Lecture 3: Analyzing Quantum Code

[Last time we saw how classical AND/OR/NOT code became quantum with 1 instruction...]

H.A.T.! = "Hat All the Things!"

[I'll reveal the real/proper names of these instructions soon, but for now, suffice it to say H.A.T.! is just doing...]

= "Hat on Thing A" for all A.

[Today I'll explain exactly what this Hat instruction does. Particularly, for the following simple programs: (Demo them in Scratch!)]

1a, b

```
Make A
(Add 1 to A)
Hat on A
Print All
```

2a, b

```
Make A
(Add 1 to A)
Hat on A
Hat on A
Print All
```

3

```
Make A
Hat on A
Make B
Add A to B
Print All
```

[Analyzing quantum code is almost identical to...]

Analyzing Probabilistic Code

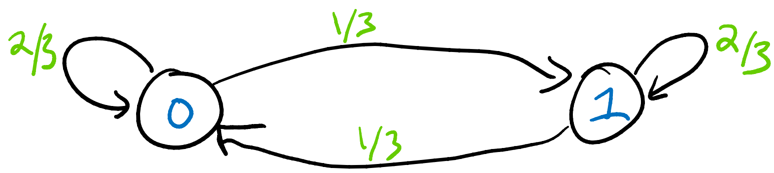
[Normally, for Prob. Computing you'd add a "Rand Bit" instruction. But to keep you on your toes, and also for comparison to Hat, I'll introduce a different probabilistic instruction:]

$\boxed{\text{Noise}}$ on Thing $\underline{A}$: does "Add 1 to A" with prob 1/3, else nothing ("Add 0 to A")

[Demo in Scratch]

$$\left\{\begin{array}{l} 0 \longmapsto \tfrac{2}{3} \text{ prob of } 0,\ \tfrac{1}{3} \text{ prob of } 1 \\ 1 \longmapsto \tfrac{1}{3} \text{ prob of } 0,\ \tfrac{2}{3} \text{ prob of } 1 \end{array}\right\}$$

[The "Markov chain"/"automaton" picture....]



[Remember from 259?]

[Remark: would get a "valid probabilistic instruction" so long as #'s out of each state are ≥ 0 & sum to 1.]

[Suppose we took all Quantum progs 1a,b, 2a,b, 3 from page 1 and changed "Hat on ___" to "Noise on ___". We'd get Probabilistic progs, and I expect you to be able to "analyze" them. Indeed

ALL OF PROB. THEORY ≡ ANALYZING PROB. CODE.

Incidentally, you can do all of Prob. Computing with just the Noise instruction, albeit that's annoying. Similarly, can do all of Q.C. with just Hat, though it's annoying. Will show you more instructions making life better, later.]

