Computability of Quantum Devices

Hyunjoon Cheon

Yonsei University

Nov. 1, 2023

Overview

Computability

Quantum Turing machine

Computation complexity

Overview

Computability

Problem

Is a given function $f : \mathbb{N} \to \mathbb{N}$ *computable*?

Example

$\mathrm{PRIMES} : \mathbb{N} \to \{0, 1\}$ is computable.

Example

Let the ‘busy beaver’ $\mathrm{BB} : n \mapsto \mathrm{BB}(n)$ be the maximum number of nonzero output symbols of n-state Turing machine over a binary alphabet. Then, BB is incomputable [Rad62].

n	1	2	3	4	5
$\mathrm{BB}(n)$	1	4	6	13	≥ 4098

Table: First five values of the busy beaver

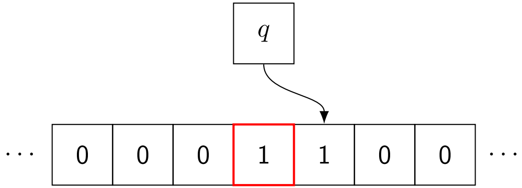
Turing machine

Let $\Sigma = \{0, 1\}$ denote a binary alphabet throughout this talk.
Turing machine (TM): A (determinstic) Turing machine $\mathcal{M} = (Q, E, i, f)$ consists of:

- A finite set Q of states,
- A transition function $E \subseteq Q \times \Sigma \times Q \times \Sigma \times \{-1, 1\} \to \{0, 1\}$ of transitions that satisfies a condition: $\forall (q, \sigma)$, $\sum_{(q', \sigma', d)} E(q, \sigma, q', \sigma', d) \leq 1$.
- An initial state $i \in Q$ and a final, accepting state $f \in Q$,

Configuration of TM

Configuration: A *configuration* of TM $\mathcal{M} = (Q, E, i, f)$ is an element of set $\mathcal{C} = Q \times \mathbb{Z} \times \Sigma^{\mathbb{Z}}$ that describes the current circumstance of $\mathcal{M}$.



Configuration: $(q, 1, 110)$

Figure: Illustration of a Turing machine, the red cell denotes the initial head position, i.e. the beginning of input.

Semantics of TM

Derivation: If two configurations $c_1 = (q, x, w)$ and $c_2 = (q', x', w')$ satisfy:

- $E(q, w_x, q, w'_x, d) = 1$ ($\mathcal{M}$ follows the transition),
- $x + d = x'$ (Head moves) and
- $w_y = w'_y$ if $y \neq x$. (Not modifying other cells),

we say $c_1 \vdash_{\mathcal{M}} c_2$.
$\vdash^*_{\mathcal{M}}$ is a transitive and reflexive closure of $\vdash_{\mathcal{M}}$.

Computation

Computation: a sequence of configurations.
Valid computation: a computation $C = c_1 c_2 \ldots$ satisfying $c_i \vdash_{\mathcal{M}} c_{i+1}$ for all i.
Accepting computation: A valid computation $C = c_0 c_1 \ldots c_n$ is accepting if $c_0 = (i, 0, w)$ and, the state of c_n is final and only the state is final.

Language of TM

Language: Given a TM $\mathcal{M} = (Q, E, i, f)$, its language $L(\mathcal{M})$ is

$$L(\mathcal{M}) = \{w \in \Sigma^* \mid \exists \text{acc. comp. of } c_0 = (i, 0, w) \text{ and } c_n = (f, 0, w')\},$$

where $w' \in \Sigma^*$.

When we consider such $\mathcal{M}$ as a function, we say that $\mathcal{M}(w) = w'$.

Computability

Church-Turing Thesis

A function $f : \mathbb{N} \to \mathbb{N}$ can be *effectively* computable if and only if it is computable by a Turing machine.

Here, *effectively* means that the computation of f is deterministic and it eventually terminates, giving an output.

Complexity

Complexity class: a set of languages (a.k.a. functions) satisfying *specific* properties.

- ALL: *all* languages.
- RE: languages *recognizable* by a Turing machine.
- R: languages *decidable* by a Turing machine.
- P: languages decidable by a *polynomial-time* Turing machine.

