Frontiers of Quantum Complexity and Cryptography Spring 2022

Lecture 6 – State and Unitary Synthesis

Lecturer: Henry Yuen *Scribes: Thomas Chen*

1 Overview

This lecture discussed the State Synthesis Problem (SSP) and Unitary Synthesis Problem (USP). The motivating questions are as follows.

(Q1) What is the complexity of synthesizing states and unitaries?

(Q2) Is there a difference in complexity of synthesizing states versus unitaries?

(Q3) How much of the complexity of these tasks is classical, versus that due to quantum aspects?

2 State Synthesis Problem

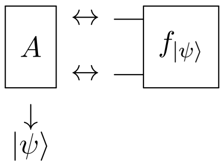
Recall from a few lectures ago our notion of the complexity of a quantum state:

Definition 1 (State complexity)**.** *The complexity, $C_\epsilon(|\psi\rangle)$, of a state $|\psi\rangle$ is the minimum size of a quantum circuit*[1] *that outputs $|\psi\rangle$ up to error ϵ.*

As discussed previously, a simple counting argument shows that for most n-qubit states $|\psi\rangle$, $C_\epsilon(|\psi\rangle) = \Omega(\exp(n))$. The same is true for the classical complexity of boolean functions, where most n-bit boolean functions $f : \{0,1\}^n \to \{0,1\}$ are such that $C(f) = \Omega(\exp(n))$, where $C(f)$ denotes the minimum size *classical* circuit required to compute f. This motivates the following natural question: can the complexity of synthesizing (i.e., generating) a quantum state be *reduced* to the complexity of computing a boolean function? This question is formalized as the *state synthesis problem* (abbreviated *SSP*):

State Synthesis Problem. Is there a quantum query algorithm A, a polynomial $p(n)$, and an encoding scheme that maps n-qubit states $|\psi\rangle$ to a function $f_\psi : \{0,1\}^{p(n)} \to \{0,1\}$ such that A makes *poly*(n) queries to f_ψ and outputs a good approximation of $|\psi\rangle$?

Diagrammatically, the task looks like the following:



[1] The measure of complexity implicitly depends on the choice of gate set, but for different universal gate sets the corresponding complexity measures are equivalent up to polylog(n) factors.

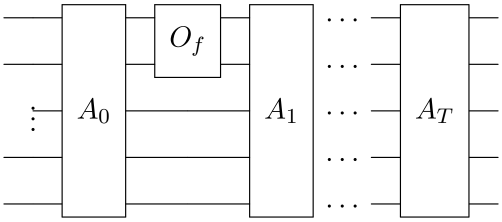
If the answer to this question is yes, then in this sense, quantum state synthesis is no harder than computing an appropriate boolean function.

2.1 Quantum Query Model

First, let's review the quantum query model. A T-query quantum algorithm A that queries a classical m-bit function $f : \{0,1\}^m \to \{0,1\}$ is a tuple of unitaries

$$A = (A_0, A_1, ...A_T)$$

where each unitary A_i acts on at least m qubits. The execution of the algorithm behaves as follows: starting with the all zeroes state ($|0 \cdots 0\rangle$), first the unitary A_0 is applied, then an *oracle* unitary O_f is applied to the first m qubits, then A_1 is applied, then an oracle unitary O_f, and so on, until the oracle unitary O_f has been called T times. This is depicted in the circuit diagram below.



There are two models for the oracle unitary O_f; one called the *phase oracle* and one called the *XOR oracle*. In the study of quantum algorithms these models are equivalent, but one may be more convenient to work with.

Phase oracle. This is a unitary O_f acting on m qubits such that for all $x \in \{0,1\}^m$,

$$O_f|x\rangle = (-1)^{f(x)}|x\rangle \ .$$

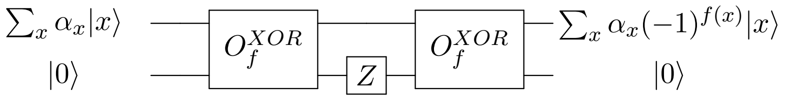
When applied to a superposition, we have

$$O_f \sum_x \alpha_x|x\rangle = \sum_x \alpha_x \, (-1)^{f(x)} \, |x\rangle \ .$$

XOR oracle. This is a unitary O_f acting on $m+1$ qubits such that for all $x \in \{0,1\}^m, b \in \{0,1\}$,

$$O_f|x, b\rangle = |x, b \oplus f(x)\rangle \ .$$

We can easily transform one of these oracles into the other. For this lecture, we just show how to transform an XOR oracle into a phase oracle.