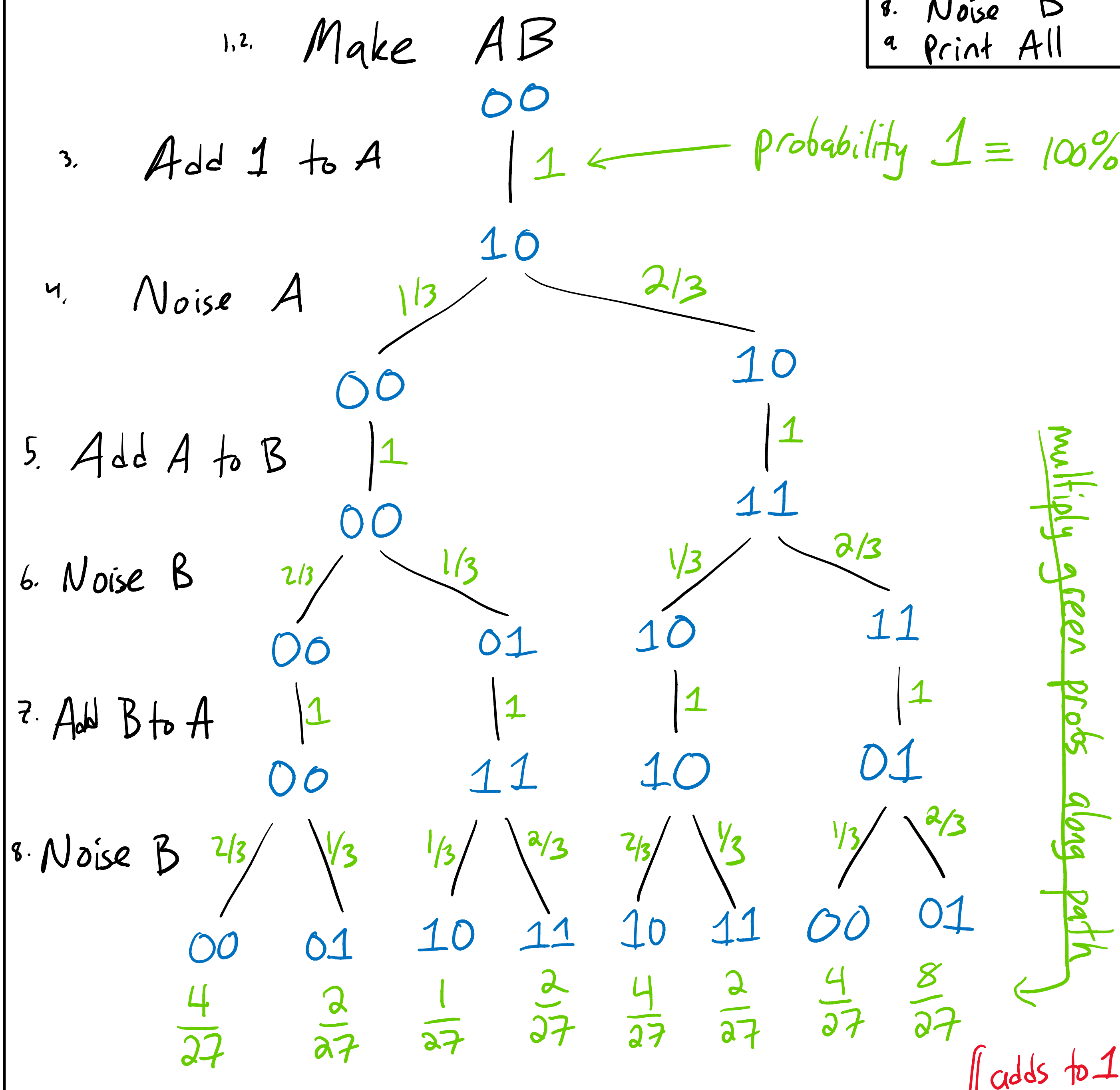
[A refresher, with some meaningless Probabilistic code...]

1. Make A
2. Make B
3. Add 1 to A
4. Noise A
5. Add A to B
6. Noise B
7. Add B to A
8. Noise B
9. Print All

[Analyzing code w/ no randomness ≡ linearly tracing thru code. With randomness, there are "forks". Need a...]

Probability Tree

1. Make A
2. Make B
3. Add 1 to A
4. Noise A
5. Add A to B
6. Noise B
7. Add B to A
8. Noise B
9. Print All



00	01	10	11	10	11	00	01
$\frac{4}{27}$	$\frac{2}{27}$	$\frac{1}{27}$	$\frac{2}{27}$	$\frac{4}{27}$	$\frac{2}{27}$	$\frac{4}{27}$	$\frac{8}{27}$

$$\text{Prob}[00] = \frac{4}{27} + \frac{4}{27} = \boxed{\frac{8}{27}}$$, and similarly

$$\text{Prob}[01] = \boxed{\frac{10}{27}}$$

$$\text{Prob}[10] = \boxed{\frac{5}{27}}$$

$$\text{Prob}[11] = \boxed{\frac{4}{27}}$$

[still adds to 1]

