Towards a quantum-inspired proof for IP = PSPACE

Yupan Liu

Hebrew University of Jerusalem

Joint work with Ayal Green and Guy Kindler

YITP, Kyoto University, Oct 2019

Delegated computation by interactions

Let us start from a computationally hard problem:

Factoring

Input: $n, k \in \mathbb{N}$ (input size is $\log(n)$).

Output: YES if n has factor $< k$; otherwise NO.

What do we know about Factoring?

- Factoring $\in$ NP since we can multiply large numbers efficiently.
- Factoring $\in$ BQP [Shor94].

Here is a protocol to verify Factoring by interactions:

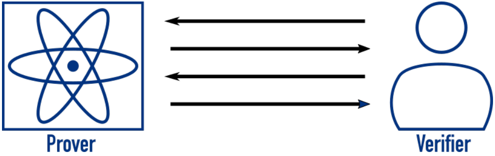
1. The verifier chooses two large number k_1, k_2, and sends n (which is $k_1 \times k_2$) and k_1 to the prover.
2. The prover answer YES if k_1 is a factor of n otherwise NO.

We can delegate a complicated computation using interactions!

An introduction to interactive proofs

Interactive proofs

Given a language $\mathcal{L} = (\mathcal{L}_{yes}, \mathcal{L}_{no})$, there is an interactive proof protocol with at most $\text{poly}(n)$ round interactions (using $\text{poly}(n)$-size *classical* messages) between a $\mathcal{P}$-power prover and a $\mathcal{V}$-power verifier.



$\mathsf{IP}[\mathcal{P}, \mathcal{V}]$ is the set of all languages which have such a protocol.

We usually assume that the power of verifier is BPP, namely all *probabilistic* polynomial-time computations. Examples:

- Factoring $\in \mathsf{IP}[\mathsf{NP}, \mathsf{BPP}]$
- Factoring $\in \mathsf{IP}[\mathsf{BQP}, \mathsf{BPP}]$
- $\mathsf{NP} \subseteq \mathsf{IP}[\mathsf{NP}, \mathsf{BPP}]$
- $\mathsf{BQP} \overset{?}{\subseteq} \mathsf{IP}[\mathsf{BQP}, \mathsf{BPP}]$ (open problem)

Could we think about delegated computation as interactive proofs?

Delegated computation, revisited

In-class interactive proofs

A class $\mathcal{P}$ has an in-class interactive proof if for any language $\mathcal{L}$ in $\mathcal{P}$, there is an interactive proof $\mathsf{IP}[\mathcal{P},\mathcal{V}]$ for $\mathcal{L}$. Denote by $\mathcal{P} = \mathsf{IP}[\mathcal{P},\mathcal{V}]$.

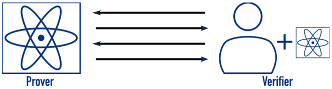
Which classes have delegated computation by interactive proofs?

- NP by simply by definition.
- $\mathsf{P}^{\#\mathsf{P}}$ [LFKN90, AG17] where $\#\mathsf{P}$ is the counting version of NP.
- $\mathsf{PSPACE} = \mathsf{IP}[\mathsf{PSPACE},\mathsf{BPP}]$ [Shamir90] where PSPACE is all computation can be done in polynomial space.
- $\mathsf{NC}(\mathsf{poly}) = \mathsf{IP}[\mathsf{NC}(\mathsf{poly}),\mathsf{BPP}]$ [GKR08] where $\mathsf{NC}(\mathsf{poly})$ is defined by poly-depth but exp-size Boolean circuits computation (upscaling version).

Even the prover is *all-powerful*, interactive proofs don't have more power ($\mathsf{IP} = \mathsf{PSPACE} = \mathsf{QIP}$ [Shamir90,JJUW09]). But *multi-prover* interactive proofs are more powerful, such as $\mathsf{MIP} = \mathsf{NEXP}$ [BFL91] and $\mathsf{NEEXP} \subseteq \mathsf{MIP}^*$ [NW19].

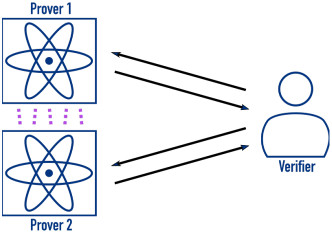
What about delegation of quantum computation?

