Combining Outcomes

2 bits: 00 01 10 11 — 4 outcomes

1 bit, 1 die: 01 02 ... 06
11 12 ... 16 — 12 outcomes

system with n outcomes + system with m outcomes → $\boxed{nm \text{ outcomes}}$

Two Bits

"20% chance of being in state 00"

Example Distribution: $0.3|00\rangle + 0.2|01\rangle + 0.2|10\rangle + 0.3|11\rangle$

- the 4 values the bits can have are being described by a 4D vector space
- $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ are the basis vectors
- any vectors with non-negative coefficients that sum to 1 are valid states

Basis Vectors (in general):

$$|\underset{n}{i}, \underset{m}{j}\rangle$$

comma is optional, but helps with readability

Combining States

first bit — second bit

$$\left(\tfrac{1}{3}|0\rangle + \tfrac{2}{3}|1\rangle\right) \otimes \left(\tfrac{1}{2}|0\rangle + \tfrac{1}{2}|1\rangle\right)$$

tensor product

can multiply out the values

$$\tfrac{1}{3}\cdot\tfrac{1}{2}|00\rangle + \tfrac{1}{3}\cdot\tfrac{1}{2}|01\rangle + \tfrac{2}{3}\cdot\tfrac{1}{2}|10\rangle + \tfrac{2}{3}\cdot\tfrac{1}{2}|11\rangle$$

$$= \tfrac{1}{6}|00\rangle + \tfrac{1}{6}|01\rangle + \tfrac{1}{3}|10\rangle + \tfrac{1}{3}|11\rangle$$

Tensor Product Rules

$$(a|x\rangle) \otimes |y\rangle = a(|x\rangle \otimes |y\rangle)$$
$$|x\rangle \otimes (a|y\rangle) = a(|x\rangle \otimes |y\rangle)$$

$$(|x\rangle + |y\rangle) \otimes |z\rangle = |x\rangle \otimes |z\rangle + |y\rangle \otimes |z\rangle$$
$$|z\rangle \otimes (|x\rangle + |y\rangle) = |z\rangle \otimes |x\rangle + |z\rangle \otimes |y\rangle$$

* all rules are equalities, so you can go both ways *

SHORTHAND: $|x\rangle \otimes |y\rangle \longrightarrow |x, y\rangle$

Applying Tensor Product Rules

$$\Big(\underbrace{\tfrac{1}{3}|0\rangle + \tfrac{2}{3}|1\rangle}_{x,\ y}\Big) \otimes \Big(\underbrace{\tfrac{1}{2}|0\rangle + \tfrac{1}{2}|1\rangle}_{z}\Big)$$

first bit

second bit

$$= \tfrac{1}{3}|0\rangle \otimes \left(\tfrac{1}{2}|0\rangle + \tfrac{1}{2}|1\rangle\right) + \tfrac{2}{3}|1\rangle \otimes \left(\tfrac{1}{2}|0\rangle + \tfrac{1}{2}|1\rangle\right)$$

$$= \tfrac{1}{3}|0\rangle \otimes \tfrac{1}{2}|0\rangle + \tfrac{1}{3}|0\rangle \otimes \tfrac{1}{2}|1\rangle + \tfrac{2}{3}|1\rangle \otimes \tfrac{1}{2}|0\rangle + \tfrac{2}{3}|1\rangle \otimes \tfrac{1}{2}|1\rangle$$

moving coefficients

$$= \tfrac{1}{6}|0\rangle \otimes |0\rangle + \tfrac{1}{6}|0\rangle \otimes |1\rangle + \tfrac{1}{3}|1\rangle \otimes |0\rangle + \tfrac{1}{3}|1\rangle \otimes |1\rangle$$

$$= \boxed{\tfrac{1}{6}|0,0\rangle + \tfrac{1}{6}|0,1\rangle + \tfrac{1}{3}|1,0\rangle + \tfrac{1}{3}|1,1\rangle}$$

! ↳ this form is not as clear about what is happening in individual bits !

Tensor Products of Operations

- "apply F to first bit, G to second" $\longrightarrow$ $\boxed{F \otimes G}$

$$(F|x\rangle \otimes G|y\rangle) = (F \otimes G)(|x\rangle \otimes |y\rangle)$$

Kronecker Products

$$\underset{|x\rangle}{\begin{bmatrix} a \\ b \end{bmatrix}} \otimes \underset{|y\rangle}{\begin{bmatrix} c \\ d \end{bmatrix}} = \begin{array}{c} \begin{bmatrix} ac \\ ad \\ bc \\ bd \end{bmatrix} \end{array} \begin{array}{c} 00 \\ 01 \\ 10 \\ 11 \end{array} = \begin{bmatrix} a|y\rangle \\ b|y\rangle \end{bmatrix}$$

Kronecker Products of Operations

$$F = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \qquad G = \begin{bmatrix} e & f \\ g & h \end{bmatrix}$$

$$F \otimes G = \begin{bmatrix} aG & bG \\ cG & dG \end{bmatrix} = \begin{bmatrix} ae & af & be & bf \\ ag & ah & bg & bh \\ ce & cf & de & df \\ cg & ch & dg & dh \end{bmatrix}$$

Example

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad F = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}$$

$$(X \otimes F)|1,0\rangle = X|1\rangle \otimes F|0\rangle$$

$$= |0\rangle \otimes |1\rangle = |0,1\rangle \checkmark$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \rightarrow \text{corresponds to } |0,1\rangle$$

↳ corresponds to $|1,0\rangle$

Size of Probabalistic States

- n-bit state requires 2^n numbers to describe
- n-bit operation requires $2^n \times 2^n$ numbers to write down

ex: we have 100 bits. Don't want to write a matrix that is $2^{100} \times 2^{100}$!

To operate on just one bit, we could describe an operation as:

$$\underbrace{I \otimes \ldots \otimes I}_{99 \text{ times}} \otimes F$$

Implementing Probabalistic Computing

Those numbers are super big, does this mean probabalistic computing isn't possible?

NOPE!

all possible outcomes

- we can't write down the full distribution on 1000 bits
- however, during computation, we don't need the full distribution
  - to get a random bit, we flip a fair coin and record 0 or 1
  - rather than the distribution on 2^n val s, we record one outcome

HOWEVER!

only n bits

Sampling

- we are no longer getting a distribution, but a random sample
- when we run it many times, we would find that each output arises only a fraction of the times indicated by the distribution
- for decision problems (BPP), the output is $\{0,1\}$ and we only need a 67% prob of sampling the correct answer

Meaning of Probabalistic States

- they represent our knowledge of the system, but the machine takes a definite value
  - can describe state as $[0.6, 0.4]^T \Rightarrow$ 60% of situation gives 0
  - the bit itself is 0 or 1
- the problem of exponential size during analysis is NOT true for quantum size
  - therefore quantum systems are hard to simulate
  - can't keep track of 2^n each time

Sampling Quantum States

- output of a quantum computation is also a random sample
- not being able to write down states of that computation does NOT mean another method is impossible
- for most quantum simulation problems, we expect another way is NOT possible
- in general, the burden is in proving another way is not possible