15-459: Undergraduate Quantum Computation September 6, 2022

Lecture 3 : Analyzing quantum code

Lecturer: Ryan O'Donnell

Scribe: Rajeev Godse

1 Probabilistic analogue

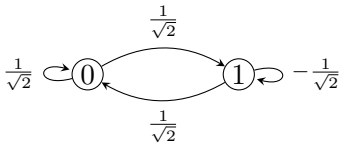
Recall our usual analysis of probabilistic algorithms, which consists of drawing a probability tree, where each edge has an associated probability, and all edges from an internal node to its children have probability summing to 1. The leaves represent the final states of the program, with probability given by the product of probabilities on the root-to-leaf path.

Recall also the associated properties:

- Forking in the probability tree only when using random bits.
- Number of leaves $= 2^{\#\text{ random bits}}$
- Leaf probabilities are all (integer) $\cdot \left(\frac{1}{2}\right)^{\#\text{ random bits}}$
- Leaf probabilities sum to 1.
- Final outcome probabilities all have positive values.
- Final outcome probabilities all sum up to 1.

2 Hat

Define the operation Hat on A given by advancing one step in the below "Markov chain":



The labels of the edges are now given by **amplitudes** instead of probabilities, which follow the below properties:

- Squares of amplitudes sum to 1.
- One still unstated but important property.

To get the amplitudes of a leaf in an amplitude tree, multiply along the root-to-leaf path. To get the total amplitude of an outcome, add amplitudes of all leaves with that outcome.

Then, observe:

- Forking in the amplitude tree only when using Hat.
- Number of leaves $= 2^{\#\text{ Hats}}$
- Leaf amplitudes are all $\left(\frac{1}{\sqrt{2}}\right)^{\#\text{ Hats}}$

- Squares of leaf amplitudes sum to 1.
- Final outcome amplitudes can be positive, negative, or zero.
- Squares of final outcome amplitudes all sum up to 1.

"Print all" probabilities are just amplitudes squared (if we ask the quantum program to print the state of all variables, it will print each state with probability equal to the amplitude of that state squared).