Overview of complexity hierarchy

$\mathsf{ALL} = 2^{\Sigma^*}$

R_{ecursively} E_{numerable}

R_{ecursive} = Decidable

PSPACE

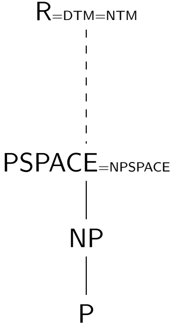
P

(Dashed lines denote proper inclusions.)

- BB is in ALL $\setminus$ RE.
- HALT and PCP are in RE $\setminus$ R.
- PRIMES is in P [AKS04].

Nondeterminism

Nondeterministic TM: A nondeterministic TM (NTM) $\mathcal{M} = (Q, E, i, f)$ is a DTM without the determinism condition on E.



To quantum era

- How we can inject the quantum concepts in TM?
- Will the model be more powerful (or less powerful) than classical TM?

Overview

Computability

Quantum Turing machine

Computation complexity

Quantum Turing machine

Quantum TM: A quantum TM (QTM) $\mathcal{Q} = (Q, U, i, f)$ has four components (c.f. TM $\mathcal{M} = (Q, E, i, f)$):

- A finite set Q of states.
- A complex-valued unitary matrix U.
- An initial state $i \in Q$.
- A final, accepting state $f \in Q$.

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*.

The unitary operation

Starting from an initial superposition $|\psi(0)\rangle$ at time 0, the unitary operator U maps its superposition $|\psi(T)\rangle$ at time T as (c.f., Schrödinger eq.):

$$|\psi(T)\rangle = U^T|\psi(0)\rangle,$$

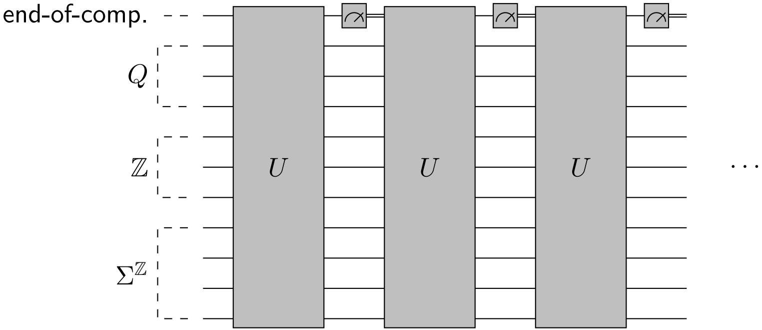
where the initial superposition $|\psi(0)\rangle$ is

$$|\psi(0)\rangle \stackrel{\text{def.}}{=} \sum_w \lambda_w |q_0; 0; w\rangle; \quad \sum_w \|\lambda_w\|^2 = 1$$

Checking the end of computation

We can use a special qubit denoting the end of computation to check whether the machine terminated or not.

Circuit diagram



Question: U unitary for a TM $\mathcal{M}$?

For any pure quantum state $|q;x;w\rangle$, U need to satisfy

$$\langle q';x';w'|U|q;x;w\rangle \\ = [\delta_{x',x+1}U^{+}(q',w'_x \mid q,w_x) + \delta_{x',x-1}U^{-}(q',w'_x \mid q,w_x)]\prod_{y\neq x}\delta_{w_y,w'_y}$$

where $U^{\pm}(q',\sigma' \mid q,\sigma)$ is a $\{0,1\}$-valued function:

$$U^{\pm}(q';\sigma' \mid q;\sigma) = \delta_{q',A(q,\sigma)} \cdot \delta_{\sigma',B(q,\sigma)} \cdot \frac{1}{2}[1 \pm C(q,w)]$$

Question: U unitary for a TM $\mathcal{M}$?

For any pure quantum state $|q; x; w\rangle$, U need to satisfy

$$\boxed{\langle q'; x'; w' | U | q; x; w \rangle} \leftarrow \boxed{\text{1 if the move is valid}}$$
$$= [\delta_{x',x+1} U^{+}(q', w'_x \mid q, w_x) + \delta_{x',x-1} U^{-}(q', w'_x \mid q, w_x)] \prod_{y \neq x} \delta_{w_y, w'_y}$$

where $U^{\pm}(q', \sigma' \mid q, \sigma)$ is a $\{0, 1\}$-valued function:

$$U^{\pm}(q'; \sigma' \mid q; \sigma) = \delta_{q', A(q,\sigma)} \cdot \delta_{\sigma', B(q,\sigma)} \cdot \frac{1}{2}[1 \pm C(q, w)]$$

Question: U unitary for a TM $\mathcal{M}$?

