CSCI-B490: Quantum Programming
Homework 9
Due: Tues, Apr 21

Exercise. (Submit this homework as a Jupyter notebook.) Implement a quantum algorithm that solves Simon's problem for $n = 3$. (That, is the function implemented by the oracle takes a 3-bit input and produces a 3-bit output.) You need not implement every possible oracle: implement two for two different non-zero secret strings of your choice and one for $s = \mathtt{000}$.

Your program should include a (classical) wrapper for your implementation of the quantum algorithm which returns the actual secret string.

Hints.

Designing U_f for $s = \mathtt{000}$. Examples of injective functions f from $\{\mathtt{0},\mathtt{1}\}^3$ to $\{\mathtt{0},\mathtt{1}\}^3$ which can easily be implemented as 6-qubit circuits are permutations of the bits of the input string. Pick any permutation $p : \{x_2, x_1, x_0\} \to \{x_2, x_1, x_0\}$. Now define f by

$$x_2x_1x_0 \mapsto p(x_2)p(x_1)p(x_0)$$

The behavior of the circuit U_f is hence describable as

$$|y_2, y_1, y_0, x_2, x_1, x_0\rangle \mapsto |p(x_2) \oplus y_2, p(x_1) \oplus y_1, p(x_0) \oplus y_0, x_2, x_1, x_0\rangle$$

and can be realized by 3 gates.

Designing U_f for $s \neq \mathtt{000}$. Start with s with Hamming weight 1 (i.e., with exactly one 1-bit). Note that the relation $\sim$ defined by

$$u \sim v \iff u \oplus v \in \{\mathtt{000}, s\} \quad \text{for all } u, v \in \{\mathtt{0},\mathtt{1}\}^3$$

is an equivalence relation. So it partitions $\{\mathtt{0},\mathtt{1}\}^3$ into four subsets, which we'll call *cosets*. Choose an element from each coset. We'll call each choice a *representative* of the coset from which it was drawn. Finally define f to be the function which maps each element to the representative of its coset.

The implementation of U_f is as short as this hint is long: it can be done with two gates.

Extra credit.

Designing U_f for $s = \texttt{000}$. Explain how you might design U_f for any suitable f. (Think back to Homework 2 and to Midterm Question 1.)

Designing U_f for $s \neq \texttt{000}$. Explain how you might design U_f for any suitable f. (Good luck.)