Interactive Proofs for Synthesizing Quantum States and Unitaries

Gregory Rosenthal\textsuperscript{1}  Henry Yuen\textsuperscript{2}

\textsuperscript{1}University of Toronto
\textsuperscript{2}Columbia University

ITCS 2022
State & unitary synthesis

- **State synthesis**: Construct a (succinctly described) quantum state.
  - E.g. quantum money, quantum PRS, ...

- **Unitary synthesis**: Apply a (succinctly described) unitary transformation to a given input register.
  - E.g. variational quantum eigensolvers, decoders for quantum error-correcting codes, ...

- Poorly understood compared to decision problems.
Why state & unitary synthesis seems hard

Quantum analogue of function problems, but

- No clear reduction to decision problems.
  - Whereas computing a string reduces to computing each bit individually.

- An $n$-qubit state $\sum_{x \in \{0,1\}^n} \alpha_x |x\rangle$ has $2^n$ amplitudes.

- For unitary synthesis, since the input state is unknown, it’s impossible to describe the output state.
Our contributions

- Progress toward “IP = PSPACE for quantum states & unitaries”:
  - statePSPACE $\subseteq$ stateQIP $\subseteq$ stateEXP.
  - special case of unitaryPSPACE $\subseteq$ unitaryQIP.

- Definitions of these classes.
- Similar results with multiple entangled provers.
- (Proofs nontrivially reduce to QIP = PSPACE [JJUW’11] and MIP* = RE [JNVWY’20].)
Our contributions

- Progress toward “IP = PSPACE for quantum states & unitaries”:
  - $\text{statePSPACE} \subseteq \text{stateQIP} \subseteq \text{stateEXP}$.
  - Special case of $\text{unitaryPSPACE} \subseteq \text{unitaryQIP}$.
- Definitions of these classes.
- Similar results with multiple entangled provers.
- (Proofs nontrivially reduce to QIP = PSPACE [JJUW’11] and MIP* = RE [JNVWY’20].)
BQP verifier does the following:

- Interact with an untrusted quantum prover (quantum messages, polynomially many rounds).
- Accept or reject.
- If accepting, also output a quantum state.

(Like QIP except the last step.)
Interactive state & unitary synthesis (2/2)

- **Completeness:** There exists an “honest” prover strategy such that with probability 1, the verifier accepts and the output state is $\approx$ correct.

- **Soundness:** For all prover strategies such that the verifier accepts with non-negligible probability, the output state conditioned on accepting is $\approx$ correct.
Interactive state synthesis

- **Completeness:** There exists an “honest” prover strategy such that with probability 1, the verifier accepts and the output state is correct to within $\exp(-\text{poly}(n))$ trace distance error.

- **Soundness:** For all prover strategies such that the verifier accepts with probability $\geq \exp(-\text{poly}(n))$, the output state conditioned on accepting is correct to within $1/\text{poly}(n)$ t.d. error.
Interactive unitary synthesis

- **Completeness**: There exists an “honest” prover strategy such that with probability 1, the verifier accepts and the output state is correct to within $1/\text{poly}(n)$ trace distance error.

- **Soundness**: For all prover strategies such that the verifier accepts with probability $\geq \exp(-\text{poly}(n))$, the output state conditioned on accepting is correct to within $1/\text{poly}(n)$ t.d. error.
State & unitary complexity classes

- \text{stateQIP} = \text{sequences } (|\psi_n\rangle)_n \text{ with } |\psi_n\rangle \text{ on } n \text{ qubits that can be synthesized as above.}
  - More generally, could consider \((|\psi_x\rangle)_{x \in \{0,1\}^*}\).
- \text{unitaryQIP} = \text{sequences } (U_n)_n \text{ with } U_n \text{ acting on } n \text{ qubits that can be synthesized as above.}
- \text{statePSPACE} = \text{sequences } (|\psi_n\rangle)_n \text{ with } |\psi_n\rangle \text{ on } n \text{ qubits that can be } \approx \text{ constructed in quantum poly}(n) \text{ space.}
- \text{unitaryPSPACE} = \text{defined similarly.}
Quantum polynomial space