For any pure quantum state $|q; x; w\rangle$, U need to satisfy

$$\begin{aligned}&\langle q'; x'; w'|U|q; x; w\rangle \\ &\quad = [\underbrace{\boxed{\delta_{x',x+1} U^{+}(q', w'_x \mid q, w_x)}}_{+1 \text{ move}} + \underbrace{\boxed{\delta_{x',x-1} U^{-}(q', w'_x \mid q, w_x)}}_{-1 \text{ move}}] \prod_{y \neq x} \delta_{w_y, w'_y}\end{aligned}$$

where $U^{\pm}(q', \sigma' \mid q, \sigma)$ is a $\{0, 1\}$-valued function:

$$U^{\pm}(q'; \sigma' \mid q; \sigma) = \delta_{q', A(q,\sigma)} \cdot \delta_{\sigma', B(q,\sigma)} \cdot \frac{1}{2}[1 \pm C(q, w)]$$

Question: U unitary for a TM $\mathcal{M}$?

For any pure quantum state $|q;x;w\rangle$, U need to satisfy

Others not modified

$$\langle q';x';w'|U|q;x;w\rangle \\ = [\delta_{x',x+1}U^+(q',w'_x \mid q,w_x) + \delta_{x',x-1}U^-(q',w'_x \mid q,w_x)]\boxed{\prod_{y\neq x}\delta_{w_y,w'_y}}$$

where $U^\pm(q',\sigma' \mid q,\sigma)$ is a $\{0,1\}$-valued function:

$$U^\pm(q';\sigma' \mid q;\sigma) = \delta_{q',A(q,\sigma)} \cdot \delta_{\sigma',B(q,\sigma)} \cdot \frac{1}{2}[1 \pm C(q,w)]$$

Question: U unitary for a TM $\mathcal{M}$?

For any pure quantum state $|q;x;w\rangle$, U need to satisfy

$$
\begin{aligned}
&\langle q';x';w'|U|q;x;w\rangle \\
&\qquad = [\delta_{x',x+1}U^{+}(q',w'_x \mid q,w_x) + \delta_{x',x-1}U^{-}(q',w'_x \mid q,w_x)]\textstyle\prod_{y\neq x}\delta_{w_y,w'_y}
\end{aligned}
$$

where $U^{\pm}(q',\sigma' \mid q,\sigma)$ is a $\{0,1\}$-valued function:

$$U^{\pm}(q';\sigma' \mid q;\sigma) = \delta_{q',A(q,\sigma)} \cdot \delta_{\sigma',B(q,\sigma)} \cdot \frac{1}{2}[1 \pm C(q,w)]$$

Question: U unitary for a TM $\mathcal{M}$?

For any pure quantum state $|q;x;w\rangle$, U need to satisfy

$$\langle q';x';w'|U|q;x;w\rangle$$
$$= [\delta_{x',x+1}U^{+}(q',w'_x \mid q,w_x) + \delta_{x',x-1}U^{-}(q',w'_x \mid q,w_x)]\textstyle\prod_{y\neq x}\delta_{w_y,w'_y}$$

where $U^{\pm}(q',\sigma' \mid q,\sigma)$ is a $\{0,1\}$-valued function:

$$U^{\pm}(q';\sigma' \mid q;\sigma) = \underbrace{\boxed{\delta_{q',A(q,\sigma)}}}_{\text{Change in states}} \cdot \underbrace{\boxed{\delta_{\sigma',B(q,\sigma)}}}_{\text{Chance in symbol}} \cdot \overbrace{\boxed{\frac{1}{2}[1 \pm C(q,w)]}}^{\text{Change in head pos.}}$$

Question: U unitary for a TM $\mathcal{M}$?

$$U^{\pm}(q';\sigma' \mid q;\sigma) = \frac{1}{2}\delta_{q',A(q,\sigma)}\delta_{\sigma',B(q,\sigma)}[1 \pm C(q,w)],$$

Since every TM $\mathcal{M}$ has an equiv. *reversible* (deterministic) TM $\mathcal{M}^R = (Q^R, E^R, i^R, f^R)$, we design a QTM $\mathcal{Q}$ from $\mathcal{M}^R$ to recognize $L(\mathcal{M})$. U becomes unitary by setting A, B and C to satisfy $E^R(q,\sigma,A(q,\sigma),B(q,\sigma),C(q,\sigma)) = 1$.