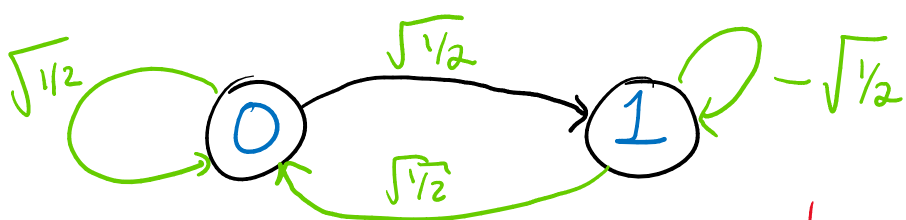
Observe:

- Forking only @ Noise instruction
- # leaves $= 2^{\#\text{Noise instrs}}$
- leaf probs are all $(\text{integer}) \cdot \left(\frac{1}{3}\right)^{\#\text{Noise instrs}}$
- leaf probs add to 1
- final outcome probs add up "constructively"
- final outcome probs also add to 1

[Deep remark: Given code, actually <u>computing</u> all outcome probs takes exponential time. Computing just Prob[00...0] is not only NP-hard, it's "harder" than that (see 15-855....). But <u>simulating</u> the final PrintAll is <u>easy</u> with a Probabilistic computer.]

[That's all of Probabilistic Computing – done! I hope you found it straightforward. On to Quantum Computing! <u>NOTE:</u> Delete the Noise instruction now; we only want Hat. It's <u>possible</u> to analyze code with both Probabilistic & Quantum instructions, but it's <u>messy</u>. We basically won't do it in this course: Noise or Hat but not both!]

The "Markov chain"/ "automaton" picture for $\boxed{\text{Hat}}$:



[!!]

[Clearly, the green labels are no longer probabilities. They can be negative!! So, change their name...]

~~probability~~

⟶ AMPLITUDE

⇐ Drake meme here

<u>Obs:</u>
- out of each state, amplitude^2 sums to 1.
- another <u>crucial</u> property holds [but more on this crucial property later....]

0	1
ampl. $+\sqrt{\frac{1}{2}}$ of 0	ampl. $+\sqrt{\frac{1}{2}}$ of 0
ampl. $+\sqrt{\frac{1}{2}}$ of 1	ampl. $-\sqrt{\frac{1}{2}}$ of 1

MEMORIZE!

