Lecture 13

Grover's algorithm kSAT instance with N variables and M clauses.

Example with $k=3$, $N=5$, and $M=4$

$$F(u_1,\dots,u_5) \coloneqq (u_1 \vee \neg u_2 \vee u_3) \wedge (\neg u_1 \vee u_2 \vee \neg u_3) \wedge (u_1 \vee \neg u_3 \vee \neg u_2) \wedge (u_4 \vee \neg u_5). \tag{80}$$

Computing $\mathrm{OR}_{2^N}(F(0^N),\dots,F(1^N))$ is the same as determining if there is a satisfying assignment.

The function F can be viewed as an element $F \in \{0,1\}^{2^N} = \{0,1\}^n$, where

$$n = 2^N. \tag{81}$$

Then $\mathrm{OR}_{2^N}(F(0^N),\dots,F(1^N)) = \mathrm{OR}_n(F)$.

Recall the query algorithm for computing $\mathrm{OR}_n(F)$ involves a procedure that produces 0 with the following probability,

$$p_F \coloneqq \| (|\psi\rangle\langle\psi| \otimes \mathbb{1}_2)((G \otimes \mathbb{1}_2)U_F)^l \, |\psi\rangle \otimes |1\rangle \, \|^2, \tag{82}$$

where

$$l = O(\sqrt{n}) = O(\sqrt{2^N}), \tag{83}$$

$G \coloneqq \mathbb{1}_n - 2|\psi\rangle\langle\psi|$, U_F is the quantum phase oracle of F, and

$$|\psi\rangle \coloneqq \frac{1}{\sqrt{n}} \sum_{i=1}^{n} |i\rangle = \frac{1}{\sqrt{2^N}} \sum_{x \in \{0,1\}^N} |x\rangle = H^{\otimes N} \left|0^N\right\rangle. \tag{84}$$

We can re-express

$$\begin{aligned} p_F =& \| (H^{\otimes N} \otimes \mathbb{1}_2)(|0^N\rangle\langle 0^N| \otimes \mathbb{1}_2)(H^{\otimes N} \otimes \mathbb{1}_2)((G \otimes \mathbb{1}_2)U_F)^k \, |\psi\rangle \otimes |1\rangle \, \|^2 \\ =& \| (|0^N\rangle\langle 0^N| \otimes \mathbb{1}_2)(H^{\otimes N} \otimes \mathbb{1}_2)((G \otimes \mathbb{1}_2)U_F)^k \, |\psi\rangle \otimes |1\rangle \, \|^2. \end{aligned} \tag{85}$$

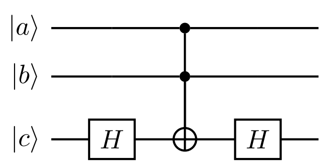
This is the probability of a circuit we drew in class outputting 0^N when measured in the computational basis at the end[12]

There are two types of unitaries in that circuit to account for in terms of elementary quantum gates.

1. O_F costs $O(kM)$ gates to implement: a circuit for F can be constructed using $O(kM)$ classical gates using the formula for F (cf. Eq. (80)), then apply the result of the last lecture. Therefore U_F also costs $O(kM)$ gates to implement (cf. the phase kickback trick).

2. $$G = \mathbb{1}_n - 2|\psi\rangle\langle\psi| = H^{\otimes N} X^{\otimes N} (\mathbb{1}_n - 2|1^N\rangle\langle 1^N|) X^{\otimes N} H^{\otimes N} \tag{86}$$

   Implementing the middle operator $(\mathbb{1}_n - 2|1^N\rangle\langle 1^N|)$ costs $O(N)$ Toffoli gates 2 H gates and $O(N)$ ancilla qubits. Example when $N=3$:

   Then did example when $N=5$.

Overall quantum time complexity of solving $kSAT$: $O(\sqrt{2^N}(kM+N))$.

Remark 7.

1. For k moderately large (say 100), the best-known classical randomized algorithm for solving $kSAT$ runs in time $\Omega(2^N)$. It is generally believed that it's impossible to do better, see Beame notes for more!

2. Randomized query lower bound of $\Omega(n) = \Omega(2^N)$ applies if we consider the *subclass* of randomized algorithm that tries to solve kSAT by only evaluating $F(u_1,u_2,\dots,u_N)$ for different settings of the u_is *without* looking into the structure of F — this is also known as "querying F". Querying F *can* be suboptimal (consider Easy3SAT with all ORs swapped with ANDs, or 2SAT). But for large k, querying F is essentially the best-known method.

3. Search-to-decision reduction. Try $u_1 = 0$ or 1, if setting $u_1 = b$ makes the formula satisfiable, then fix $u_1 = b$ and try $u_2 = 0$ or 1, etc. costs $O(\sum_{i=0}^{N} \sqrt{2^{N-i}}(kM+N-i)) = O(\sqrt{2^N}(kM+N))$.

[12] But the output should be 1 bit? Can consider classial postprocessing that outputs 0 if 0^N obtained, else output 1. This classical postprocessing can be simulated quantumly by the result from last lecture.