}^* \to \{0,1\}^*$ if for any x we have

$$\sum_z | \langle F(x), z | U \rangle \, x, 0^{N-n} |^2 \geq 1 - \epsilon. \tag{1.7}$$

The expression on the left-hand side is, indeed, the probability of getting $F(x)$ padded with with arbitrary z in the measurement of the outcome of U applied to the initial state $|x, 0^{N-n}\rangle$.

Defining Computation of a Quantum Circuit

Let ϵ be a constant $0 < \epsilon < 1/2$. A circuit U computes $F : \{0,1\}^* \to \{0,1\}^*$ if for any x we have

$$\sum_z | \langle F(x), z|U\rangle \, x, 0^{N-n}|^2 \geq 1 - \epsilon. \tag{1.7}$$

The expression on the left-hand side is, indeed, the probability of getting $F(x)$ padded with with arbitrary z in the measurement of the outcome of U applied to the initial state $|x, 0^{N-n}\rangle$.

Defining BQP

A function $\{0,1\}^* \to \{0,1\}^*$ is in BQP, if there exists a deterministic Turing machine M and a polynomial p such that M runs in time $p(|x|)$ and produces a description of a quantum circuit that computes the function.

State preparation

We are now ready to start building quantum circuits. The ingredients will be qubits and unitary operators or gates.

First of all we need to discuss where we will start, i.e., what is the initial state of the system, or the input of the circuit, and how do we prepare that? A simple choice of input vector that is most commonly used is to pick $|0\dots0\rangle$ as the initial state vector. Given some general initial state, how do we prepare it in the $|0\dots0\rangle$? Well, one very simple way is found by remembering that measurements will make the system collapse to a given eigenvector of the observable being measured. We can then simply make a measurement of σ_z on each qubit, which will return the results ±1 with some probabilities. If we get $+1$ we know that the qubit is in the state $|0\rangle$ as desired, while if we find -1 we know that it will be in the state $|1\rangle$. Then we simply keep the qubits that are in the $|0\rangle$ state and act with σ_x on the others, since we saw previously that $\sigma_x|1\rangle = |0\rangle$. Now we have our input vector $|\psi\rangle = |0\dots0\rangle$.

Unitary gates

The quantum circuit will then start with a number of qubits in the $|0\rangle$ state and act on this with some number of gates, or unitary operators. The most basic gates are NOT, CNOT, CCNOT:

- $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$, $\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$ and other classical gates. When acting upon two qubits, he controlled-not, or CNOT, gate, acts in the following way:

$$\begin{aligned} |00\rangle &\to |00\rangle, \\ |01\rangle &\to |01\rangle, \\ |10\rangle &\to |1\rangle \otimes \sigma_x|0\rangle = |11\rangle, \\ |11\rangle &\to |1\rangle \otimes \sigma_x|1\rangle = |10\rangle. \end{aligned} \tag{1.8}$$

Examples

One example circuit is Figures 1.2. One important thing to note is that when we read the circuits we read it from left to right, but when we write it down mathematically the gates act in the opposite order.

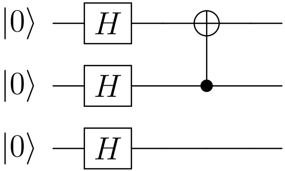


Figure: A simple example of a quantum circuit using the H and CNOT gates.

Looking beyond the Basics

Let us now summarize a few important results briefly, following papers of Zhang, Vala, et al., Dorit Aharonov, and Maarten Van den Nest. These concern:

- "role of entanglement" and "interference": are maximally-entangled states sufficient and necessary?
- "universality": what gates are sufficient to implement any unitary matrix in $SU(n)$?
- "weak simulation": can we sample from the distribution on the measurement of a quantum circuit's first qubit in polynomial time using a classical computer?
- "strong simulation": can we compute the probability of measuring 1 on a quantum circuit's first qubit to any given precision in polynomial time a classical computer?

Universality

A crucial questions relate to "universality": what gates are sufficient to implement any unitary matrix in $SU(n)$?

Traditionally , one considers all one-qubit gates plus CNOT. One often implements controlled rotations by a given angle, the phase shift gate, and CNOT, which are sufficient.

Aharonov defines computational universal the set of gates that can be used to simulate to within ϵ error any quantum circuit which uses n qubits and t gates from a strictly universal set with only polylogarithmic overhead in $(n, t, 1/\epsilon)$. Then, she shows that the set of Toffoli and Hadamard gate is computationally universal.

