Computability of Quantum Devices

SeungYeop Baik

Yonsei University

Nov. 22, 2023

Overview

Introduction

Today we will summarize the past 3 sessions.

1. How is a 2(1)-way QFA, QTM defined and its expressive power?
2. The complexity hierarchy.
3. Example problems in complexity hierarchy.
4. The biased coin toss problem(the background side)

Overview

Deterministic Finite State Automata

DFA: A *deterministic finite state automaton (DFA)* is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$, where

1. Q is a finite set of *states*;
2. Σ is a finite *input alphabet*;
3. $\delta : Q \times \Sigma \to Q$ is a *transition function*;
4. $q_0 \in Q$ is the *initial state*; and
5. $F \subseteq Q$ is a set of *(final) accepting states*.

2-Way Deterministic Finite State Automata

Definition

2DFA: A *2-way deterministic finite state automaton (2DFA)* is a 6-tuple $(Q, \Sigma, \delta, q_0, Q_{\mathsf{acc}}, Q_{\mathsf{rej}})$, where

1. $\delta : Q \times d \to Q \times \{-1, 0, 1\}$ is a *transition function*; and
2. $Q_{\mathsf{acc}} \subseteq Q$ and $Q_{\mathsf{rej}} \subseteq Q$ are the sets of *accepting states* and *rejecting states*, respectively.

2-Way Deterministic Finite State Automata

Details

Details:

1. $Q_{\mathsf{non}} := Q \setminus (Q_{\mathsf{acc}} \cup Q_{\mathsf{rej}})$;
2. $q_0 \in Q_{\mathsf{non}}$;
3. $Q_{\mathsf{acc}} \cap Q_{\mathsf{rej}} = \emptyset$;
4. ¢ $\notin \Sigma$ and $\$ \notin \Sigma$ are the *start of string* and *end of string* symbols, respectively; and
5. The *tape alphabet* $\Gamma := \Sigma \cup \{¢, \$\}$.

Turing machine

definition

TM: A (deterministic) Turing machine is a 7-tuple $\mathcal{M} = (Q, \Gamma, b, \Sigma, \delta, q_0, F)$, where

1. $\delta : \Sigma \times Q \otimes \Gamma \to \Sigma \times Q \otimes \Gamma \times -1, 1$
2. $F \in Q$ is the set of accepting states.

Turing machine

Details

Details:

1. $b \in \Gamma$ is the blank symbol;
2. An initial state $q_0 \in Q$, and accepting states $F \in Q$;

2-Way Probabilistic Finite State Automata

Definition

2PFA: A *2-way probabilistic finite state automaton (2PFA)* is a 6-tuple $(Q, \Sigma, \delta, q_0, Q_{\mathsf{acc}}, Q_{\mathsf{rej}})$, where

$$\delta : \underline{(Q \times \Gamma)} \times \underline{(Q \times \{-1, 0, 1\})} \to \mathbb{R}.$$

A *distribution* of M on x is a probabilistic distribution $D : C_n \to \mathbb{R}$.

1. For each $c \in C_n$, $[\![c]\!]$ is denotes the distribution $c \mapsto 1$.
2. We can denote a distribution D by $\sum_{c \in C_n} p_c \cdot [\![c]\!]$, where $p_c := D(c)$.

2-Way Probabilistic Finite State Automata

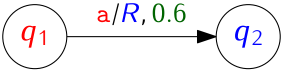
Operator

$$\delta : \underline{(Q \times \Gamma)} \times \underline{(Q \times \{-1, 0, 1\})} \to \mathbb{R}$$

The operator $U_\delta^x : D \mapsto U_\delta^x D$ is defined as:

$$U_\delta^x [\![q, k]\!] := \sum_{q', d} \delta(\underline{q, x(k)}, \underline{q', d}) \cdot [\![q', k + d]\!],$$

and is extended to all distributions of M on x by linearity.



Note

δ is restricted to U_δ^x be valid. What should the restriction be?

Tapes

A *tape* is a mapping $x : \mathbb{Z}_n \to \Gamma$, where $n =: |x|$ is the length of the tape.
We can modify the definition to match each variant of a DFA, 2DFA, 2PFA, and a TM.

