Frontiers of Quantum Complexity and Cryptography Spring 2022

Lecture 7 - Quantum Programs and Hamiltonian Simulation

Lecturer: Henry Yuen *Scribes: Alfred Davidson*

1 Overview

In the last lecture we discussed state and unitary synthesis. While we saw a specific algorithm for state synthesis, the unitary synthesis problem is still open.

In this lecture we discuss how the unitary synthesis problem can be efficiently solved for a special class of unitaries, by encoding the unitary into a *quantum program state.*

2 Quantum Program

Consider the following alternative formulation of the unitary synthesis problem, where instead of encoding a unitary U into a boolean function f_U that the synthesis algorithm gets to query, we encode the unitary U into a quantum state $|\psi_U\rangle$ called a *program state.*

More precisely, the goal is to come up with a quantum algorithm A that takes as input an n-qubit quantum state $|\theta\rangle$ and a quantum program state $|\psi_U\rangle$ that depends on an n-qubit unitary U, and the goal is to output (an approximation of) $U|\theta\rangle$. Ideally, the quantum algorithm A should be reasonably efficient, such as run in poly(n) time.

Questions. What encoding schemes $U \mapsto |\psi_U\rangle$ are possible? Are there special classes of unitaries U for which this can be performed efficiently?

A trivial solution is the class of unitaries that have a poly(n) sized circuit description. Then $|\psi_U\rangle$ can be the classical description (bitstream) of this quantum circuit. The execution process then simply takes this description and runs the circuit on $|\theta\rangle$ to get $U|\theta\rangle$. However this is not really taking advantage of any "quantumness".

Question. Can we obtain efficient program state encodings that go beyond unitaries U that have poly(n) sized circuits?

We will see in today's lecture that it is possible. Before explaining how, we have to take a quick detour to explain *Hamiltonians.*

3 Hamiltonians and Hamiltonian Simulation

A Hamiltonian H is an observable (a Hermitian matrix). In physics, Hamiltonians are an important and useful way of describing the constraints and dynamics governing a particular system. In particular:

- A Hamiltonian specifies a numerical quantity called the *energy* for all states $|\psi\rangle$ of the system, given by the following expectation value:

  $$E\left(|\psi\rangle\right) = \langle\psi|H|\psi\rangle.$$

  This has the following interpretation. Since H is Hermitian, it can be diagonalized via the spectral theorem:

  $$H = \sum_j \underbrace{\lambda_j}_{\substack{\text{Energy corresponding}\\ \text{to eigenstate}}} \overbrace{|a_j\rangle}^{\substack{\text{Eigenstates of}\\ \text{the Hamiltonian}}} \langle a_j| \ .$$

  The vectors $|a_j\rangle$ are states of the system that have a definite energy λ_j. For a general state $|\psi\rangle$, the energy $\langle\psi|H|\psi\rangle = \sum_j \lambda_j |\langle\psi \mid a_j\rangle|^2$ can be interpreted as an *average* energy value that's weighted by the (squared) overlap of $|\psi\rangle$ with each of the eigenstates of H.

- A Hamiltonian also specifies how the state of the system (in isolation) evolves over time. The Schrödinger equation relates the unitary evolution of a system to its Hamiltonian. If we let $|\psi(t)\rangle$ denote the state of the system at time t, then its evolution is governed by the following differential equation:

  $$i\frac{d|\psi\left(t\right)\rangle}{dt} = H|\psi\left(t\right)\rangle \ .$$

  In other words, the instantaneous direction that the state $|\psi(t)\rangle$ moves in at time t is a function of the current state and the Hamiltonian H.

  This differential equation has the solution $|\psi\left(t\right)\rangle = e^{-iHt}|\psi\left(0\right)\rangle$, where $|\psi(0)\rangle$ is the state of the system at time $t = 0$.

Matrix exponentiation. Given a Hermitian matrix $H = \sum\limits_j \lambda_j |a_j\rangle\langle a_j|$ (expressed using spectral decomposition) and a function $f : \mathbb{C} \to \mathbb{C}$,

$$f\left(H\right) = \sum_j f\left(\lambda_j\right) |a_j\rangle\langle a_j|$$

Therefore, $e^{-iHt} = \sum\limits_j e^{-i\lambda_j t}|a_j\rangle\langle a_j|$. Note that while H is Hermitian, e^{-iHt} is unitary.

4 The LMR algorithm

With the notion of Hamiltonians under our belt, we now turn to a *Hamiltonian simulation* algorithm due to Lloyd, Mohseni, and Rebentrost:

Theorem 1 (S. Lloyd, M. Mohseni, P. Rebentrost [1])**.** *There exists a polynomial time quantum algorithm, that given k-copies of an n-qubit program state density matrix ρ, an n-qubit input state $|\theta\rangle$ and a parameter $t \in \mathbb{R}$, outputs a density matrix σ such that $D\left(\sigma, e^{-i\rho t}|\theta\rangle\langle\theta|e^{i\rho t}\right) \leq O\left(t^2/k\right)$.*

Here, $D(\cdot,\cdot)$ denotes the trace distance between two density matrices. The density matrix ρ is a program state that is interpreted as a Hamiltonian, and the output of the Lloyd-Mohseni-Rebentrost (abbreviated as the *LMR*) algorithm is the time-evolution of the input state $|\theta\rangle$ with respect to the Hamiltonian ρ for time t.