Contrast this with the classical computation, where Toffoli on its own is universal.

Universality

A crucial questions relate to “universality”: what gates are sufficient to implement any unitary matrix in $SU(n)$?

Traditionally , one considers all one-qubit gates plus CNOT. One often implements controlled rotations by a given angle, the phase shift gate, and CNOT, which are sufficient.

Aharonov defines computational universal the set of gates that can be used to simulate to within ϵ error any quantum circuit which uses n qubits and t gates from a strictly universal set with only polylogarithmic overhead in $(n, t, 1/\epsilon)$. Then, she shows that the set of Toffoli and Hadamard gate is computationally universal.

Contrast this with the classical computation, where Toffoli on its own is universal.

Universality

A crucial questions relate to “universality”: what gates are sufficient to implement any unitary matrix in $SU(n)$?

Traditionally , one considers all one-qubit gates plus CNOT. One often implements controlled rotations by a given angle, the phase shift gate, and CNOT, which are sufficient.

Aharonov defines computational universal the set of gates that can be used to simulate to within ϵ error any quantum circuit which uses n qubits and t gates from a strictly universal set with only polylogarithmic overhead in $(n, t, 1/\epsilon)$. Then, she shows that the set of Toffoli and Hadamard gate is computationally universal.

Contrast this with the classical computation, where Toffoli on its own is universal.

Universality

A crucial questions relate to “universality”: what gates are sufficient to implement any unitary matrix in $SU(n)$?

Traditionally , one considers all one-qubit gates plus CNOT. One often implements controlled rotations by a given angle, the phase shift gate, and CNOT, which are sufficient.

Aharonov defines computational universal the set of gates that can be used to simulate to within ϵ error any quantum circuit which uses n qubits and t gates from a strictly universal set with only polylogarithmic overhead in $(n, t, 1/\epsilon)$. Then, she shows that the set of Toffoli and Hadamard gate is computationally universal.

Contrast this with the classical computation, where Toffoli on its own is universal.

Universality and Lack thereof

We clearly need to be able to produce maximally entangled states, using CNOT, Toffoli, or similar.

Let us consider the question of what gates produce maximally entangled states from some separable states.

One can consider, e.g., using Hadamard and a non-local gate such as CNOT. (Non-local gate is from $SU(4) \setminus SU(2) \otimes SU(2)$.)

Zhang, Vala, et al. have shown that the local equivalence classes of two-qubit gates are in one-to-one correspondence with the points in a tetrahedron, except on the base. Using this tetrahedral representation of non-local gates, they have shown that exactly half the non-local gates are perfect entanglers.

This means that the second half of the non-local gates are imperfect entanglers.

While we need a perfect entangler, the role of CNOT is hence not particularly "central".

Universality and Lack thereof

We clearly need to be able to produce maximally entangled states, using CNOT, Toffoli, or similar.

Let us consider the question of what gates produce maximally entangled states from some separable states.

One can consider, e.g., using Hadamard and a non-local gate such as CNOT. (Non-local gate is from $SU(4) \setminus SU(2) \otimes SU(2)$.)

Zhang, Vala, et al. have shown that the local equivalence classes of two-qubit gates are in one-to-one correspondence with the points in a tetrahedron, except on the base. Using this tetrahedral representation of non-local gates, they have shown that exactly half the non-local gates are perfect entanglers.

This means that the second half of the non-local gates are imperfect entanglers.

While we need a perfect entangler, the role of CNOT is hence not particularly "central".

Universality and Lack thereof

We clearly need to be able to produce maximally entangled states, using CNOT, Toffoli, or similar.

Let us consider the question of what gates produce maximally entangled states from some separable states.

One can consider, e.g., using Hadamard and a non-local gate such as CNOT. (Non-local gate is from $SU(4) \setminus SU(2) \otimes SU(2)$.)

Zhang, Vala, et al. have shown that the local equivalence classes of two-qubit gates are in one-to-one correspondence with the points in a tetrahedron, except on the base. Using this tetrahedral representation of non-local gates, they have shown that exactly half the non-local gates are perfect entanglers.

This means that the second half of the non-local gates are imperfect entanglers.

While we need a perfect entangler, the role of CNOT is hence not particularly "central".

Universality and Lack thereof

We clearly need to be able to produce maximally entangled states, using CNOT, Toffoli, or similar.