For DFAs and PFAs, a string $w = w_1 \cdots w_{|w|} \in \Sigma^*$, we define the tape x_w of w as

1. $x_w(0) :=$ ¢,
2. $x_w(i) := w_i$ for $1 \leq i \leq |w|$, and
3. $x_w(|w| + 1) := \$$.

For TMs, a string $w = w_1 \cdots w_{|w|} \in \Sigma \cup b^*$, we define the tape x_w of w as

1. $x_w(i) := w_i$ for $1 \leq i \leq |w|$, and
2. $x_w(j) := b$ for $j \leq 0$ and $j \geq |w| + 1)$.

Configurations

Fix a DFA $\mathcal{M} := (Q, \Sigma, \delta, q_0, F)$ and a tape x with length n. $C_n := Q \times \mathbb{Z}_n$ is the set of *configurations* of M.

The *time-evolution operator* $U_\delta^x : C_n \to C_n$ of M on tape x is defined as:

$$U_\delta^x(q, k) := (p, k + d),$$

where $\delta(q, x(k)) := (p, d) \in Q \times \{1\}$.

For each time step t, let $(q_t, \_) := (U_\delta^x)^t(q_0, 0)$.
If $q_t \in Q_{\text{acc}}$, then M *accepts* a string w at a time step t.

Configurations

Fix a 2DFA $\mathcal{M} := (Q, \Sigma, \delta, q_0, Q_{\mathsf{acc}}, Q_{\mathsf{rej}})$ and a tape x with length n.
$C_n := Q \times \mathbb{Z}_n$ is the set of *configurations* of M.

The *time-evolution operator* $U_\delta^x : C_n \to C_n$ of M on tape x is defined as:

$$U_\delta^x(q, k) := (p, k + d),$$

where $\delta(q, x(k)) := (p, d) \in Q \times \{-1, 0, 1\}$.

For each time step t, let $(q_t, \_) := (U_\delta^x)^t(q_0, 0)$.
If $q_t \in Q_{\mathsf{acc}}$, then M *accepts* a string w at a time step t.

Configurations

Fix a TM $\mathcal{M} := (Q, \Gamma, b, \Sigma, \delta, q_0, F)$ and a tape x with length n.
$C_n := Q \times \mathbb{Z}_n \Sigma_n$ is the set of *configurations* of M.

The *time-evolution operator* $U_\delta^x : C_n \to C_n$ of M on tape x is defined as:

$$U_\delta^x(q, k) := (p, \sigma, k + d),$$

where $\delta(q, x(k)) := (p, \sigma, d) \in Q \times \Sigma \times \{-1, 0, 1\}$.

For each time step t, let $(q_t, \_) := (U_\delta^x)^t(q_0, 0)$.
If $q_t \in Q_{\mathsf{acc}}$, then M *accepts* a string w at a time step t.

Known Results

Known results:

1. DFA and 2DFA have the same power of expression (regular).
2. Under constant error bound and *exponential* expected time constraints, 2PFA can express the non-regular language $\{\mathtt{a}^n\mathtt{b}^n \mid n > 0\}$.
3. Under constant error bound and *polynomial* expected time constraints, 2PFA cannot express non-regular languages.

Theorem (Dwork89)

For any 2PFA recognizing a non-regular language with a constant error bound, the 2PFA must take exponential expected time with respect to the length of the input.

Overview

Definition of Quantum Automata

2QFA: A *2-way quantum finite state automaton (2QFA)* is a 6-tuple $(Q, \Sigma, \delta, q_0, Q_{\text{acc}}, Q_{\text{rej}})$, where

$$\delta : Q \times \Gamma \times Q \times \{-1, 0, 1\} \to \mathbb{C}.$$

1QFA: A *1-way quantum finite state automaton (1QFA)* is a 2QFA $\mathcal{M} = (Q, \Sigma, \delta, q_0, Q_{\text{acc}}, Q_{\text{rej}})$ where

1. $\delta : Q \times \Gamma \times Q \times \{1\} \to \mathbb{C}$,
2. $\delta(q, \sigma, q', 1) = \langle q' | V_\sigma | q \rangle$, and
3. $\delta(q, \sigma, q', 0) = \delta(q, \sigma, q', -1) = 0$.