Delegation of quantum computation

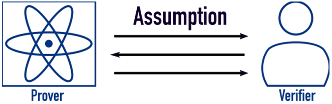


- A single quantum prover and a classical verifier with a small quantum device.

Ref BFK09,ABOE10/ABOEM17,FHM18.



- Multiple quantum provers with maximal entanglement, a classical verifier, poly-szie (i.e. $O(n^{8192})$) communication [RUV13].
- Two provers, one round, and quasi-linear size communication [CGJV19].



- A single quantum prover and a classical verifier, with a *computational* soundness.

Ref Mahadev18, CCKW18, GV19.

Besides, a few *subclasses* of BQP is in IP[BQP, BPP], such as MA $\cap$ BQP [MTN17] and computing the order of solvable groups [LGMNT18].

Quantum characterization of classical complexity classes

Starting from classes regarding precise quantum computation:

- PreciseBQP: Performing an *efficient* quantum computation within *inverse-exponential* accuracy.
- PreciseQMA: Given a *quantum* "proof" (i.e. witness), verifying an *efficient* quantum computation within *inverse-exponential* accuracy.

A few classical complexity classes have a quantum characterization:

- PreciseBQP $=$ PP [Aar05, Kup09, GSSSY18].
- PreciseQCMA $=$ NP^{PP} [MN17, GSSSY18].
- PreciseQMA $=$ PSPACE [FL16,FL18].

Delegation of precise quantum computation [AG17]

PreciseBQP $=$ IP[PreciseBQP, BPP], or a quantum-inspired proof for [LFKN90].

Q: Could we extend their protocol to PreciseQMA?

A: Partially YES! We provide an in-class interactive proof protocol for NP^{PP}.

An in-class interactive proof protocol for PreciseQCMA

Main result

PreciseQCMA $\subseteq$ IP[PreciseQCMA, BPP], namely $\mathsf{NP}^{\mathsf{PP}} \subseteq$ IP[$\mathsf{NP}^{\mathsf{PP}}$, BPP].

The protcol

For any language $\mathcal{L} \in$ PreciseQCMA, given an instance $x \in \mathcal{L}$, one can verify $\mathcal{L}$:

1. The verifier V sends the instance x (i.e. a problem) to the prover P.
2. The verifier V asks the prover P for a classical witness w of x.
3. The prover P and the verifier V follows an in-class interactive proof protocol W for PreciseBQP, and V accepts iff W accepts.

An explicit example:

1. A local Hamiltonian H which its ground states $|\Omega\rangle$ can be prepared in a polynomial-depth circuit within inverse-exponential accuracy.
2. The witness is an efficient PEPS representation of a ground state $|\Omega\rangle$.
3. Verifying the ground energy $\langle\Omega|H|\Omega\rangle$ of $|\Omega\rangle$ by contracting a tensor network.

Q: Is a PreciseQCMA-power prover powerful enough to find a *classical* witness?

Finding the classical witness by an adaptive search

We said that a prover has PreciseQCMA-power if this prover can access a PreciseQCMA oracle *polynomially* many times.

A witness-finding algorithm $\mathcal{A}$ for NP (i.e. search-to-decision reduction)

1. The prover P queries the oracle $\mathcal{O}$ whether the claim S_0, *"there exists a witness for the instance x where the first bit $b = 0$"*, is true or not.
2. If the answer is NO. The prover P queries the oracle $\mathcal{O}$ about S_0 where the value of the first bit b is flipped; otherwise, the first bit $b = 0$.
3. The prover P can find the first bit b of the witness, and P can find a witness by querying statements S_{b0} adaptively for all bits. Namely, repeating first two steps for each bit in the witness.

Indeed, the witness-finding algorithm $\mathcal{A}$ works for NP $\subseteq$ PreciseQCMA. Does such an algorithm work for PreciseQCMA?

Why the witness-finding algorithm works for PreciseQCMA?

To prove that the witness-finding algorithm works for NP, it is enough to show that the language $\hat{\mathcal{L}}$ associated with the witness-finding algorithm is in NP.