Let us consider the question of what gates produce maximally entangled states from some separable states.

One can consider, e.g., using Hadamard and a non-local gate such as CNOT. (Non-local gate is from $SU(4) \setminus SU(2) \otimes SU(2)$.)

Zhang, Vala, et al. have shown that the local equivalence classes of two-qubit gates are in one-to-one correspondence with the points in a tetrahedron, except on the base. Using this tetrahedral representation of non-local gates, they have shown that exactly half the non-local gates are perfect entanglers.

This means that the second half of the non-local gates are imperfect entanglers.

While we need a perfect entangler, the role of CNOT is hence not particularly "central".

Universality and Lack thereof

We clearly need to be able to produce maximally entangled states, using CNOT, Toffoli, or similar.

Let us consider the question of what gates produce maximally entangled states from some separable states.

One can consider, e.g., using Hadamard and a non-local gate such as CNOT. (Non-local gate is from $SU(4) \setminus SU(2) \otimes SU(2)$.)

Zhang, Vala, et al. have shown that the local equivalence classes of two-qubit gates are in one-to-one correspondence with the points in a tetrahedron, except on the base. Using this tetrahedral representation of non-local gates, they have shown that exactly half the non-local gates are perfect entanglers.

This means that the second half of the non-local gates are imperfect entanglers.

While we need a perfect entangler, the role of CNOT is hence not particularly "central".

Universality and Lack thereof

We clearly need to be able to produce maximally entangled states, using CNOT, Toffoli, or similar.

Let us consider the question of what gates produce maximally entangled states from some separable states.

One can consider, e.g., using Hadamard and a non-local gate such as CNOT. (Non-local gate is from $SU(4) \setminus SU(2) \otimes SU(2)$.)

Zhang, Vala, et al. have shown that the local equivalence classes of two-qubit gates are in one-to-one correspondence with the points in a tetrahedron, except on the base. Using this tetrahedral representation of non-local gates, they have shown that exactly half the non-local gates are perfect entanglers.

This means that the second half of the non-local gates are imperfect entanglers.

While we need a perfect entangler, the role of CNOT is hence not particularly "central".

Universality and Lack thereof

Having said that, even the role of Hadamard and CNOT is not particularly central either.

Hadamard, CNOT, and one particular phase shift gate (phase shift by $\pi/2$) generate a group called the Clifford group.

By a non-trivial Gottesman-Knill theorem, the Clifford gates does not make a universal gate set.

In particular, the Gottesman-Knill theorem shows that a uniform family of Clifford circuit(s) acting on the computational basis state $|0\rangle^N$ followed by a computational basis measurement, can be simulated efficiently on a classical computer.

(Actually, their simulation of Clifford-gate circuits belongs to the complexity class $\oplus\mathbf{L}$ ("parity-L") as classical computation with NOT and CNOT gates, which is not believed to equal to P.)

This shows that while maximally entangled states are provably necessary to disallow efficient classical simulation, they are not sufficient.

Universality and Lack thereof

Having said that, even the role of Hadamard and CNOT is not particularly central either.

Hadamard, CNOT, and one particular phase shift gate (phase shift by $\pi/2$) generate a group called the Clifford group.

By a non-trivial Gottesman-Knill theorem, the Clifford gates does not make a universal gate set.

In particular, the Gottesman-Knill theorem shows that a uniform family of Clifford circuit(s) acting on the computational basis state $|0\rangle^N$ followed by a computational basis measurement, can be simulated efficiently on a classical computer.

(Actually, their simulation of Clifford-gate circuits belongs to the complexity class $\oplus\mathbf{L}$ ("parity-L") as classical computation with NOT and CNOT gates, which is not believed to equal to P.)

This shows that while maximally entangled states are provably necessary to disallow efficient classical simulation, they are not sufficient.

Universality and Lack thereof

Having said that, even the role of Hadamard and CNOT is not particularly central either.

Hadamard, CNOT, and one particular phase shift gate (phase shift by $\pi/2$) generate a group called the Clifford group.

By a non-trivial Gottesman-Knill theorem, the Clifford gates does not make a universal gate set.

In particular, the Gottesman-Knill theorem shows that a uniform family of Clifford circuit(s) acting on the computational basis state $|0\rangle^N$ followed by a computational basis measurement, can be simulated efficiently on a classical computer.