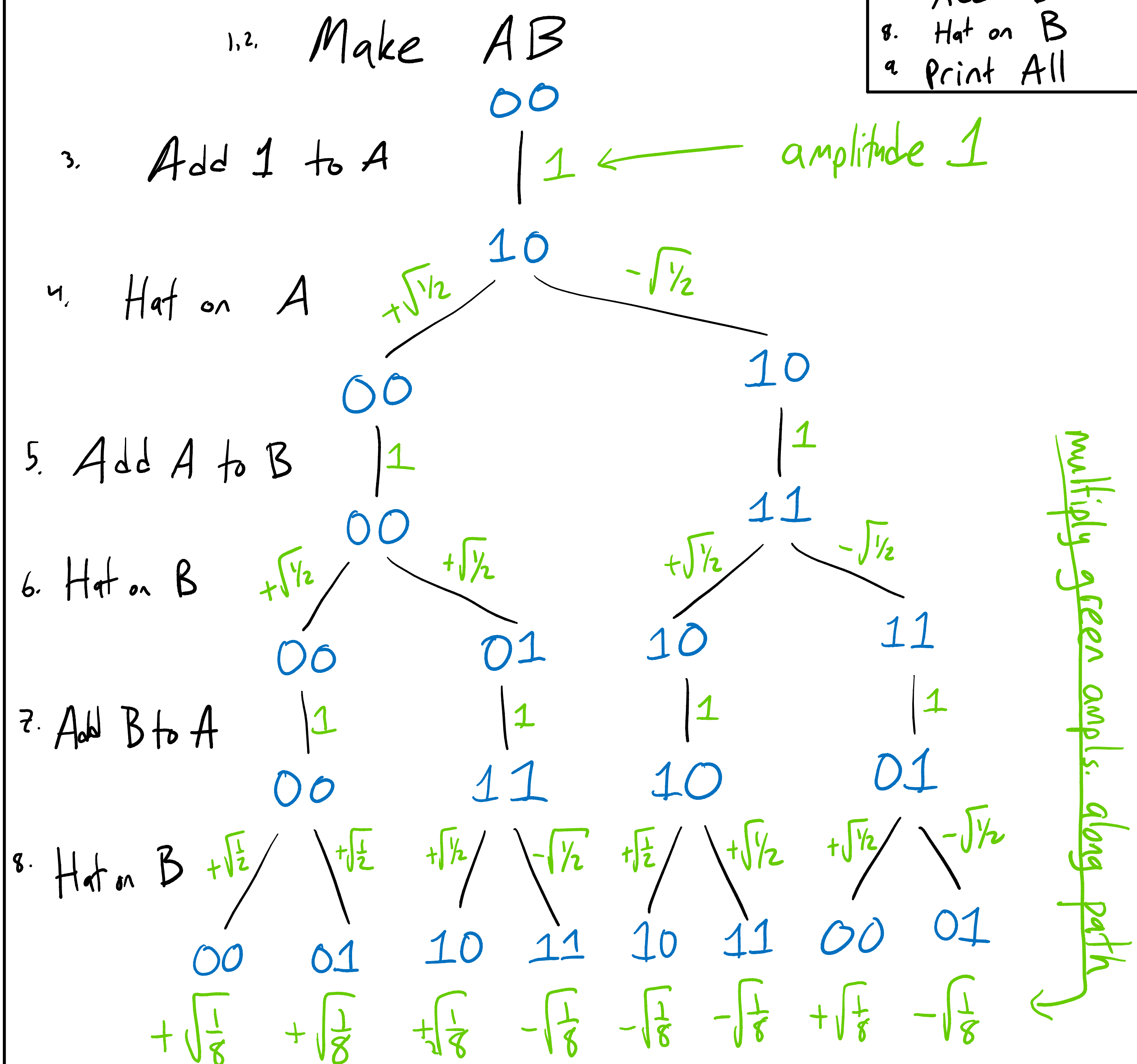
$$\text{Hat} \equiv \begin{matrix} +\sqrt{1/2} & +\sqrt{1/2} \\ +\sqrt{1/2} & -\sqrt{1/2} \end{matrix}$$

THE PROCESS OF ANALYZING QUANTUM CODE UP UNTIL THE "PRINT ALL" IS THE SAME AS WITH PROBABILISTIC CODE

except you say "amplitude" instead of "probability".

1. Make A
2. Make B
3. Add 1 to A
4. Hat on A
5. Add A to B
6. Hat on B
7. Add B to A
8. Hat on B
9. Print All

"Amplitude Tree"



00	01	10	11	10	11	00	01
$+\sqrt{\frac{1}{8}}$	$+\sqrt{\frac{1}{8}}$	$+\sqrt{\frac{1}{8}}$	$-\sqrt{\frac{1}{8}}$	$-\sqrt{\frac{1}{8}}$	$-\sqrt{\frac{1}{8}}$	$+\sqrt{\frac{1}{8}}$	$-\sqrt{\frac{1}{8}}$

$$\mathrm{Ampl}[00] = +\sqrt{\tfrac{1}{8}}+\sqrt{\tfrac{1}{8}} = +2\sqrt{\tfrac{1}{8}} = +\sqrt{\tfrac{4}{8}}$$

$$\mathrm{Ampl}[01] = +\sqrt{\tfrac{1}{8}}-\sqrt{\tfrac{1}{8}} = 0$$

[!!] → "destructive interference"

$$\mathrm{Ampl}[10] = \quad = 0$$

$$\mathrm{Ampl}[11] = \quad = -\sqrt{\tfrac{4}{8}}$$

Observe:
- Forking only @ Hat instruction
- $\#$ leaves $= 2^{\#\text{ Hat instrs}}$
- leaf amplitudes are all $\pm\sqrt{\frac{1}{2}}^{\,\#\text{ Hat instrs}}$,
- leaf amplitudes^2 add to 1
- final outcome ampls add up {constructively & destructively}
- final outcome amplitudes^2 also add up to 1!