- A language $\mathcal{L} = (\mathcal{L}_{yes}, \mathcal{L}_{no}) \in$ NP if there is an efficient classical verifier $V_{\mathcal{L}}$ where $\mathcal{L}_{yes} = \{x | \exists w \text{ s.t. } V_{\mathcal{L}}(x,w) = 1\}$, $\mathcal{L}_{no} = \{x | \forall w, V_{\mathcal{L}}(x,w) = 0\}$.
- The language $\hat{\mathcal{L}}$ describes all (instance, partial witness) pairs, which can be found by the witness-finding algorithm $\mathcal{A}$ given a verifier $V_{\mathcal{L}}$, is defined by $\hat{\mathcal{L}} := \{(x, w_0) | \exists w_1 \text{ s.t. } V_{\mathcal{L}}(x, w_0 \circ w_1) = 1\}$, where w_0 is a prefix of a correct witness.
- It is easy to see that $(\hat{\mathcal{L}}, \{0,1\}^* \setminus \hat{\mathcal{L}}) \in$ NP.

What about PreciseQCMA?

Why the witness-finding algorithm works for PreciseQCMA? (Cont.)

The language of partial witnesses for PreciseQCMA

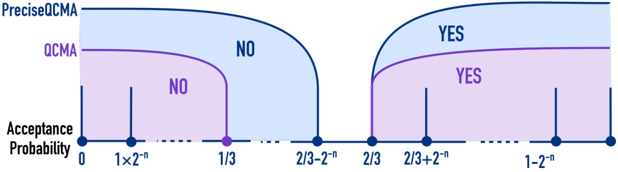
Given a (c,s)-PreciseQCMA verifier $V_{\mathcal{L}}$, one can define $\hat{\mathcal{L}}'$ similarly,

$$\hat{\mathcal{L}}' := \{(x,w_0) | \exists w_1 \text{ s.t. } \Pr[V_{\mathcal{L}}|x\rangle|w_0 \circ w_1\rangle = |\text{Acc}\rangle] \geq c\}.$$

It implies that $\{0,1\}^* \setminus \hat{\mathcal{L}}' = \{(x,w_0) | \forall w_1, \Pr[V_{\mathcal{L}}|x\rangle|w_0 \circ w_1\rangle = |\text{Acc}\rangle] \leq c - \delta\}$, where δ is the accuracy required of the acceptance probability.

Would δ be *arbitrarily* small? Thanks to the lemma below, δ is only *exponentially* small, which means that $(\hat{L}', \{0,1\}^* \setminus \hat{\mathcal{L}}') \in$ PreciseQCMA.

Lemma The acceptance probability of $x \in \mathcal{L}$ where $\mathcal{L} \in$ PreciseQCMA locates on an inverse-exponentially-separated grid.



In-class interactive proofs for PreciseBQP

Q-CIRCUIT problem

Approximating an amplitude $\langle 0^n|U|0^n\rangle$ of a polynomial-size quantum circuit U (i.e. U consists of $\mathrm{poly}(n)$ local gates) on n qubits within inverse-exponential accuracy is PreciseBQP-complete.

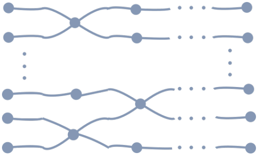
To show that PreciseBQP $\subseteq$ IP[PreciseBQP, BPP], it is enough to find an approach to verify $\langle 0^n|U|0^n\rangle \approx_\epsilon C$ where $\epsilon = \exp(-\mathrm{poly}(n))$.

- Preparing a poly-size quantum circuit U is *not necessarily* in classical polynomial-time.
- Using the correspondence between degree-3 polynomials and quantum circuits [Montanaro17], an amplitude $\langle 0^n|U|0^n\rangle$ can be converted into a $\#$SAT instance, then it follows from the original sum-check [LFKN90].

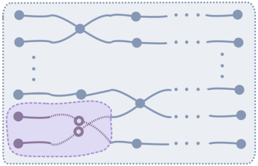
[AG17] provides a *structure-preserving* in-class interactive proof for PreciseBQP. In some sense, it reinterprets the sum-check from a *tensor-network contraction* perspective.