(Actually, their simulation of Clifford-gate circuits belongs to the complexity class $\oplus\mathbf{L}$ ("parity-L") as classical computation with NOT and CNOT gates, which is not believed to equal to P.)

This shows that while maximally entangled states are provably necessary to disallow efficient classical simulation, they are not sufficient.

Universality and Lack thereof

Having said that, even the role of Hadamard and CNOT is not particularly central either.

Hadamard, CNOT, and one particular phase shift gate (phase shift by $\pi/2$) generate a group called the Clifford group.

By a non-trivial Gottesman-Knill theorem, the Clifford gates does not make a universal gate set.

In particular, the Gottesman-Knill theorem shows that a uniform family of Clifford circuit(s) acting on the computational basis state $|0\rangle^N$ followed by a computational basis measurement, can be simulated efficiently on a classical computer.

(Actually, their simulation of Clifford-gate circuits belongs to the complexity class $\oplus$**L** ("parity-L") as classical computation with NOT and CNOT gates, which is not believed to equal to P.)

This shows that while maximally entangled states are provably necessary to disallow efficient classical simulation, they are not sufficient.

Universality and Lack thereof

Having said that, even the role of Hadamard and CNOT is not particularly central either.

Hadamard, CNOT, and one particular phase shift gate (phase shift by $\pi/2$) generate a group called the Clifford group.

By a non-trivial Gottesman-Knill theorem, the Clifford gates does not make a universal gate set.

In particular, the Gottesman-Knill theorem shows that a uniform family of Clifford circuit(s) acting on the computational basis state $|0\rangle^N$ followed by a computational basis measurement, can be simulated efficiently on a classical computer.

(Actually, their simulation of Clifford-gate circuits belongs to the complexity class $\oplus$**L** ("parity-L") as classical computation with NOT and CNOT gates, which is not believed to equal to P.)

This shows that while maximally entangled states are provably necessary to disallow efficient classical simulation, they are not sufficient.

Universality and Lack thereof

Having said that, even the role of Hadamard and CNOT is not particularly central either.

Hadamard, CNOT, and one particular phase shift gate (phase shift by $\pi/2$) generate a group called the Clifford group.

By a non-trivial Gottesman-Knill theorem, the Clifford gates does not make a universal gate set.

In particular, the Gottesman-Knill theorem shows that a uniform family of Clifford circuit(s) acting on the computational basis state $|0\rangle^N$ followed by a computational basis measurement, can be simulated efficiently on a classical computer.

(Actually, their simulation of Clifford-gate circuits belongs to the complexity class $\oplus$**L** ("parity-L") as classical computation with NOT and CNOT gates, which is not believed to equal to P.)

This shows that while maximally entangled states are provably necessary to disallow efficient classical simulation, they are not sufficient.

Universality and Lack thereof

In a striking result, Maarten Van den Nest shows that circuits implementing unitaries from the Clifford group (Clifford circuits), which may contain many Hadamard gates at different places in the circuit, causing rounds of constructive and destructive interference, are (efficiently) mapped to circuit that do not utilize any interference at all.

In particular, to circuits where threre is one round of Hadamard gates applied to a subset of the qubits, followed by a round of "classical gates" such as Toffoli, CNOT, NOT, etc.

Let $\mathcal{C}$ be an arbitrary n-qubit Clifford operation. Then there exist: (a) poly-size circuits M_1 and M_2 composed of CNOT, PHASE and CPHASE gates and (b) a tensor product of Hadamard gates and identities $\mathcal{H} = H^S \otimes I$ acting nontrivially on a subset S of the qubits, such that $\mathcal{C} \propto M_2 \mathcal{H} M_1$. Moreover, M_1, M_2 and $\mathcal{H}$ can be determined efficiently.

Universality and Lack thereof

In a striking result, Maarten Van den Nest shows that circuits implementing unitaries from the Clifford group (Clifford circuits), which may contain many Hadamard gates at different places in the circuit, causing rounds of constructive and destructive interference, are (efficiently) mapped to circuit that do not utilize any interference at all.

In particular, to circuits where threre is one round of Hadamard gates applied to a subset of the qubits, followed by a round of “classical gates” such as Toffoli, CNOT, NOT, etc.

Let $\mathcal{C}$ be an arbitrary n-qubit Clifford operation. Then there exist: (a) poly-size circuits M_1 and M_2 composed of CNOT, PHASE and CPHASE gates and (b) a tensor product of Hadamard gates and identities $\mathcal{H} = H^S \otimes I$ acting nontrivially on a subset S of the qubits, such that $\mathcal{C} \propto M_2 \mathcal{H} M_1$. Moreover, M_1, M_2 and $\mathcal{H}$ can be determined efficiently.