[This last thing always happens, but it shouldn't be obvious why. It relies on "crucial" property of Hat instruction not discussed yet.]

[[Final amplitudes (before "Print All" ... →]]

$\text{Ampl}[00] = +\sqrt{\frac{4}{8}}$

$\text{Ampl}[01] = 0$

[[Their squares sum to 1.]]

$\text{Ampl}[10] = 0$

$\text{Ampl}[11] = -\sqrt{\frac{4}{8}}$

[[What do you might imagine happens when you do "Print All"?]]

"Print All" probabilities = amplitudes^2.

In example:

$\text{Prob}[\text{printing } 00] = \left(+\sqrt{\frac{4}{8}}\right)^2 = \frac{4}{8} = \frac{1}{2}$

$\text{Prob}[01] = 0^2 = 0$

$\text{Prob}[10) = 0^2 = 0$

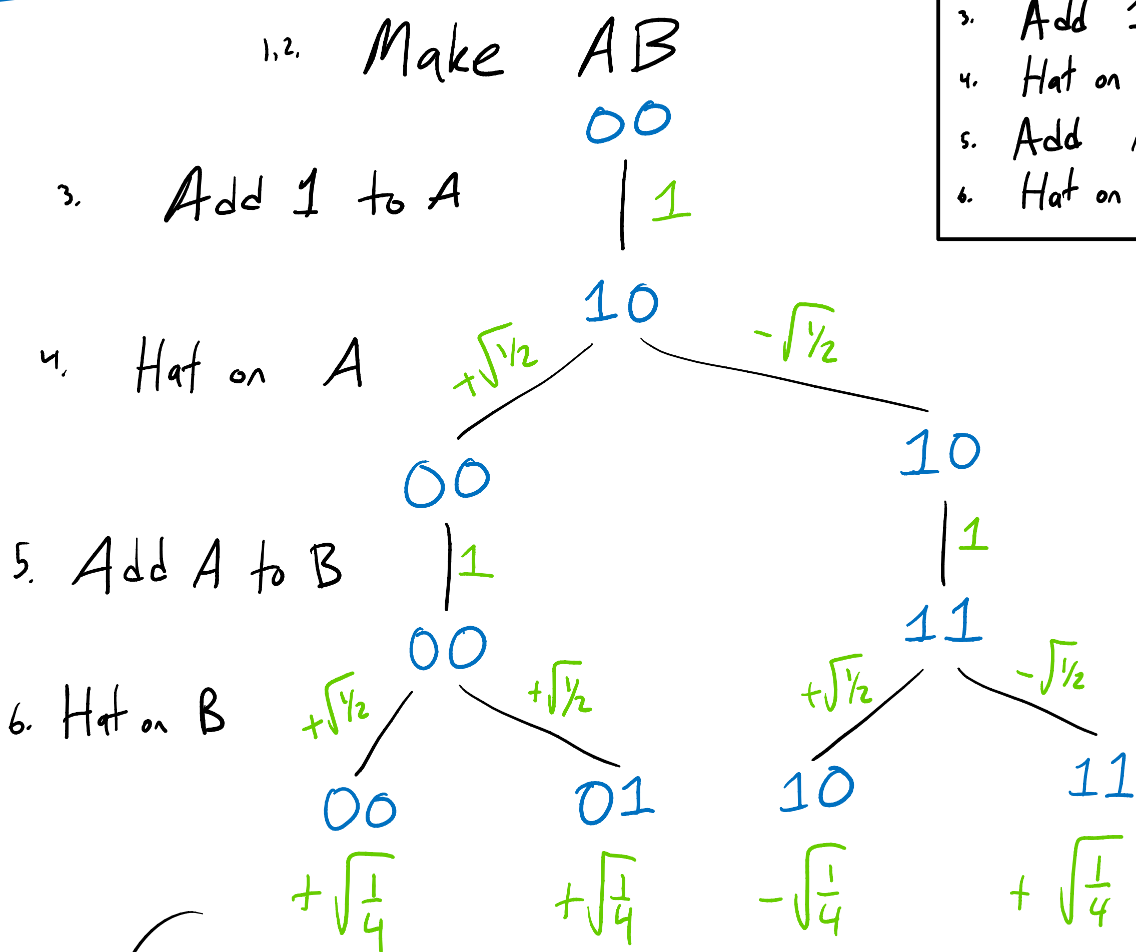
$\text{Prob}[11] = \left(-\sqrt{\frac{4}{8}}\right)^2 = \frac{4}{8} = \frac{1}{2}$