\((C_n)_n\) is a family of quantum polynomial-space circuits if

- There is a PSPACE machine that on input \(1^n\) outputs the description of \(C_n\).
- \(C_n\) consists of the following operations:
  - one- and two-qubit gates from a universal gate set,
  - standard-basis measurements,
  - tracing out qubits,
  - introducing new qubits (initialized to \(|0\rangle\)).
- \(C_n\) uses at most \(\text{poly}(n)\) qubits at any point.
Our contributions

▶ Progress toward “IP = PSPACE for quantum states & unitaries”:
  ▶ statePSPACE ⊆ stateQIP ⊆ stateEXP.
  ▶ special case of unitaryPSPACE ⊆ unitaryQIP.
▶ Definitions of these classes.
▶ Similar results with multiple entangled provers.
▶ (Proofs nontrivially reduce to QIP = PSPACE [JJUW’11] and MIP* = RE [JNVWY’20].)
State synthesis with a trusted prover [Aaronson’16]

- Write the target state as $|\psi\rangle = \sum_{i=0}^{1} \beta_i |i\rangle |\theta_i\rangle$.
- Query $(\beta_0, \beta_1)$ to finite precision.
- Construct $\beta_0 |0\rangle + \beta_1 |1\rangle$ in a register R.
- Uncompute $(\beta_0, \beta_1)$.
- Controlled on the bit $i$ in R, recursively construct $|\theta_i\rangle$.

Why do we uncompute $(\beta_0, \beta_1)$?

- Otherwise instead of constructing $|\psi\rangle = \sum_{x \in \{0, 1\}^n} \alpha_x |x\rangle$ we’d construct $\sum_{x \in \{0, 1\}^n} \alpha_x |x\rangle |\text{garbage}_x\rangle$. 
First attempt at state synthesis with an *untrusted* prover

- For statePSPACE states, the queries from the trusted-prover protocol are computable in PSPACE.
  - Follows from PSPACE = BQPSPACE [Watrous’03] and quantum state tomography.
- Idea: run the trusted-prover protocol & answer the queries using IP = PSPACE (in superposition).
- However the prover might not uncompute honestly.
  - E.g. if the target state is $|\psi\rangle = \sum_{x\in\{0,1\}^n} \alpha_x |x\rangle$, the verifier might output the first $n$ qubits of $\sum_{x\in\{0,1\}^n} \alpha_x |x\rangle|\phi_x\rangle$ for some state $|\phi_x\rangle$ held by the prover.
The actual protocol (1/3)

- Notation: for $0 \leq k \leq n$ let $|\psi_k\rangle$ denote the $k$-qubit state after $k$ iterations of the trusted-prover protocol.

- Given two copies of $|\psi_k\rangle$, “Copy 1” and “Copy 2”, the verifier obtains two copies of $|\psi_{k+1}\rangle$ as follows:
  - Flip a coin. If heads:
    - [Should yield two copies of $|\psi_{k+1}\rangle$.]
  - If tails:
    - [Should maintain the two copies of $|\psi_k\rangle$; the point is to detect cheating.]
    - Flip another coin.
The actual protocol (2/3)

- If heads:
  - Simulate a round of the trusted-prover protocol on Copy 1 (should yield $|\psi_{k+1}\rangle$).
  - Request a second copy of $|\psi_{k+1}\rangle$ from the prover.
  - Swap test to ensure these are the same state.

- If tails:
  - Simulate a round of the trusted-prover protocol on Copy 1, *minus the private step that grows the state by a qubit* (should yield $|\psi_k\rangle$).
  - Swap test with Copy 2 to ensure it’s actually $|\psi_k\rangle$.
  - Flip another coin.
State synthesis with a trusted prover [Aaronson’16]

- Write the target state as $|\psi\rangle = \sum_{i=0}^{1} \beta_i |i\rangle |\theta_i\rangle$.
- Query $(\beta_0, \beta_1)$ to finite precision.
- Construct $\beta_0 |0\rangle + \beta_1 |1\rangle$ in a register R.
- Uncompute $(\beta_0, \beta_1)$.
- Controlled on the bit $i$ in R, recursively construct $|\theta_i\rangle$.