$\therefore$ TM $\subseteq$ QTM

Since a TM can simulate a QTM, TM and QTM has the same power.

U construction

Let $n(q; x; w) : \mathcal{C} \to \mathbb{N}$ be a fixed numbering on the configurations. Then, for two configurations c_1 and c_2, $U_{n(c_2),n(c_1)} = 1$ iff $c_1 \vdash_{\mathcal{M}} c_2$.

QTM and quantum circuits

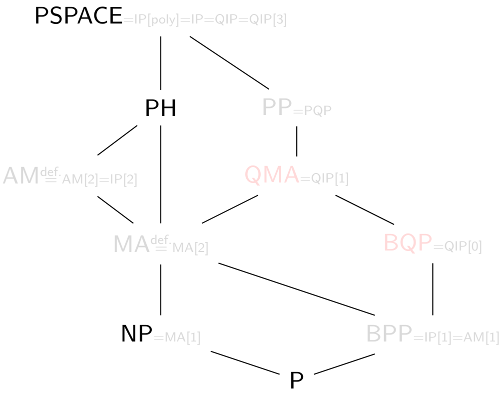
Remark

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 $|0\rangle$.

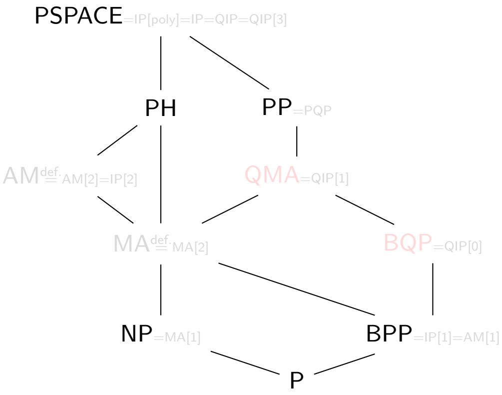
Overview

Complexity hierarchy
Overview



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

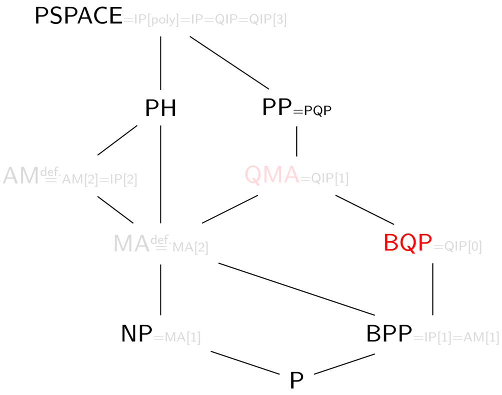
Complexity hierarchy
Overview



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

Complexity hierarchy

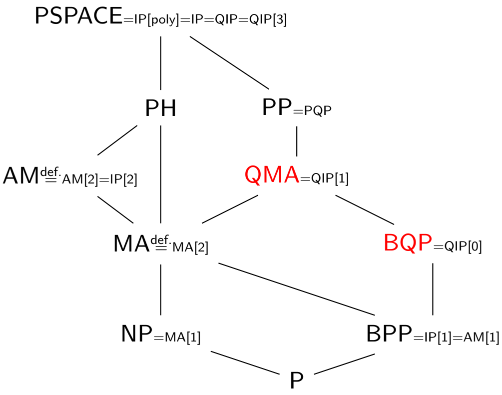
Overview



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

Complexity hierarchy

Overview



$\mathcal{C}[k]$: 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, $\exists \mathcal{Q} \exists c \forall w \ [c \in [0, 0.5) \land \Pr[\mathcal{Q}(\langle w \rangle) \neq 1(w \in L)] \leq c]$.

