# Efficient Quantum State Synthesis with One Query

Gregory Rosenthal
University of {Cambridge, Warwick}

SODA 2024

# Computation reduces to decision problems

- $f : \{0,1\}^n \to \{0,1\}^m$ is $m$ decision problems.
- Or one quantum query to $g : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}$, $g(x,r) = \langle f(x), r\rangle_{\mathbb{F}_2}$ [BV97].
- Search, sampling, etc. reduce to functions.
- This talk: what about constructing quantum states?

# State synthesis

- Goal: algorithm $A$ making quantum queries to a boolean function, such that $\forall|\psi\rangle : \exists f : A^f$ maps $|0\rangle$ to $\approx |\psi\rangle$.

| | | |
|---|---|---|
| Clean solution | $\vert\psi\rangle\vert 0\rangle$ | 😊 |
| Non-clean solution | $\vert\psi\rangle\vert\text{garbage}_\psi\rangle$ | 🙂 |

# State synthesis algorithms

# Exponential time (trivial)

- Query the description of $|\psi\rangle$, then construct it.
- For a clean construction, uncompute the description with a second query.

# Polynomial time [Z98,KM01,GR02,A16]

1. Write $|\psi\rangle = \alpha_0|0\rangle|\psi_0\rangle + \alpha_1|1\rangle|\psi_1\rangle$.
2. Query $\alpha_0, \alpha_1$ to finite precision.
3. Construct $\alpha_0|0\rangle + \alpha_1|1\rangle$.
4. Controlled on $b \in \{0,1\}$, recursively construct $|\psi_b\rangle$.
5. Uncompute $\alpha_0, \alpha_1$.

- Problem: for some applications we want $O(1)$ queries.

# Polynomial space, $O(1)$ queries [INNRY22]

- $\exists$ nonuniform poly($n$)-qubit circuit $C_n$ of size $2^{\mathrm{poly}(n)}$ making 1 (resp. 2) queries:
- $\forall$ $n$-qubit states $|\psi\rangle$:
- $\exists$ $f$:
- $C_n^f$ non-cleanly (resp. cleanly) constructs $|\psi\rangle$ to within error $1/\mathrm{poly}(n)$ (resp. $2^{-\mathrm{poly}(n)}$).

# Polynomial time, $O(1)$ queries

- $\exists$ uniform poly($n$)-size circuit $C_n$ making 1 (resp. 4) queries:
- $\forall$ $n$-qubit states $|\psi\rangle$:
- $\exists$ $f$ depending explicitly on $|\psi\rangle$:
- $C_n^f$ non-cleanly (resp. cleanly) constructs $|\psi\rangle$ to within error $2^{-\mathrm{poly}(n)}$.

# Comparison of state synthesis algorithms

<table>
<tr><th>Algorithm</th><th>Queries</th><th>Size</th><th>Space</th><th>Error</th><th>Uniform</th><th>Clean</th></tr>
<tr><td rowspan="2">Trivial</td><td>1</td><td rowspan="2">exp</td><td rowspan="2">exp</td><td rowspan="2">1/exp</td><td rowspan="2">yes</td><td>no</td></tr>
<tr><td>2</td><td>yes</td></tr>
<tr><td>[A16]</td><td>poly</td><td>poly</td><td>poly</td><td>1/exp</td><td>yes</td><td>yes</td></tr>
<tr><td rowspan="2">[INNRY22]</td><td>1</td><td rowspan="2">exp</td><td rowspan="2">poly</td><td>1/poly</td><td rowspan="2">no</td><td>no</td></tr>
<tr><td>2</td><td>1/exp</td><td>yes</td></tr>
<tr><td rowspan="2">This paper</td><td>1</td><td rowspan="2">poly</td><td rowspan="2">poly</td><td rowspan="2">1/exp</td><td rowspan="2">yes</td><td>no</td></tr>
<tr><td>4</td><td>yes</td></tr>
</table>

# Proof sketch

# Constant-error solution [INNRY22]

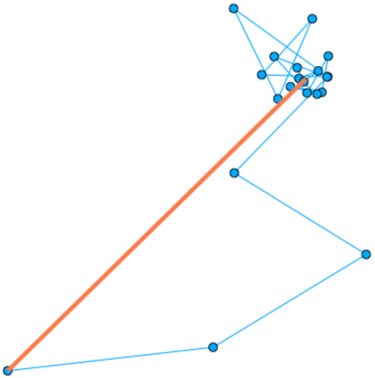
- $\forall|\psi\rangle : \exists$ Clifford $C$: $\left|\langle\psi| \cdot C \sum_{x\in\{0,1\}^n} \pm 2^{-n/2}|x\rangle\right| \geq \Omega(1)$.
- Intuition: Cliffords are a 2-design and Haar random states have high $\ell_1$ norm.
- Query maps $x \in \{0,1\}^n$ to sign bit and description of $C$.

