Mini-project
Quantum Computation

Grover's algorithm applied to a satisfiability problem

The 3-SAT problem is a well known NP-complete problem. It consists in looking for vectors $(x_1, \ldots, x_n) \in \{0,1\}^n$ satisfying a given predicate f in the form of a conjunction of clauses which are all 3-disjunctions. For instance, here is a 3-SAT predicate:

$$f(x_0, x_1, x_2) = (x_0 \vee x_1 \vee \neg x_2) \wedge (\neg x_0 \vee \neg x_1 \vee x_2) \wedge (x_0 \vee \neg x_1 \vee x_2)$$

One looks for solutions of $f(x_0, x_1, x_2) = 1$. One can check by exhaustive search for this example that the solutions are 000, 100, 011, 101, 111. Using Grover's algorithm can help finding solutions of a predicate with a quadratic speed-up over a naive exhaustive search.

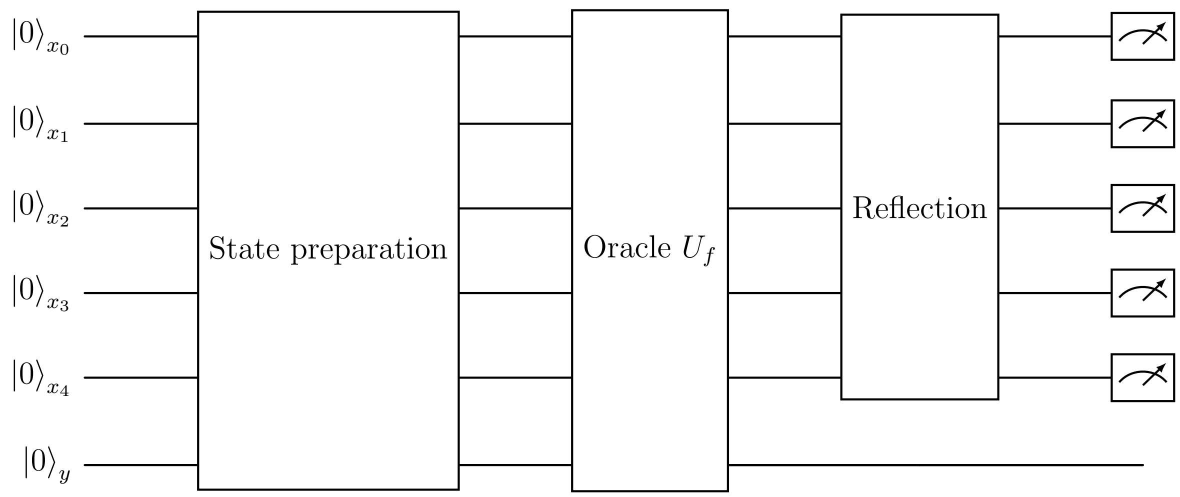
For the purpose of the present mini-project (and in order to avoid circuits with too many gates), we consider a slightly easier problem with some clauses involving only 2 variables.

Question 0. *We consider the predicate f defined as:*

$$\begin{aligned} f(x_0, x_1, x_2, x_3, x_4) =& (\neg x_0 \vee x_1) \wedge (x_0 \vee x_1 \vee \neg x_2) \wedge (x_2 \vee x_3) \\ & \wedge (\neg x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_3 \vee \neg x_4) \wedge (\neg x_2 \vee \neg x_3 \vee x_4) \end{aligned}$$

Find all possible solutions by exhaustive (or slightly refined ;-) search. How many are there?

You will now design Grover's circuit to solve the above problem:



Important remark: In the following, the aim is to work on the above problem using Grover's algorithm and *without* using any classical insight (number of solutions or formula simplification).

Question 1. *Design the state preparation circuit.*

Question 2. *Design the oracle gate U_f such that*

$$U_f |x_0, x_1, x_2, x_3, x_4, y\rangle = |x_0, x_1, x_2, x_3, x_4\rangle \otimes |y \oplus f(x_0, x_1, x_2, x_3, x_4)\rangle$$

Hints: - The basic principle here is to use one ancilla bit for each clause, using each time De Morgan's law to transform OR's into AND's, which can be easily simulated with CNOT gates; and then one extra ancilla bit to join together all clauses (so in total, there will be more than a single ancilla bit, contrary to what the circuit drawn on the former page suggests).
- Do not forget also that you have also to "invert" the circuit so that both the information bits and the intermediary ancilla bits outputs are equal to their inputs.
- Finally, please note that you are allowed to use here a generalization of the Toffoli gate with more than 2 control bits (although in practice, building such a gate might require lots of elementary gates).

Question 3. *Design the reflection operator R (not forgetting the pre and post Hadamard gates!)*

Hint: See Hw. 10, Ex. 2 for the case $n = 2$; the general case is relatively similar.

Question 4. *Run your circuit in the IBM Qiskit simulator (without noise and with noise), using k times Grover's operator $G = R \cdot U_f$ in order to retrieve the solutions. Try different values of $k \in \{1, 2, 3, 4, \ldots\}$ and see what value(s) lead(s) to the best results.*

Please comment on the solutions you find and the presence of noise, drawing the appropriate corresponding graphs. What part of the noise is due to the algorithm itself and what part is due to the noise in the machine? Try both with the simulator and one of the real machines (your choice) and describe what you see.

Hint: Watch out that the order of the output bits is the opposite of the one given above.

Question 5. *Now, using the answer to question 0: at what initial angle θ_0 does the algorithm start, and how many times should Grover's operator be applied, ideally?*

What we expect from you: Obviously, answers to the above questions! The best, by far, would be in the form of a Jupyter notebook, with all circuits and resulting histograms.

Deadline: Wednesday, May 29, 2024, 11:59 PM (on Moodle).