Introduction to Quantum Computing (IN2381) – WS021/2022 **Sheet 10**

Christian B. Mendl, Irene López Gutiérrez, Keefe Huang due: 17 January 2022, 08:00 on Moodle

Tutorial 10 (Grover as a database search algorithm[1])
Grover's algorithm is sometimes referred to as a *database search algorithm*. In this tutorial we will examine how the algorithm could in principle be used to search in an unstructured database, and discuss the feasibility of this approach.

Assume we have a database containing $N = 2^n$ items, each of length l bits: $\{d_1, d_2, \ldots, d_N\}$. We want to determine where a particular item, s, is in this database.

(a) Discuss how a classical computer (with a CPU and a memory) would approach this problem. How many queries to the memory are required on average? What is the worst-case scenario?

Now imagine we are given a "quantum processing unit" (QPU) containing four registers:

- An n qubit register for the database index.
- An l qubit register for our query $|s\rangle$.
- An l qubit register for items loaded from the database, initialized as $|0\rangle$.
- A 1 qubit register initialized as $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$.

With this QPU, we can perform the following load operation: for an index x

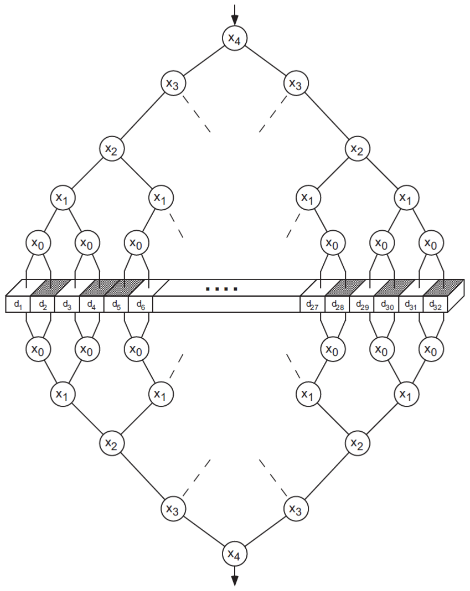
$$|x\rangle\,|s\rangle\,|t\rangle\,|-\rangle \overset{\text{LOAD}}{\mapsto} |x\rangle\,|s\rangle\,|t \oplus d_x\rangle\,|-\rangle\,.$$

In particular, for $|t\rangle = |0\rangle$ the third register will contain $|d_x\rangle$. Then, the second and the third register are compared and, if they are the same, a bit flip is applied to the forth register.

(b) What is the effect of this operation? What is its connection to Grover's algorithm?

Recall, however, that in Grover's algorithm we make use of superposition states. Thus, it may seem that in order to implement this we need a quantum memory besides our QPU. But in fact, we only need a classical memory which can be addressed by a quantum scheme.

The figure to the right shows a conceptual diagram of a 32 cell memory with a five qubit addressing scheme. The tree diagram represents the possible paths taken by an input qubit. At each node, the input qubit is sent left or right depending on the value of the qubit inside the circle. A superposition of paths is possible.



(c) What is the advantage of this scheme as opposed to a fully quantum memory?

(d) Discuss in general the practicality of this algorithm. Is the unstructured database problem common? Is the required hardware achievable?

[1] M. A. Nielsen, I. L. Chuang: *Quantum Computation and Quantum Information*. Cambridge University Press (2010), section 6.5

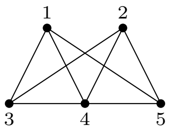
Exercise 10.1 (Hidden Linear Function problem on a specific graph)
Recall from tutorial 9 the following function q, given a square matrix A with binary entries:

$$q(x) = \sum_{i,j=1}^{n} A_{i,j} x_i x_j \bmod 4, \quad x \in \{0,1\}^n.$$

The Hidden Linear Function (HLF) problem asks to find a binary string y such that

$$q(x) = 2 \sum_{i=1}^{n} y_i x_i \bmod 4, \quad x \in \mathrm{Ker}(A).$$

A is chosen as adjacency matrix of a graph. Instead of a general square grid, here we consider the following graph as specific realization:



(a) Write down the corresponding adjacency matrix A.

(b) Compute the kernel $\mathrm{Ker}(A) \bmod 2$. (You are allowed to use a computer algebra system for this task.)

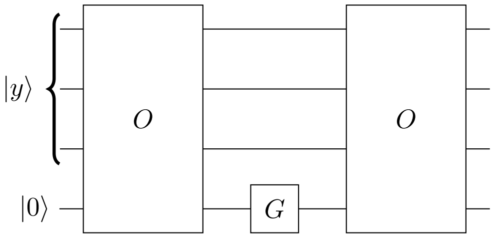
(c) Implement the quantum algorithm from part (b) and (c) of tutorial 9 using the circuit composer of IBM Q (`https://quantum-computing.ibm.com/`). Verify that one of the computational basis states appearing in the output with non-zero probability is indeed a solution to the HLF problem. Submit a screenshot showing the circuit as well as the output amplitudes or measurement probabilities.
Hint: You can create a controlled-Z gate by adding a control modifier to the Z gate in the circuit composer.

Exercise 10.2 (Quantum search as quantum simulation, part 2)
Continuing from exercise 9.2, the goal here is to *simulate* the time evolution governed by the Hamiltonian $H = |x\rangle\langle x| + |\psi\rangle\langle\psi|$ on a quantum computer. For that purpose, we can decompose $H = H_1 + H_2$ with $H_1 = |x\rangle\langle x|$ and $H_2 = |\psi\rangle\langle\psi|$, and approximate its effect via the Trotter formula, based on the identity:

$$\lim_{n\to\infty} \left(e^{-iH_1 t/n} e^{-iH_2 t/n}\right)^n = e^{-i(H_1+H_2)t}.$$

In our case, we can apply H_1 and H_2 in an alternating fashion using a small time step $\Delta t = t/n$ for some large n.

(a) Show that the following circuit implements $e^{-iH_1\Delta t}$, where $G = \left(\begin{smallmatrix} 1 & 0 \\ 0 & e^{-i\Delta t}\end{smallmatrix}\right)$ and the oracle O is defined as in exercise 9.1, i.e., O maps $|y\rangle|0\rangle \mapsto |y\rangle|1\rangle$ precisely if $y = x$, and leaves $|y\rangle|0\rangle$ invariant otherwise.



Hint: Represent the input as

$$|y\rangle \otimes |0\rangle = (I - |x\rangle\langle x|)|y\rangle \otimes |0\rangle + |x\rangle\langle x \,|\, y\rangle \otimes |0\rangle,$$

and use the series expansion of the exponential to derive that $e^{-i|x\rangle\langle x|\Delta t} = I - |x\rangle\langle x| + e^{-i\Delta t}|x\rangle\langle x|$.

(b) Modify the oracle to design a circuit analogous to part (a) that implements the time evolution with respect to $H_2 = |\psi\rangle\langle\psi|$ for the cases

(i) $|\psi\rangle = |+\rangle^{\otimes 3}$, i.e., $|\psi\rangle$ the equal superposition state

(ii) $|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)|1\rangle$

(c) Identify the circuits from (a) and (b) for a time step $\Delta t = \pi$ with the building blocks of the circuit diagram of Grover's algorithm.