# Linear Combinations of Unitaries (LCU) [CW12]

- Assume query access to unitaries $U_j$.
- Let $M = \sum_j c_j U_j$.
- Can implement $|\psi\rangle \mapsto M|\psi\rangle / \|M|\psi\rangle\|$ with success probability $\left(\|M|\psi\rangle\| / \sum_j |c_j|\right)^2$.

# Solution with constant success probability

- $|\psi\rangle \approx \sum_{j=0}^{\mathrm{poly}(n)} \alpha\beta^j |\phi_j\rangle$ where $|\phi_j\rangle$ is a "Clifford times phase state" and $0 < \alpha, \beta < 1$ are universal constants.



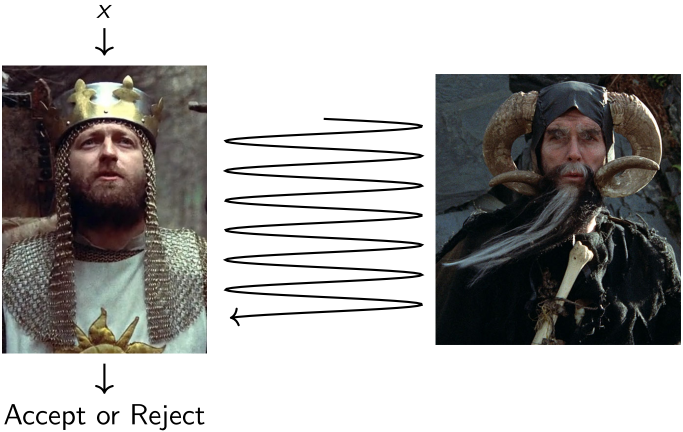
- Do LCU.

## Boosting the success probability

- Parallel repetition and merge queries $\implies$ 1 query, non-clean.
- Amplitude amplification $\implies$ $O(1)$ queries, clean.
- Hybrid approach $\implies$ 4 queries, clean.

$$\mathsf{stateQIP}(6) = \mathsf{statePSPACE}$$

# Interactive proof for a language $L$

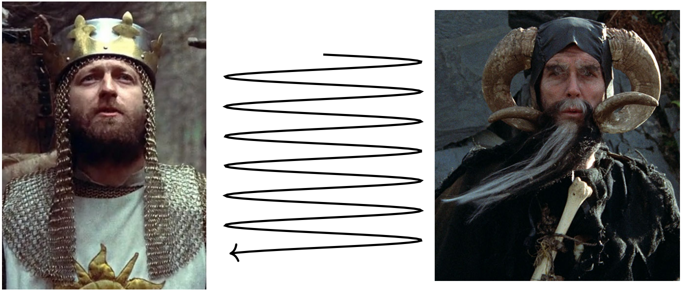



- *Completeness*: $x \in L \implies \exists$ prover s.t. Verifier accepts.
- *Soundness*: $x \notin L \implies \forall$ provers, Verifier rejects w.h.p.

# How powerful are interactive proofs?

- IP = languages with interactive proofs.
- = PSPACE (i.e. polynomial space) [LFKN92,S92].
- = QIP (i.e. IP with a quantum verifier) [JJUW11].
- = QIP(3) (i.e. QIP with three messages) [W03].

# Interactive proof for constructing a state $\rho$ [RY22]



$\downarrow$

(Accept, $\tilde{\rho}$) or Reject

- *Completeness*: $\exists$ prover s.t. Verifier accepts.
- *Soundness*: $\forall$ provers s.t. w.p. $\geq 1/\text{poly}(n)$ Verifier accepts, $\|\tilde{\rho} - \rho\|_{\text{tr}} \leq 1/\text{poly}(n)$.

# stateQIP = statePSPACE

- stateQIP = state sequences with interactive proofs.
- statePSPACE = quantum state analogue of PSPACE.
- statePSPACE $\subseteq$ stateQIP [RY22]:
  - Polynomial-time state synthesis [A16].
  - Answer queries using IP=PSPACE in superposition.
  - Additional steps to uncompute entangled garbage.
- stateQIP $\subseteq$ statePSPACE [MY23].