Definition of Quantum Automata

QTM: A *quantum Turing machine (QTM)* is a 7-tuple $(Q, \Gamma, b, \Sigma, \delta, q_0, F)$, where

1. Q is a set of states on a Hilbert Space.
2. $\delta : (\Gamma \times Q) \times (\Sigma \times Q \times \{-1, 0, 1\}) \to \mathbb{C}$,
3. $q_0 \in Q$ is a pure or mixed state, and

Configurations of Quantum Automata

A *superposition* of M on x is a $|C_n|$-dimensional quantum state.

1. $\mathcal{H}_n$ denotes the set of all superpositions.
2. For each $c \in C_n$, $|c\rangle$ denotes the unit vector with value 1 at c.
3. For $|\psi\rangle = \sum_{c \in C_n} \alpha_c |c\rangle$, $\alpha_c \in \mathbb{C}$ is the *amplitude* of c in $|\psi\rangle$.

Transitions of 2QFA

$$\delta : \underline{(Q \times \Gamma)} \times \underline{(Q \times \{-1, 0, 1\})} \to \mathbb{C}.$$

For a tape x, the *time-evolution operator* $U_\delta^x : \mathcal{H}_n \to \mathcal{H}_n$ of M on tape x is defined as:

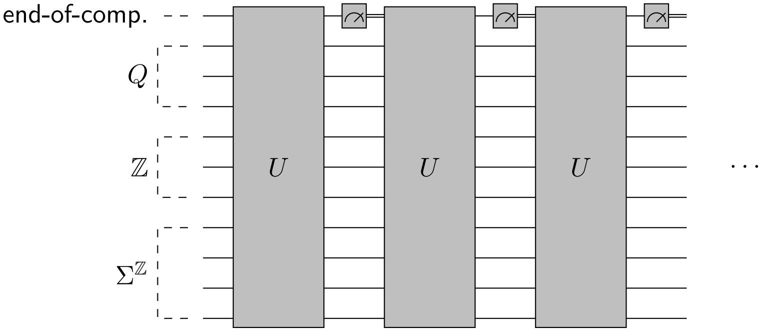
$$U_\delta^x |q, k\rangle := \sum \delta(\underline{q, x(k)}, \underline{q', d}) \cdot |q', k + d\rangle,$$

and is extended to all $|\psi\rangle \in \mathcal{H}_n$ by linearity.

Note

- δ is restricted to U_δ^x be valid, that is, U_δ^x must be *unitary*.
- The configuration of M is in "superposition", we must "carefully observe" whether the configuration is accepting.

Transitions of a QTM



$$\delta : (\Sigma \times Q \times \Gamma) \times (\Sigma \times Q \times \{-1, 0, 1\}) \to \mathbb{C}.$$

Configuration of QTM

Let $\mathcal{H}_A$ be a Hilbert space containing A.

A configuration $(q, x, w) \in Q \times \mathbb{Z} \times \Sigma^{\mathbb{Z}}$ of $\mathcal{Q}$ is encoded into (infinite) qubits as $|q; x; w\rangle = |q\rangle \otimes |x\rangle \otimes |w\rangle \in \mathcal{H}_Q \oplus \mathcal{H}_{\mathbb{Z}} \oplus \mathcal{H}_{\Sigma^{\mathbb{Z}}}$.

Note that, even we describe a QTM configuration in infinite qubits, we only use a *finite* portion of them to compute *effectively*.

Observables

An *observable* $\mathcal{O}$ is a decomposition $\{E_1, \ldots, E_k\}$ of the Hilbert space $\mathcal{H}_n$ into subspaces, where

- $\mathcal{H}_n = E_1 \oplus E_2 \cdots \oplus E_k$; and
- E_j are pairwise orthogonal.

Consider that we observe $|\psi\rangle \in \mathcal{H}_n$ with an observable $\mathcal{O} = \{E_1, \ldots, E_k\}$.
Let $|\psi_j\rangle$ be the projection of $|\psi\rangle$ onto E_j.
Then, after the observation,

1. We observe each outcome j with probability $\||\psi_j\rangle\|^2$.
2. The machine "collapse" to $\frac{1}{\||\psi_j\rangle\|}|\psi_j\rangle$.

Note

It is similar to *conditional probabilities*.

Observables (Examples)

Let $|\psi\rangle = \frac{1}{2}|00\rangle + \frac{1}{2}|01\rangle + \frac{1}{2}|10\rangle + \frac{1}{2}|11\rangle$.