Why do we uncompute $(\beta_0, \beta_1)$?

- Otherwise instead of constructing $|\psi\rangle = \sum_{x\in\{0,1\}^n} \alpha_x |x\rangle$ we’d construct $\sum_{x\in\{0,1\}^n} \alpha_x |x\rangle |\text{garbage}_x\rangle$. 
The actual protocol (2/3)

- If heads:
  - Simulate a round of the trusted-prover protocol on Copy 1 (should yield $|\psi_{k+1}\rangle$).
  - Request a second copy of $|\psi_{k+1}\rangle$ from the prover.
  - Swap test to ensure these are the same state.

- If tails:
  - Simulate a round of the trusted-prover protocol on Copy 1, *minus the private step that grows the state by a qubit* (should yield $|\psi_k\rangle$).
  - Swap test with Copy 2 to ensure it’s actually $|\psi_k\rangle$.
  - Flip another coin.
The actual protocol (3/3)

Soundness amplification:

▶ Execute the above protocol \(\text{poly}(n)\) times.
▶ If any execution rejects, then reject.
▶ Otherwise, accept and output the output state of a uniform random one of these executions.
Our contributions

▶ Progress toward “IP = PSPACE for quantum states & unitaries”:
   ▶ statePSPACE ⊆ stateQIP ⊆ stateEXP.
   ▶ special case of unitaryPSPACE ⊆ unitaryQIP.
▶ Definitions of these classes.
▶ Similar results with multiple entangled provers.
▶ (Proofs nontrivially reduce to QIP = PSPACE [JJUW’11] and MIP* = RE [JNVWY’20].)
\( \text{stateQIP} \subseteq \text{stateEXP} \)

- Find an \( \approx \) honest prover by optimizing over an SDP.
  - The SDP variables are the density matrices held by the verifier at the beginning/end of each round.
  - Constraints describe start state, transitions between rounds, end state accepted w.h.p.
  - Like [KW’00]’s original proof of QIP \( \subseteq \) EXP.

- Simulate the stateQIP protocol with that prover.
An $n$-qubit unitary $U$ has polynomial action if $U$ acts nontrivially on a subspace of dimension at most $\text{poly}(n)$.

Use [LMR'14]'s Hamiltonian simulation algorithm and $\text{statePSPACE} \subseteq \text{stateQIP}$, i.e.

- If $U = \exp(it\rho)$ then a purification of $\rho$ is in $\text{statePSPACE}$.
- Evolution time $t$ is computable in $\text{PSPACE} = \text{QIP}$.

Polynomial-action assumption $\Rightarrow t \leq \text{poly}(n) \Rightarrow$ at most $\text{poly}(n)$ copies of $\rho$ required.
Multiple entangled provers

- \( \text{stateR} = \) sequences \((|\psi_n\rangle)_n\) with \(|\psi_n\rangle\) on \(n\) qubits such that a description of \(\approx |\psi_n\rangle\) is computable as a function of \(n\).

- \( \text{stateR} = \text{stateQMIP} \).
  - \( \subseteq \): like the proof of \(\text{statePSPACE} \subseteq \text{stateQIP}\) but using \(\text{MIP}^* = \text{RE}\).
  - \( \supseteq \): brute-force over provers, which terminates because an honest prover exists.
    - (Whereas for \(L \in \text{MIP}^*\) and \(x \notin L\), the search fails to terminate on input \(x\).)

- “polynomial-action unitaryR” \(\subseteq \text{unitaryQMIP}\).
Open problems

- \(\text{stateQIP} \subseteq \text{statePSPACE}\)?
- Improve \(1/\text{poly}(n)\) errors in some of our results to \(\exp(-\text{poly}(n))\).
- Reduce the number of rounds.
  - We conjecture that a particular constant-round variant of our protocol works.
- \(\text{unitaryPSPACE} \subseteq \text{unitaryQIP}\)?
- Synthesis of mixed states?
- State/unitary synthesis with efficient provers?
- Multiple unentangled provers?
- Zero-knowledge? Crypto applications?