$\sum_x \alpha_x|x\rangle$
$|0\rangle$
O_f^{XOR}
Z
O_f^{XOR}
$\sum_x \alpha_x(-1)^{f(x)}|x\rangle$
$|0\rangle$

To see that this indeed produces the claimed output, note that the state right after the first XOR oracle is

$$|\psi_1\rangle = \sum_x \alpha_x |x\rangle|f(x)\rangle$$

Applying the Z gate yields

$$|\psi_1\rangle = \sum_x \alpha_x (-1)^{f(x)} |x\rangle|f(x)\rangle$$

Reapplying the XOR oracle will reverse the original computation on the $m+1$th qubit, reverting the last qubit to $|0\rangle$. Thus we can simulate the phase oracle using two calls to the XOR oracle.

Remark: There is a more efficient transformation that uses just one call to an XOR oracle to produce a phase oracle. If we let the last qubit be a superposition,

$$O_f^{XOR}(|x\rangle|-\rangle) = \frac{1}{\sqrt{2}}(|x\rangle|f(x)\rangle + |x\rangle|1 \oplus f(x)\rangle) = (-1)^{f(x)}|x\rangle|-\rangle \ .$$

2.2 A solution to the State Synthesis Problem

We now discuss the following solution to the State Synthesis Problem presented by Aaronson [1].

Theorem 2. *There exists an $(n+1)$-query algorithm A and an encoding of states $|\psi\rangle$ into boolean functions $f_\psi : \{0,1\}^{poly(n)} \to \{0,1\}$ that solves the State Synthesis Problem. Moreover, the algorithm A is space efficient (the unitaries A_i act on $poly(n)$ qubits) and the non-oracle unitaries A_i are time-efficient (meaning that they can be implemented by $poly(n)$-size circuits). However, the oracle unitaries O_{f_ψ} will in general require exponential time to compute.*

Proof. The key idea of the proof is that a state can be decomposed into its *conditional amplitudes.* We let f hold all these conditional amplitudes, and we show an efficient procedure to reconstruct the state from its conditional amplitudes.

For the following state, the conditional amplitudes for the first qubit are $\frac{1}{\sqrt{2}}$ and $\frac{1}{\sqrt{2}}$, corresponding to when the qubit equals 0 and 1, respectively.

$$|\psi\rangle = \frac{1}{\sqrt{4}}|000\rangle - \frac{1}{\sqrt{4}}|010\rangle + \frac{1}{\sqrt{2}}|111\rangle = \frac{1}{\sqrt{2}}|0\rangle \otimes (\frac{1}{\sqrt{2}}|00\rangle - \frac{1}{\sqrt{2}}|10\rangle) + \frac{1}{\sqrt{2}}|1\rangle \otimes (|11\rangle) \ .$$

We can build a binary tree to hold all the conditional amplitudes, like in figure 1. The nodes correspond to the conditioned states– the left child of a node is the state obtained by conditioning the node state on the next qubit being 0, and the right child conditioning on 1. The edges of the tree are annotated with the conditional amplitudes. In general, the decomposition is as follows:

$$\begin{aligned}
|\psi\rangle &= \sum_{x\in\{0,1\}^n} \beta_x |x\rangle \\
&= \alpha_0|0\rangle|\psi_0\rangle + \alpha_1|1\rangle|\psi_1\rangle \qquad \cdots \\
&= \sum_{x\in\{0,1\}^n} \Big(\gamma_x \prod_{j=1}^{n} \alpha_{x_1\dots x_j}\Big)|x\rangle
\end{aligned}$$

Where $\alpha_{x_1...x_j}$ is the amplitude of the kets of $|\psi\rangle$ whose jth bit is x_j, conditioned on their first $j-1$ bits are $x_1...x_{j-1}$. We assume these conditional amplitudes are all real, while γ_x is a complex number on the unit circle. Note that we retrieve the original amplitude, β_x of $|x\rangle$ by multiplying all conditional amplitudes along the path to leaf $|x\rangle$, then multiplying by γ_x.