# statePSPACE $\subseteq$ stateQIP(6)

- stateQIP(6) $=$ six-message stateQIP.
- Follows from PSPACE $\subseteq$ QIP(3) [W03] and polynomial-time, one-query state synthesis.

$x$ (in superposition) $\longrightarrow$

QIP(3) $=$ PSPACE on input $x$ $\longleftarrow$

Uncompute $|\text{garbage}_x\rangle$ $\longleftarrow$

# Barrier to $\mathrm{QAC}_\mathrm{f}^0$ lower bounds for approximately constructing explicit states

# Circuit lower bounds for explicit states

- Exponential-size lower bounds for *exact* constructions [JW23].
- Trivial $\mathsf{QNC}^0$ lower bounds for *approximate* constructions.
- Why can't we prove *nontrivial* lower bounds for *approximate* constructions?

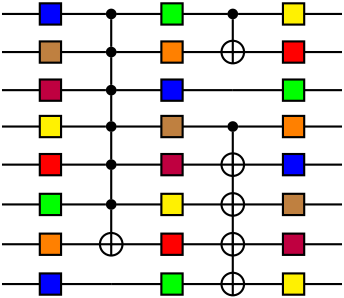
# Barrier [A16]

- Assume $|\psi\rangle$ cannot be (approximately) constructed by a poly-size circuit.
- $A \leftarrow$ poly-time state synthesis algorithm [A16].
- $f \leftarrow$ function such that $A^f$ constructs $|\psi\rangle$.
- $f \notin \mathsf{BQP/poly}$ because otherwise $A^f$ would be a poly-size circuit for constructing $|\psi\rangle$.
- This would be a huge breakthrough.

...But what about in weaker quantum circuit classes?

# $\mathsf{QAC}_\mathsf{f}^0$

- Polynomial-size, constant-depth with one-qubit gates and unbounded-arity AND, OR and FANOUT gates.
- FANOUT$\left|b, 0^{n-1}\right\rangle = |b^n\rangle$ for $b \in \{0, 1\}$.



- Physically motivated [GKHMDBC21,GDCEBDSCG22].

# Barrier to $\mathsf{QAC}^0_\mathsf{f}$ lower bounds for explicit states

- Clifford unitaries are in $\mathsf{QAC}^0_\mathsf{f}$ [$\sim$AG04].
- $\implies$ This paper's state synthesis algorithm is in $\mathsf{QAC}^0_\mathsf{f}$.
- $\implies$ $\mathsf{QAC}^0_\mathsf{f}$ lower bounds for explicit states imply $\mathsf{QAC}^0_\mathsf{f}$ lower bounds for explicit functions.
- $\mathsf{TC}^0 \subseteq \mathsf{QAC}^0_\mathsf{f}$ [HS05,TT16] and we don't have $\mathsf{TC}^0$ lower bounds for explicit functions.

# Circuit complexity of approximately constructing worst-case states

# Upper and lower bounds for constructing worst-case states

- $G \leftarrow$ universal gate set including AND, OR, NOT.
- Constructing worst-case $n$-qubit states to within error $\varepsilon \geq 2^{-\mathsf{poly}(n)}$ requires $G$-circuit size $\Theta(2^n \log(1/\varepsilon)/n)$.

- Worst-case $n$-qubit states require circuit size $\Theta(2^n)$ to *exactly* construct with *arbitrary* $O(1)$-qubit gates [ZLY22,GDASC23, STYYZ23,YZ23].

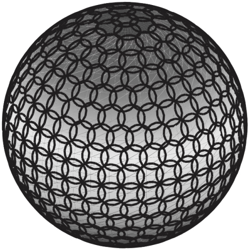
## Proof sketch

Upper bound:

- This paper's state synthesis algorithm.
- Simulate $m$-bit queries with $O(2^m/m)$-size circuits [L58].
- Solovay-Kitaev theorem on the non-query operations.

Lower bound:

- Counting argument.



# Open problems

# Generalization to unitaries?

- The "unitary synthesis problem": $\forall U : \exists f : U$ efficiently reduces to $f$ [AK07,A16]?
- $\tilde{O}(2^{n/2})$ queries & time suffices [R21].
- 1 query and $o(2^n)$ qubits does not suffice [LMW23].

# Search-to-decision reduction for QMA?

- SAT has efficient search-to-decision reductions.
- Constructing ground states of local Hamiltonians efficiently reduces to one quantum query to a PP oracle [INNRY22].
- What about to a QMA oracle?