1. Using observable $\{\langle|00\rangle\rangle, \langle|10\rangle\rangle, \langle|01\rangle\rangle, \langle|11\rangle\rangle\}$, with the same probability 0.25,
   - $|\psi\rangle$ collapses to $|00\rangle, |10\rangle, |01\rangle$, or $|11\rangle$.

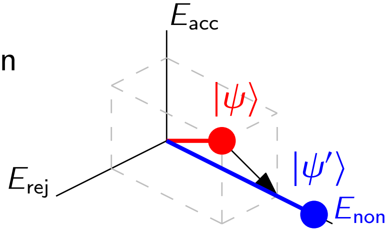
2. Using observable $\{\langle|00\rangle, |10\rangle\rangle, \langle|01\rangle, |11\rangle\rangle\}$, with the same probability 0.5,
   - $|\psi\rangle$ collapses to $\frac{\sqrt{2}}{2}(|00\rangle + |10\rangle)$; or
   - $|\psi\rangle$ collapses to $\frac{\sqrt{2}}{2}(|01\rangle + |11\rangle)$.

Observable of 2QFA

For a 2QFA M and an input x,
we use an observable $\mathcal{O} := \{E_{\mathsf{acc}}, E_{\mathsf{rej}}, E_{\mathsf{non}}\}$, where

- $E_{\mathsf{acc}} := \langle C_{\mathsf{acc}} \rangle$, $C_{\mathsf{acc}} := Q_{\mathsf{acc}} \times \mathbb{Z}_n$ (C_{acc} is the set of all accepting configurations);
- $E_{\mathsf{rej}} := \langle C_{\mathsf{rej}} \rangle$, $C_{\mathsf{rej}} := Q_{\mathsf{rej}} \times \mathbb{Z}_n$; and
- $E_{\mathsf{non}} := \langle C_{\mathsf{non}} \rangle$, $C_{\mathsf{non}} := Q_{\mathsf{non}} \times \mathbb{Z}_n$.

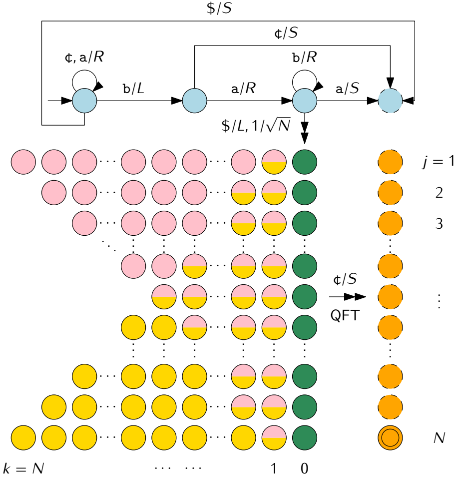
Observation



Overview

2QFA

A 2QFA can recognize a non-regular language $\{\mathtt{a}^n\mathtt{b}^n \mid n > 0\}$, and the non-context-free language $\{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^n \mid n > 0\}$.



Total-States and Computation of 1QFA

A *total-state* of an 1QFA M is $(\psi, p_{\text{acc}}, p_{\text{rej}}) \in \mathcal{V} := \ell_2(Q) \times \mathbb{R} \times \mathbb{R}$.
Intuitively,

1. ψ denotes *unnormalized* superposition $|\psi\rangle$,
2. p_{acc} is the (accumulated) accepting probability, and
3. p_{rej} is the rejecting probability.

The intuition become clearer with the following operator T_σ.

$$\begin{aligned} T_\sigma : (\psi, p_{\text{acc}}, p_{\text{rej}}) \\ \mapsto (P_{\text{non}} V_\sigma \psi, \|P_{\text{acc}} V_\sigma \psi\|^2 + p_{\text{acc}}, \|P_{\text{rej}} V_\sigma \psi\|^2 + p_{\text{rej}}), \end{aligned}$$

where P_{acc}, P_{rej} and P_{non} are the projection matrices onto $\langle Q_{\text{acc}} \rangle$, $\langle Q_{\text{rej}} \rangle$ and $\langle Q_{\text{non}} \rangle$.