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

and other error-bounded complexities

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 $j \in [1, i]$ **do**
    collect $O(w)$
end for
return majority

Implication of BQP

and other error-bounded complexities

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

and other error-bounded complexities

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

An open problem

P $\neq$ PSPACE

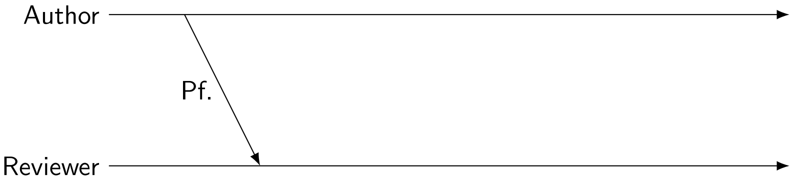
Author

Reviewer

Interactive proof system

An open problem

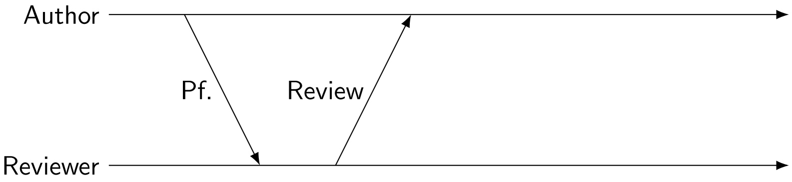
P $\neq$ PSPACE



Interactive proof system

An open problem

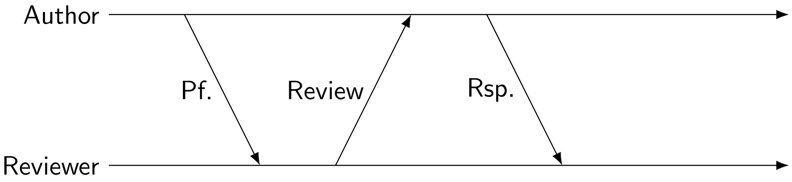
P $\neq$ PSPACE



Interactive proof system

An open problem

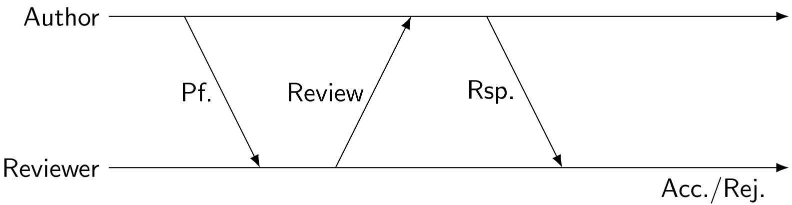
P $\neq$ PSPACE



Interactive proof system

An open problem

P $\neq$ PSPACE



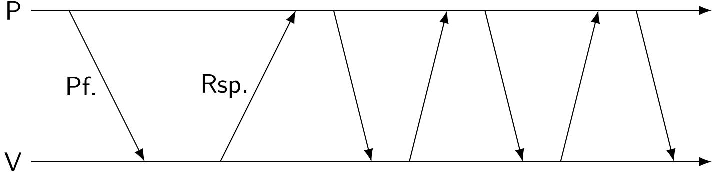
Interactive proof system

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

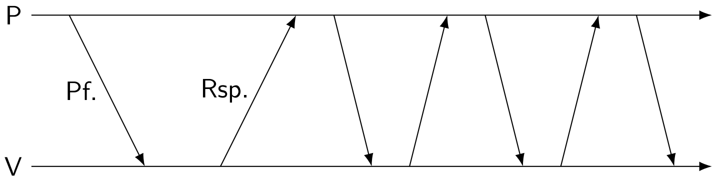
$\boxed{\text{P} \neq \text{NP},\ w \in L,\ \phi \text{ satisfiable, etc.}}$

Assume a 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 [Bab85] is another name of the interactive proof system, due to [GS86], 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

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

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

Example: Deutsch-Jozsa

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

Promise problem

Example: Deutsch-Jozsa

Promise $(L_1 \cup L_0)$

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

L_1

BQP-complete

Problem (Canonical PromiseBQP problem[Zha12])

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[Zha12])

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

Reduction

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

Reduction

Assume: an oracle O for the canonical problem.
For a PromiseBQP problem (L_1, L_0), ∃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

Other complete problems

From [Zha12] (BQP-c) and [Boo13] (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.