[[Demo it in Scratch!]]

[[Let's also try the same program with last two lines deleted...]]

"Amplitude Tree"

1. Make A
2. Make B
3. Add 1 to A
4. Hat on A
5. Add A to B
6. Hat on B



[no "interference" in this example]

Upon "Print All..."

$$\mathrm{Ampl}[00] = +\sqrt{\tfrac{1}{4}} \qquad \mathrm{Prob}[00] = \left(+\sqrt{\tfrac{1}{4}}\right)^2 = \tfrac{1}{4}$$

$$\mathrm{Ampl}[01] = +\sqrt{\tfrac{1}{4}} \implies \mathrm{Prob}[01] = \tfrac{1}{4}$$

$$\mathrm{Ampl}[10] = -\sqrt{\tfrac{1}{4}} \qquad \mathrm{Prob}[10] = \tfrac{1}{4}$$

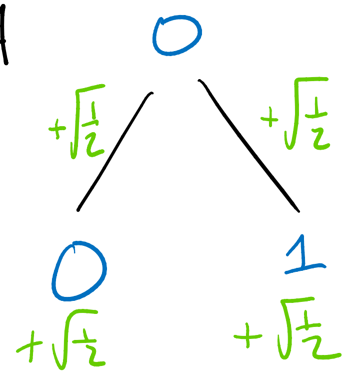
$$\mathrm{Ampl}[11] = +\sqrt{\tfrac{1}{4}} \qquad \mathrm{Prob}[11] = \tfrac{1}{4}$$

[Demo in Scratch]

[[Those examples were to practice calculation. Now let's return to the more interesting (and simpler) Quantum programs from the beginning...]]

(1a) Make A

Hat on A

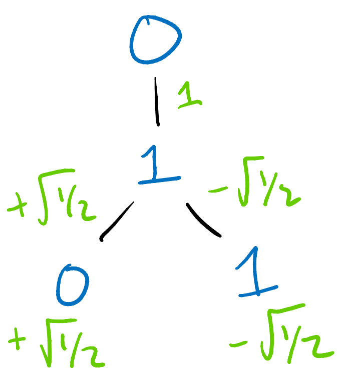


$\longrightarrow$ $\text{Ampl}[0] = +\sqrt{\frac{1}{2}}$

$\text{Ampl}[1] = +\sqrt{\frac{1}{2}}$

Print All : $\text{Prob}[0] = \frac{1}{2}$, $\text{Prob}[1] = 1/2$.

(1b) Make A, Add 1 to A, Hat on A...

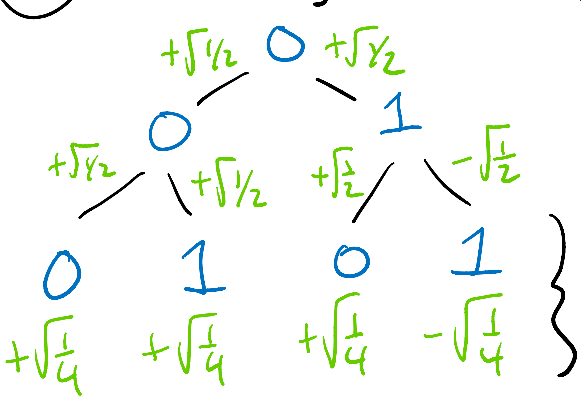


$\text{Ampl}[0] = +\sqrt{1/2}$

$\text{Ampl}[1] = -\sqrt{1/2}$

Print All: $\text{Prob}[0] = \frac{1}{2}$, $\text{Prob}[1] = \frac{1}{2}$.