Distance of Total-States

$$T_\sigma : \mathcal{V} : (\psi, p_{\mathsf{acc}}, p_{\mathsf{rej}}) \\ \mapsto (P_{\mathsf{non}} V_\sigma \psi, \|P_{\mathsf{acc}} V_\sigma \psi\|^2 + p_{\mathsf{acc}}, \|P_{\mathsf{rej}} V_\sigma \psi\|^2 + p_{\mathsf{rej}}),$$

For two total-states $v = (\psi, p_{\mathsf{acc}}, p_{\mathsf{rej}})$ and $v' = (\psi', p'_{\mathsf{acc}}, p'_{\mathsf{rej}})$, we define a *norm* of v as:

$$\|v\| := \frac{1}{2}(\|\psi\| + |p_{\mathsf{acc}}| + |p_{\mathsf{rej}}|).$$

Then a *distance* between total-states v and v' is

$$d(v, v') := \|v - v'\|.$$

Reachable Total-States

If $v = T_{¢w}|q_0, 0, 0\rangle$ for some $w \in \Sigma^*$, we call v is *reachable* by w.

Let $\mathcal{B} := \{v \in \mathcal{V} \mid \|v\| \leq 1\}$.
Clearly, any valid total-state v must be in $\mathcal{B}$.

Note that

1. T_x increases the distance at most linearly: $d(T_\sigma v, T_\sigma v') \leq c \cdot d(v, v')$,
2. $A \subset \mathcal{B}$ and $\exists \epsilon > 0, \forall v, v' \in A, d(v, v') > \epsilon$ implies A is finite.

QTM and quantum circuits

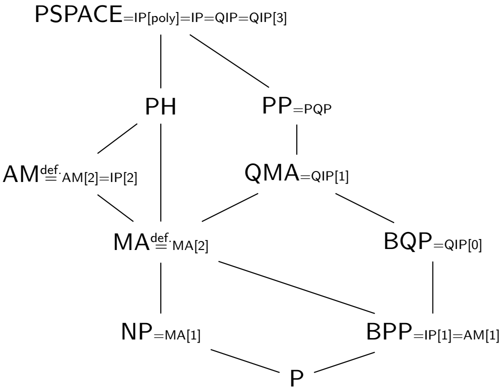
Remark

For a language L, the followings are equivalent:

1. There exists a poly-time QTM $\mathcal{Q}$ for L.
2. There exists a uniform family of quantum circuits $\{Q_n\}_n$ and a poly-time DTM $\mathcal{M}$ such that $\langle Q_n \rangle = \mathcal{M}(1^n)$ and $Q_{|\langle w \rangle|}(\langle w \rangle) = 1(w \in L)$. Q_n may have poly(n) ancilla qubits initialized to $|0\rangle$.

Overview

Complexity hierarchy



$\mathcal{C}[k]$: k machine activations; QIP has a slightly different definition: k messages passing.

Bounded quantum polynomial

BQP: The class BQP contains a language L which has a polynomial QTM $\mathcal{Q}$ with error bounded by $0 \leq c < 0.5$.
Formally, $\exists \mathcal{Q} \exists c \forall w \ [c \in [0, 0.5) \wedge \Pr[\mathcal{Q}(\langle w \rangle) \neq 1(w \in L)] \leq c]$.

Bounded quantum polynomial

BQP: The class BQP contains a language L which has a polynomial QTM $\mathcal{Q}$ with error bounded by $c = 1/3$.
Formally, $\exists \mathcal{Q} \forall w \left[\Pr[\mathcal{Q}(\langle w \rangle) \neq 1(w \in L)] \leq \frac{1}{3} \right]$.

Implication of BQP

Let O be an oracle for a BQP language L; we can effectively decide $w \in L$ by querying O repeatedly.

```
given: input w and iteration count i
for j ∈ [1, i] do
    collect O(w)
end for
return majority
```

Implication of BQP

Let O be an oracle for a BQP language L; we can effectively decide $w \in L$ by querying O repeatedly.

$i = 1$	T	F
$P > N$	$\geq 2/3$	$\geq 1/3$
$P < N$	$\leq 1/3$	$\leq 2/3$

Implication of BQP

Let O be an oracle for a BQP language L; we can effectively decide $w \in L$ by querying O repeatedly.