Universality and Lack thereof

In a striking result, Maarten Van den Nest shows that circuits implementing unitaries from the Clifford group (Clifford circuits), which may contain many Hadamard gates at different places in the circuit, causing rounds of constructive and destructive interference, are (efficiently) mapped to circuit that do not utilize any interference at all.

In particular, to circuits where threre is one round of Hadamard gates applied to a subset of the qubits, followed by a round of "classical gates" such as Toffoli, CNOT, NOT, etc.

Let $\mathcal{C}$ be an arbitrary n-qubit Clifford operation. Then there exist: (a) poly-size circuits M_1 and M_2 composed of CNOT, PHASE and CPHASE gates and (b) a tensor product of Hadamard gates and identities $\mathcal{H} = H^S \otimes I$ acting nontrivially on a subset S of the qubits, such that $\mathcal{C} \propto M_2 \mathcal{H} M_1$. Moreover, M_1, M_2 and $\mathcal{H}$ can be determined efficiently.

Role of Noise

In real world, all quantum systems interact with the environment.

We often use classical distributions over quantum states to reason about such "partially known" quantum states. Let us associate probability p_k to the event of system being in state $|\alpha_k\rangle$. Such a classical distribution is called a "mixed states", as opposed the usual "pure" state.

A unitary matrix U acts on a mixture $\{p_k, |\alpha_k\rangle\}$ component-wise $\{p_k, U|\alpha_k\rangle\}$.

Role of Noise

In real world, all quantum systems interact with the environment.

We often use classical distributions over quantum states to reason about such "partially known" quantum states. Let us associate probability p_k to the event of system being in state $|\alpha_k\rangle$. Such a classical distribution is called a "mixed states", as opposed the usual "pure" state.

A unitary matrix U acts on a mixture $\{p_k, |\alpha_k\rangle\}$ component-wise $\{p_k, U|\alpha_k\rangle\}$.

Role of Noise

In real world, all quantum systems interact with the environment.

We often use classical distributions over quantum states to reason about such "partially known" quantum states. Let us associate probability p_k to the event of system being in state $|\alpha_k\rangle$. Such a classical distribution is called a "mixed states", as opposed the usual "pure" state.

A unitary matrix U acts on a mixture $\{p_k, |\alpha_k\rangle\}$ component-wise $\{p_k, U|\alpha_k\rangle\}$.

Role of Noise

In a simple model of an open quantum system due to Aharonov and Ben-Or, one assumes:

- single qubit faults: each qubit decoheres independently, or undergoes a fault with probability η per step.
- all operations equal: no decoherence takes place inside the gates.

There, η is referred to as the *decoherence rate*.

This is equivalent to a model, where at each timestep, at each qubit i, we can have a fault with a probability η_i, as long as $\sum_i \eta_i = \eta$.

Role of Noise

In a simple model of an open quantum system due to Aharonov and Ben-Or, one assumes:

- single qubit faults: each qubit decoheres independently, or undergoes a fault with probability η per step.
- all operations equal: no decoherence takes place inside the gates.

There, η is referred to as the *decoherence rate*.

This is equivalent to a model, where at each timestep, at each qubit i, we can have a fault with a probability η_i, as long as $\sum_i \eta_i = \eta$.

Role of Noise

Aharonov and Ben-Or have shown that for models of quantum computing with gates on up to $\log(n)$ qubits, considering the noise model above introduces a delay into the simulation by a probabilistic machine that is polynomial in the number of qubits and depth of the circuit, for any decoherence rate.

Gottesman and Knill suggested that one can correct for a substantial decoherence rate in a Clifford circuit using quantum error correcting codes, at the expense of some overhead in terms of numbers of "physical" qubits.

Role of Noise

Aharonov and Ben-Or have shown that for models of quantum computing with gates on up to $\log(n)$ qubits, considering the noise model above introduces a delay into the simulation by a probabilistic machine that is polynomial in the number of qubits and depth of the circuit, for any decoherence rate.

Gottesman and Knill suggested that one can correct for a substantial decoherence rate in a Clifford circuit using quantum error correcting codes, at the expense of some overhead in terms of numbers of “physical” qubits.