AG protocol

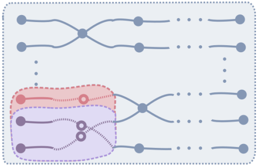
1. The verifier V sends a gate sequence associated with U which consists of $\mathrm{poly}(n)$ local quantum gates.



2.
   - V replaces a two-qubit gate U_1 by two single-qubit random rotations $V_1^{(1)} \otimes V_1^{(2)}$.
   - V asks P for a small tensor M_1, receive M_1'.
   - V rejects if $\mathrm{contract}(M_1', U_1) \neq_\epsilon C$.



3.
   - V replaces a single-qubit gate U_2 by a single-qubit random rotations V_2.
   - V asks P for a small tensor M_2 and receive M_2'
   - V rejects if $\mathrm{contract}(M_2', U_2, V_1^{(1)} \otimes V_1^{(2)}) \neq_\epsilon \mathrm{contract}(M_1', V_1^{(1)} \otimes V_1^{(2)})$.



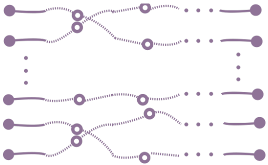
⋮ Repeat the third round for i-th $(3 \leq i \leq T-1)$ local gate in the given gate sequence associated with U.

AG protocol (Cont.)

Now is the final round of the AG protocol..

(T+1)

- V replaces a local gate U_T by a tensor product of single-qubit random rotations.
- V rejects if $\mathrm{contract}(M'_{T-1}, V_T^{(1)} \otimes V_T^{(2)}) \neq_\epsilon \mathrm{contract}(V_1^{(1)} \otimes V_1^{(2)}, V_2, \cdots, V_T^{(1)} \otimes V_T^{(2)})$.
- Otherwise V accepts.



Completeness

- At the i-th $(2 \leq i \leq T+1)$ round, the prover P can compute the small tensor M_{i-1} since contracting a tensor network defined on an arbitrary graph is in $\#$P [SWVC06,AL08].
- At the $(T+1)$-th round, notice that the tensor network here only consists of *strands* and *loops* which its bond dimension is *constant*, the verifier V can compute $\mathrm{contract}(V_1^{(1)} \otimes V_1^{(2)}, V_2, \cdots, V_T^{(1)} \otimes V_T^{(2)})$.

AG protocol: Soundness

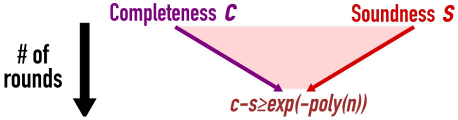
Soundness (unlimited precision)

One can show that both cases below are impossible by a direct calculation:

- A cheating prover passes on the round associated with the i-th gate with $M_{i-1} - M'_{i-1} \neq 0$ and $M_i - M'_i = 0$.
- A cheating prover passes on the round ascsoiate with the T-th gate with $M_{T-1} - M'_{T-1} \neq 0$.

Soundness

To prevent from a cheating prover, the required accuracy of $\langle 0^n | U | 0^n \rangle$ decays *exponentially* on the number of rounds (Claim 6.2 in [AG17]).



- A similar behavior also appears in [LFKN90].

Discussion: Towards an in-class interactive proof for PreciseQMA

Extending the protocol for PreciseQCMA

- Sending quantum witness directly requires *exponential-bit* communication.
- For *quantum* interactive proofs, it is not known how to achieve *inverse exponential accuracy* without *exponentially many* copies of the witness.

$\mathsf{QMA} \subseteq \mathsf{IP}[\mathsf{PreciseBQP}, \mathsf{BPP}]$

- The witness-preserving gap amplification for QMA [MW05, NWZ09] deduces an efficient quantum circuit U_V associated a QMA verifier V.
- One can verify any QMA computation by verifying a circuit amplitude $\langle 0^n | U_V | 0^n \rangle$ within inverse-exponential accuracy.

Extending the protocol for QMA

- It fails for PreciseQCMA since such an amplification deduces an *exponential-size* quantum circuit due to the inverse-exponential gap $c - s$.
- PostQMA [MN17] seems avoid this issue due to a *constant gap* $c - s$. However, witness-preserving gap amplification for PostQMA is unknown since its acceptance probability described by *conditional probability*.

Thank you!