$i = n^1$	T	F
$P > N$	$\geq 1 - \delta$	$\leq \delta$
$P < N$	$\leq \delta$	$\geq 1 - \delta$

[1] $n \geq -48 \log \delta$

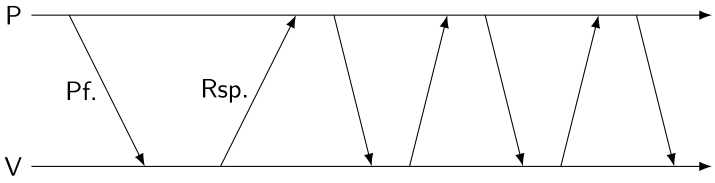
Interactive proof system

Assume a statement is given and there are two people to prove this statement: prover (P) and verifier (V).

- P tries to convince V that the given statement is true.
- V checks P's proof with randomness.

The system accept/reject the statement by passing messages between the two.

It is important that P is unreliable; He may give a false proof.



Interactive proof system

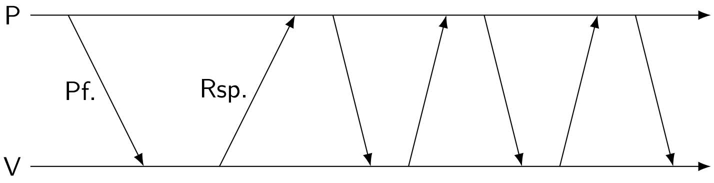
$\boxed{\text{P} \neq \text{NP},\ w \in L,\ \phi \text{ satisfiable, etc.}}$

Assume a $\boxed{\text{statement}}$ is given and there are two people to prove this statement: prover (P) and verifier (V).

- P tries to convince V that the given statement is true.
- V checks P's proof with randomness.

The system accept/reject the statement by passing messages between the two.

It is important that P is unreliable; He may give a false proof.



Arthur-Merlin Framework

Arthur-Merlin framework is another name of the interactive proof system, where we have two machines: Merlin (P) and Arthur (V) with the random coin tosses of Arthur must be revealed.

A language L is in the class Merlin-Arthur (MA) if, there exists a poly-time probabilistic machine (Arthur, V) such that,

- $\forall x \in L$, there exists a proof that makes V accept the statement with prob. at least $2/3$.
- $\forall x \notin L$, for any proof, V accepts the statement with prob. at most $1/3$.

QMA

The QMA class is similar to MA; but here, Arthur is a quantum device without randomness, and the proof is a quantum state (superposition) encoded in a poly-number of qubits.

Formally,
QMA: A language L is in class QMA if there exists a poly-time quantum verifier V such that

- $\forall x \in L$, $\exists |\psi\rangle \;\; \Pr[V(|x\rangle, |\psi\rangle) = 1] \geq 2/3$,
- $\forall x \notin L$, $\forall |\psi\rangle \;\; \Pr[V(|x\rangle, |\psi\rangle) = 1] \leq 1/3$,

where $|\psi\rangle$ is encoded in poly($|x|$) qubits.

Note that V is a BQP machine.

Analogy

BQP and QMA has similar relation to P and NP.

- Any language L in NP has a poly-time verifier V checking certificate.
- Any language L in QMA has a BQP verifier V checking certificate with high prob..

Recap: reduction

Oracle TM: A TM M with oracle O is a TM together with

- a dedicated tape for the oracle O,
- two dedicated states O_{start} and O_{end}.

The oracle O will read an input and write the output using the dedicated tape; in the view of TM M, this takes a single computation step.

Recap: reduction

Turing reduction: For two languages A and B, A Turing reduction from A to B is a TM M with B oracle that decides $w \in A$. If there exists a Turing reduction, A is B-computable.
Then, we can recognize A using *any* machine recognizing B.

Remark

If Turing reduction (from A to B) runs in polynomial time, it is Cook reduction.

BQP-complete

BQP-hard: A language L is BQP-hard if every BQP language L' has a BPP (bounded probabilistic polynomial) TM with an oracle for L. (Assuming that BPP $\neq$ BQP.)

Remark

There are *no* known BQP-complete problems yet.

Promise problem

Promise problem: A promise problem $P : \Sigma^* \to \{0, 1\}$ has two disjoint languages $L_1, L_0 \in \Sigma^*$, where

- $P(w) = 1$ when $w \in L_1$ and
- $P(w) = 0$ when $w \in L_0$.