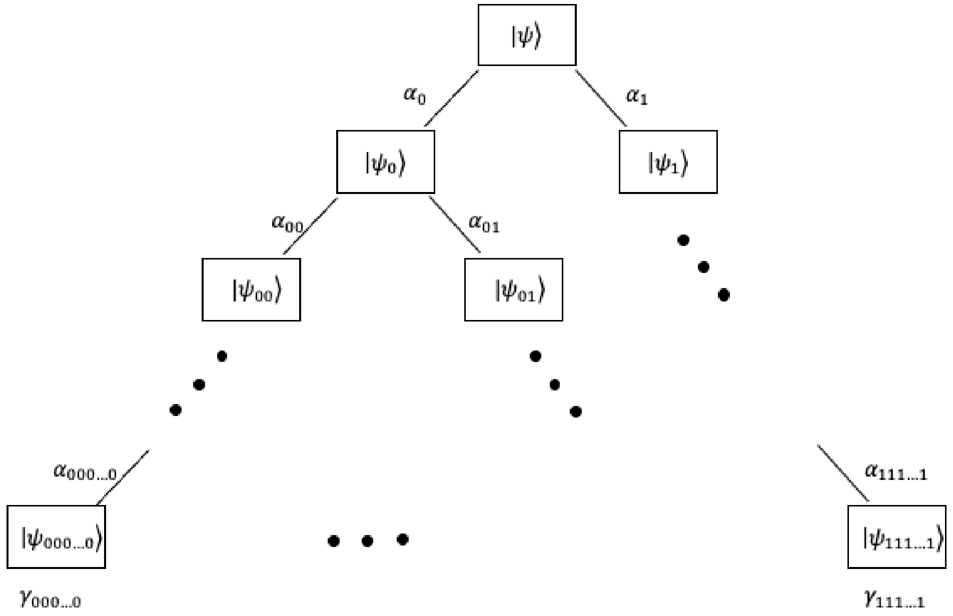


Figure 1: Decomposition of n-qubit state into its conditional amplitudes

Assume we are given oracle access to a classical function f that encodes the conditional amplitudes. The way we interact with this function is that when we query it, it writes a classical description of the conditional amplitudes we asked for in an ancilla qubit.

The following is an algorithm to iteratively reconstruct $|\psi\rangle$ from oracle calls to f.

(1) Ask for 1st pair of conditional amplitudes, (α_0, α_1) and prepare $\alpha_0|0\rangle + \alpha_1|1\rangle$

(2) Controlled on the first qubit in superposition, ask oracle for the conditional amplitudes corresponding to the left or right of the tree.

$$\alpha_0|0\rangle \otimes |“\alpha_{00}, \alpha''_{01}\rangle + \alpha_1|1\rangle \otimes |“\alpha_{10}, \alpha''_{11}\rangle$$

Here, the quotations denote that the information of the conditional amplitudes is written out classically. For each classical encoding of amplitudes, construct the corresponding qubit to obtain:

$$\alpha_0|0\rangle \otimes |“\alpha_{00}, \alpha''_{01}\rangle \otimes (\alpha_{00}|0\rangle + \alpha_{01}|1\rangle) + \alpha_1|1\rangle \otimes |“\alpha_{10}, \alpha''_{11}\rangle \otimes (\alpha_{10}|0\rangle + \alpha_{11}|1\rangle)$$

Finally, as in the phase oracle transformation, we uncompute the classical information to obtain

$$\alpha_0|0\rangle \otimes (\alpha_{00}|0\rangle + \alpha_{01}|1\rangle) + \alpha_1|1\rangle \otimes (\alpha_{10}|0\rangle + \alpha_{11}|1\rangle)$$

(2 to n) We can repeat this for the next batches of conditional amplitudes, so that by stage n, we have the state

$\sum_x \alpha_{x_1}\alpha_{x_1x_2}...\alpha_x|x\rangle$

($n+1$) We run the same procedure to call the oracle in superposition to obtain γ_x, generate the state $\sum_x \beta_x|x\rangle \otimes |\text{``}\gamma_x''\rangle$, then uncompute the classical information to obtain $\sum_x \beta_x|x\rangle$, as desired.

The correctness of the algorithm follows from the decomposition we can give of each quantum state into its conditional amplitudes. Moreover, the algorithm makes $n+1$ queries. A follow up question is whether we can do better than $n+1$ queries.

2.3 State synthesis algorithms with fewer queries

Can we do better than $n+1$ queries?