The reason this is called a Hamiltonian simulation algorithm is because it resembles a very important class of quantum algorithms that take in a *classical* description of a Hamiltonian H, a time t, an input state $|\theta\rangle$, and output (an approximation of) $e^{-iHt}|\theta\rangle$. This is one of the most important applications of quantum computing that we know of. The difference between this and the LMR algorithm is that the Hamiltonian is provided in *quantum form.*

The LMR algorithm. The algorithm works as follows. It takes as input $|\theta\rangle$, k copies of a density matrix ρ, and a time parameter t. Let δ be an accuracy parameter (which is either fixed or provided as input to the algorithm).

1. Set $\Delta = \delta/t$.
2. Load the input state $|\theta\rangle$ into register A.
3. For $i = 1, \ldots, O\left(t^2/\delta\right)$:
   (a) Load fresh copy of ρ into register B.
   (b) Perform the following unitary operation (that has an efficient implementation) on register A and B:

$$\underbrace{V}_{\substack{\text{`Partial Swap'} \\ \text{Operator}}} = \cos(\Delta)\,\mathbb{I} - i\sin(\Delta)\,\underbrace{U_{\text{Swap}}}_{\substack{\text{Swaps the states} \\ \text{in register } A \text{ and } B}}$$

At the end of this algorithm , the state in register A is the output of this algorithm.

A well-written analysis of this algorithm (as well as an optimality result) can be found in [2].

5 Unitary Synthesis using the LMR algorithm

We now discuss using the LMR algorithm to solve (a special case of) the Unitary Synthesis Problem. First, we explore the tantalizing possibility that we have obtained a complete solution for it.

Let U be the unitary that we wish to synthesize. The LMR algorithm computes the action of the unitary $e^{-i\rho t}$. Therefore, we need $U = e^{-i\rho t}$. By spectral decomposition,

$$U = \sum_j e^{-i\theta_j} |a_j\rangle\langle a_j| \quad 0 \le \theta_j < 2\pi$$

$$\to e^{-i\rho t} = \sum_j e^{-i\theta_j} |a_j\rangle\langle a_j|$$

$$\to \rho t = \sum_j \theta_j |a_j\rangle\langle a_j| \to \rho = \frac{1}{t}\sum_j \theta_j |a_j\rangle\langle a_j|$$

Since ρ is a density matrix, $\mathrm{Tr}\left(\rho\right) = 1 \rightarrow \frac{1}{t}\sum_j \theta_j = 1$

$$\rightarrow t = \sum_j \theta_j$$

In general, the parameter t would be an exponentially large number as the summation runs over 2^n values. In order for LMR to work correctly, we need $k \geq t^2$ copies of the program state, which is exponentially large in general.

LMR gives an efficient solution to the unitary synthesis problem for unitaries for which $\sum\limits_j \theta_j$ is poly(n) in magnitude.

Question: Are there an interesting class of unitaries for which $\sum\limits_j \theta_j$ is poly(n) in magnitude?

Yes: an example would be unitaries with *polynomial action.*

5.1 Unitaries with "polynomial action"

A unitary U acting on n-qubits has *polynomial action* if there exists a subspace $S \subseteq \left(\mathbb{C}^2\right)^{\otimes n}$ of dimension poly(n) such that for all $|\psi\rangle$ orthogonal to S, $U|\psi\rangle = |\psi\rangle$.

A special case of polynomial action unitaries are reflections about a single state.

$$U = \mathbb{I} - 2|\psi\rangle\langle\psi|$$

For this unitary, $U|\psi\rangle = -|\psi\rangle$ and $U|\theta\rangle = |\theta\rangle$ if $\langle\psi|\theta\rangle = 0$

Such unitaries can be complex, for example if $|\psi\rangle$ is chosen from the Haar distribution, then $\mathcal{C}\left(|\psi\rangle\right) \geq \exp(n) \rightarrow \mathcal{C}\left(U\right) \geq \exp(n)$. Unitary synthesis for such unitaries is non-trivial, but the LMR algorithm provides a efficient solution. Here, the complexity of unitary synthesis is pushed into the complexity of synthesizing the program state $\rho^{\otimes k}$.

References

[1] S. Lloyd, M. Mohseni, P. Rebentrost, *Quantum principal component analysis*, Nature Phys 10, 631–633 (2014). `https://doi.org/10.1038/nphys3029`

[2] S. Kimmel, C. Y. Y. Lin, G. H. Low, M. Ozols, & T.J. Yoder, *Hamiltonian simulation with optimal sample complexity.* npj Quantum Information, 3(1), 1-7. (2017). `https://doi.org/10.1038/s41534-017-0013-7`