(2a) Continuing from (1a), Hat A again...



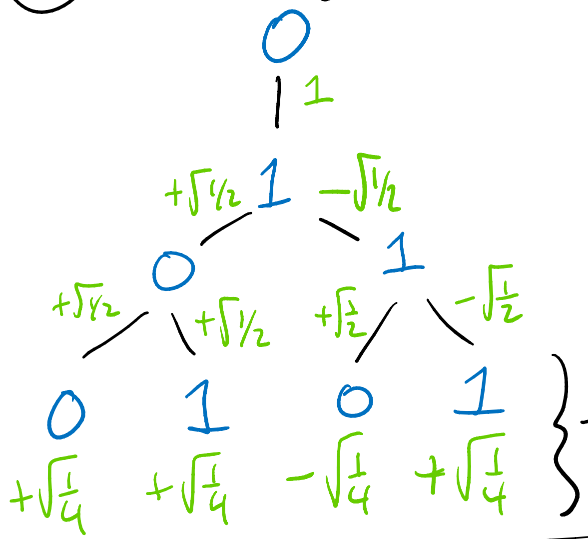
$\text{Ampl}[0] = +\sqrt{\frac{1}{4}} + \sqrt{\frac{1}{4}} = 1$

$\text{Ampl}[1] = +\sqrt{\frac{1}{4}} - \sqrt{\frac{1}{4}} = 0$

$\therefore$ Print All $\Rightarrow$

$\text{Prob}[0] = 1$, $\text{Prob}[1] = 0$.

2b Continuing from 1b, Hat A again...



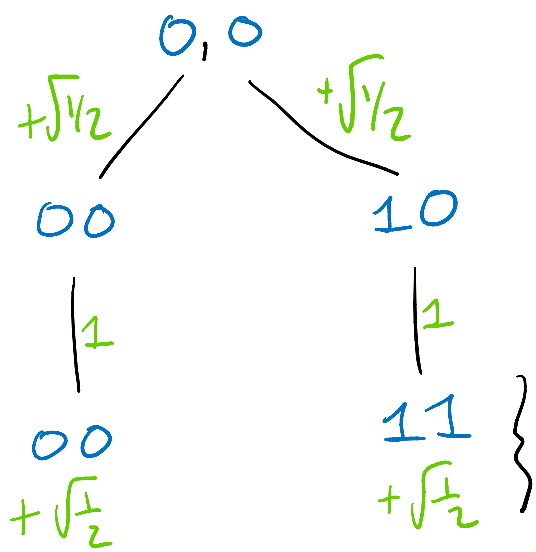
$\mathrm{Amp}([0]) = +\sqrt{\frac{1}{4}} - \sqrt{\frac{1}{4}} = 0$

$\mathrm{Amp}([1]) = +\sqrt{\frac{1}{4}} + \sqrt{\frac{1}{4}} = 1$

$\therefore$ Print All $\Rightarrow$

$\mathrm{Prob}[0] = 0$, $\mathrm{Prob}[1] = 1$.

3 Make A, B. Hat on A. Add A to B.



$\mathrm{Amp}([00]) = +\sqrt{\frac{1}{2}}$

$\mathrm{Amp}([11]) = +\sqrt{\frac{1}{2}}$

$(\mathrm{Amp}([01]), \mathrm{Amp}([10]) = 0)$

"Entangled qubits"

ooooh, spooky!

[[Sure, if you Print All now, both 0 or both 1]]

[[But this is no spookier than the Probabilistic analogue...]]

Make A, B. Noise on A. Add A to B. $\rightarrow$ $\Pr[00] = 2/3$

$\Pr[11] = 1/3$

"Correlated bits" $\rightarrow$ not spooky.

(Upcoming lectures... the real names of quantum instructs, genuinely interesting programs, & more...)