Theorem 3 (Irani, Natarajan, Nirkhe, Rao, Yuen [2])**.** *There exist solutions for the State Synthesis Problem that use only a constant number of queries. In particular:*

- *There exists a* 1 *query solution to SSP, but approximation error is* $O(\frac{1}{poly(n)})$*, algorithm runs in* $poly(n)$ *space, but is not time efficient. Unities* $\{A_0, A_1\}$ *require exponential amount of time to implement on a quantum computer.*
- *There is also a* 2 *query* $poly(n)$ *space algorithm with* $\exp(-\Omega(n))$ *error, also not time efficient.*

Open question. Can we make the algorithms from [2] time-efficient? Separately, can we improve the 1-query algorithm to have inverse exponential error (or prove some lower bounds)?

Remark. If we are allowed unlimited space, a 1-query solution to SSP is very easy! Essentially, the idea is to use one call to the oracle to extract the entire classical description of the desired quantum state. This uses a trick inspired by the *Bernstein-Vazirani algorithm.*

Consider a state $|\psi\rangle = \sum_x \alpha_x|x\rangle$. Let $m = 2^n \cdot \text{poly}(n)$, and let $Z_\psi \in \{0,1\}^m$ denote the classical description of $|\psi\rangle$ (in other words, one simply writes out all the amplitudes α_x in binary up to poly(n) precision). Now we can encode $|\psi\rangle$ into a classical function $f_\psi : \{0,1\}^m \to \{0,1\}$ as

$$f_\psi(y) = \langle y, Z_\psi\rangle \bmod 2$$

for all $y \in \{0,1\}^m$, where $\langle y, Z_\psi\rangle = y_1 Z_{\psi,1} + y_2 Z_{\psi,2} + \cdots + y_m Z_{\psi,m}$. In other words, f_ψ encodes every single inner product with the string Z_ψ.

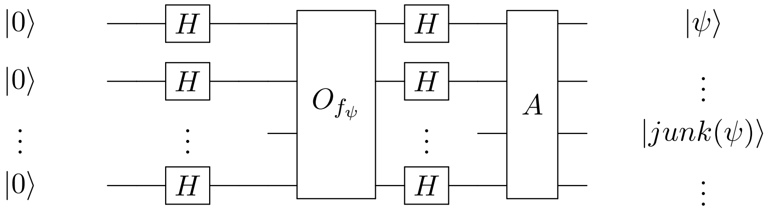
The Bernstein-Vazirani algorithm [3] allows a quantum algorithm to make *one* query to f_ψ in superposition to extract the entire string Z_ψ: first, initialize m qubits into the uniform superposition $2^{-m/2}\sum_{y\in\{0,1\}^m}|y\rangle$. This can be done by starting with the all zeroes state $|0\cdots 0\rangle$ and applying the Hadamard gate H to each qubit. Then, query the phase oracle corresponding to f_ψ. This results in the state

$$2^{-m/2}\sum_y (-1)^{f_\psi(y)}|y\rangle = 2^{-m/2}\sum_y (-1)^{\langle y, Z_\psi\rangle}|y\rangle \ .$$

Then, apply the Hadamard gate to each qubit again. This performs the Fourier transform to obtain $|Z_\psi\rangle$.

Given this, there exists a universal unitary A which takes as input $|Z_\psi\rangle \otimes |0\cdots 0\rangle$ and outputs $|\psi\rangle \otimes |junk(\psi)\rangle$ where $|junk(\psi)\rangle$ is some residual "junk" state that depends on $|\psi\rangle$. This unitary A might be extremely complicated, but the important thing is that it is a fixed unitary.

Thus, putting everything together our synthesis algorithm looks like the following:



This algorithm clearly uses exponentially many qubits, but only makes one query to an oracle.

References

[1] Scott Aaronson (2016). *Lecture notes for the 28th McGill Invitational Workshop on Computational Complexity.* arxiv preprint arXiv:1607.05256.

[2] Sandy Irani, Anand Natarajan, Chinmay Nirkhe, Sujit Rao, Henry Yuen (2021). *Quantum search-to-decision reductions and the state synthesis problem.* arxiv preprint arxiv:2111.02999.

[3] Ethan Bernstein and Umesh Vazirani (1997) *Quantum Complexity Theory.* SIAM Journal on Computing, Vol. 26, No. 5: 1411-1473.

[4] Gregory Rosenthal (2021) *Query and Depth Upper Bounds for Quantum Unitaries via Grover Search.* arxiv preprint arXiv:2111.07992.