The language $L = L_1 \cup L_0$ is the *promise* of P.
Note that, for $w \notin L$, $P(w)$ has no requirements.

Remark

A decision problem L is equivalent to a promise problem $(L, \Sigma^* \setminus L)$.

Promise problem

Deutsch-Jozsa: For given an oracle for a constant or balanced function function $f:\{0,1\}^n \rightarrow \{0,1\}$, determine if f is constant.

Promise problem

Promise $(L_1 \cup L_0)$

Deutsch-Jozsa: For given an oracle for a $\boxed{\text{constant or balanced function}}$ function $f : \{0,1\}^n \to \{0,1\}$, determine if $\boxed{f \text{ is constant}}$.

L_1

BQP-complete

Problem (Canonical PromiseBQP problem)

Given a family $\{Q_n\}_n$ of poly-size uniform quantum circuits associated with two disjoint languages L_1 and L_0, and an input $x \in L_1 \cup L_0$ with the *promise* that $Q_{|x|}(x)$ gives

- 1 with prob. at least 2/3 for all $x \in L_1$,
- 1 with prob. at most 1/3 for all $x \in L_0$,

determine that which case holds, i.e., determine that the probability of $Q_{|x|}(x) = 1$ is either above 2/3, or below 1/3.

BQP-complete

Problem (Canonical PromiseBQP problem)

Given a family $\{Q_n\}_n$ of poly-size uniform quantum circuits associated with two disjoint languages L_1 and L_0, and an input $x \in L_1 \cup L_0$ with the *promise* that $Q_{|x|}(x)$ gives

In decision version, there may be a circuit that does not satisfy this condition.

- 1 with prob. at least $2/3$ for all $x \in L_1$,
- 1 with prob. at most $1/3$ for all $x \in L_0$,

determine that which case holds, i.e., determine that the probability of $Q_{|x|}(x) = 1$ is either above $2/3$, or below $1/3$.

BQP-complete

Assume: an oracle O for the canonical problem.
For a PromiseBQP problem (L_1, L_0), $\exists$DTM M that generates a family $\{Q_n = M(1^n)\}_n$ of quantum circuits.

Then, we can solve (L_1, L_0) by the following algorithm.

1. Query O with input $\langle M, x\rangle$.
2. Return [Prob. is at least 2/3] iff O gives an output 1.

This is a (det. linear time, with the oracle O) reduction to the canonical problem.

BQP-complete

Assume: an oracle O for the canonical problem.
For a PromiseBQP problem (L_1, L_0), $\exists$DTM M that generates a family $\{Q_n = M(1^n)\}_n$ of quantum circuits.

Then, we can solve (L_1, L_0) by the following algorithm

M fixed; copy x to oracle tape.

1. Query O with input $\boxed{\langle M, x \rangle}$.
2. Return [Prob. is at least 2/3] iff O gives an output 1.

This is a (det. linear time, with the oracle O) reduction to the canonical problem.

QMA-complete

Problem (QCSAT)

(A quantum variant of classical circuit SAT problem) Given a quantum circuit Q, with n input qubits and m ancilla qubits with the *promise* that Q is either

- $\exists|\psi\rangle$ such that $Q(|\psi\rangle)$ accepts with prob. at least 2/3 or
- $\forall|\psi\rangle$, $Q(|\psi\rangle)$ accepts with prob. at most 1/3,

determine that which case holds.

Reduction is similar to that for the canonical BQP problem.

Complete problems

From [?] (BQP-c) and [?] (QMA-c) (This has several others),
BQP-complete:

- A sampling variant of k-local Hamiltonian: approximate distribution of k-local Hamiltonian's eigenvalues.

QMA-complete:

- Quantum circuit equivalence: Deciding whether two quantum circuits are equivalent
- k-local Hamiltonian: Find the smallest eigenvalue of k-local Hamiltonian.

Overview

The biased coin toss problem

Biased Coin Toss: Given an infinite seq of coin tosses (HTHTT...) such that each toss is an independent event, investigate the ability of a finite automaton to distinguish fair coin, or biased ($p = \frac{1}{2} \pm \epsilon$) coin

Backgrounds

1. What is a mixed state?
2. Extending unitary operation to a mixed state.
3. Distance between spaces?