15-459: Undergraduate Quantum Computation	September 15, 2022

Lecture 6 : Enter minus signs

Lecturer: Ryan O'Donnell

Scribe: Rajeev Godse

1 Warm-up

Notation: where the amplitude of a state s in a quantum state is Ampl(s), we can write the state as $\sum_s \text{Ampl}|s\rangle$. We can also omit states with 0 amplitudes for this sum.

So the quantum state that is 00000 with amplitude 0.36, 00100 with amplitude 0.48, and 10000 with amplitude -0.8, we can write $0.36|00000\rangle + 0.48|00100\rangle - 0.8|10000\rangle$.

For practice, we can consider what happens if we apply H to the third qubit, which we call C. We can break into cases to reason about this lucidly.

With amplitude 0.36, the state is 00000. From here, we go to 00000 with amplitude $\sqrt{\frac{1}{2}}$ and 00100 with amplitude $\sqrt{\frac{1}{2}}$.

With amplitude 0.48, the state is 00100. From here, we go to 00000 with amplitude $\sqrt{\frac{1}{2}}$ and 00100 with amplitude $-\sqrt{\frac{1}{2}}$.

With amplitude -0.8, the state is 10000. From here, we go to 10000 with amplitude $\sqrt{\frac{1}{2}}$ and 10100 with amplitude $\sqrt{\frac{1}{2}}$.

The final state is $0.84\sqrt{\frac{1}{2}}|00000\rangle - 0.12\sqrt{\frac{1}{2}}|00100\rangle - 0.8\sqrt{\frac{1}{2}}|10000\rangle - 0.8\sqrt{\frac{1}{2}}|10100\rangle$.

See that between the first two initial states there was **interference**: the final states overlapped, so the amplitudes combined.

On the other hand, the third state was independent, its result states did not overlap with any other non-zero amplitude state.

Summary: If you're doing H on C, you can pair up all the 5-bit strings based on the remaining bits and apply the usual H on C given the initial amplitudes on C in the pair. The results of each pair are independent from others. This generalizes to arbitrary instructions: we can group together the inputs with shared unused state and apply the operations on the groups in the usual way on the relevant bits.

2 Minus signs

To get *some* minus signs, we can apply H to 1. In particular, consider the program Make-M (assuming A starts at 0):

1. Add 1 to A.
2. H on A.

The resulting quantum state we will call $\mathbf{m} = \left(+\sqrt{\frac{1}{2}}\right) - \sqrt{\frac{1}{2}}$. The real name is $|-\rangle$.

Fun fact: If we do "logical negation" (aka "Add 1 to A") on $\mathbf{m}$, the resulting quantum state is $\left(-\sqrt{\frac{1}{2}}\right) + \sqrt{\frac{1}{2}} = -\mathbf{m}$. So logical negation also gives us "amplitude negation.

Question: What if after doing logical negation, we now did $(\text{Make-m})^{-1}$?

$(\text{Make-m})^{-1}(-\mathbf{m}) = -\ (\text{Make-m})^{-1}(\mathbf{m}) = -\begin{pmatrix}1\\0\end{pmatrix}$, i.e. the negation of the amplitudes of our input prior to the program (we started with $A = 0$ with amplitude 1, the final state is $A = 0$ with amplitude -1).

Question: What if we did the same, but there exist variables X, Y, Z with values $1, 1, 0$ (with amplitudes 1).

Coming in, we had $XYZ, A = 110, 0$.

Make-m on A: $\text{Ampl}[1100] = \sqrt{\frac{1}{2}}$, $\text{Ampl}[1101] = -\sqrt{\frac{1}{2}}$.

Add to A: $\text{Ampl}[1101] = \sqrt{\frac{1}{2}}$, $\text{Ampl}[1100] = -\sqrt{\frac{1}{2}}$.

$(\text{Make-m})^{-1}$ on A: $\text{Ampl}[1100] = -1$.

Idea: X, Y, Z have certain 0/1 values. $A = 0$, temp1, temp2, ... $= 0$. ["ancillas"]. Then we apply the operations:

1. Make-m on A
2. Add $F(X, Y, Z)$ to A
3. $(\text{Make-m})^{-1}$ on A

If $F(x, y, z) = 0$, $\text{Ampl}[X, Y, Z, A = 0] = 1$.

If $F(x, y, z) = 1$, $\text{Ampl}[X, Y, Z, A = 0] = -1$.

Notice that no bits changed values, but the amplitude sign changed if $F(X, Y, Z) = 1$.

Let's call this operation "If F(X, Y, Z) then Minus". We again assume some number of ancillas (including A) initialized to 0.

This gives an important quantum operation: "Sign-computing F" or $F^{\pm}$ (only appropriate for $F : \{0, 1\}^n \to \{0, 1\}$).

Sign-computation is nice for a couple of reasons. We save a bit of storage by using the amplitudes, but more importantly, we have a natural way to introduce minuses and get some nice cancellation (destructive interference) going.

3 HAD

Say you have two numbers like 8 and 6.

Average & Displacement(8, 6) = (7, 1) because $7 \pm 1 = 8, 6$

Add & Difference(7, 1) = (8, 6) because $7 \pm 1 = 8, 6$

See that Add&Diff $\circ$ Avg&Disp = Avg&Disp $\circ$ Add&Diff = 1.

Add&Diff is a linear transformation, the corresponding matrix is $\begin{pmatrix}1 & 1\\1 & -1\end{pmatrix}$. The same is true for Avg&Disp, with corresponding matrix $\begin{pmatrix}+\frac{1}{2} & +\frac{1}{2}\\+\frac{1}{2} & -\frac{1}{2}\end{pmatrix}$.

Notice that the matrix corresponding to Hadamard, we get $\begin{pmatrix}+\sqrt{\frac{1}{2}} & +\sqrt{\frac{1}{2}}\\+\sqrt{\frac{1}{2}} & -\sqrt{\frac{1}{2}}\end{pmatrix}$. Hadamard scales both matrices, in fact using reciprocal scaling factors, which we can think of as splitting the difference between the two matrices. This yields an intuitive explanation for why Hadamard is an involution from the above numerical examples.

In summary, Avg&Disp is “basically” H except for an extra scalar factor of $\sqrt{\frac{1}{2}}$, and Add&Diff is “basically” H except for an extra scalar factor of $\sqrt{2}$.

This yields a nice **trick**: If you do an even number of H instructions, it’s okay to pretend half are Aug&Disp and half are Add&Diff.

In fact, there is an even nicer **trick**: It is okay to change H instructions to A&D willy-nilly as long as you tell the reader you’re working with “unnormalized quantum states.” We can just scale down the sum of the squares at the end to get the true quantum state.