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 \( \rightarrow \) \( nm \) outcomes

Two Bits

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

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

Basis Vectors (in general):

\[ |i, j\> \]

\( n \uparrow \quad m \downarrow \)

common is optional, but helps with readability

Combining States

\[
\left( \frac{1}{3} |0\> + \frac{2}{3} |1\> \right) \otimes \left( \frac{1}{2} |0\> + \frac{1}{2} |1\> \right)
\]

can multiply out the values

\[
\frac{1}{3} \cdot \frac{1}{2} |00\> + \frac{1}{3} \cdot \frac{1}{2} |01\> + \frac{2}{3} \cdot \frac{1}{2} |10\> + \frac{2}{3} \cdot \frac{1}{2} |11\>
\]

= \( \frac{1}{6} |00\> + \frac{1}{6} |01\> + \frac{1}{3} |10\> + \frac{1}{3} |11\> \)
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:** \( |x\rangle \otimes |y\rangle \rightarrow |x,y\rangle \)

**Applying Tensor Product Rules**

\[
\left( \frac{1}{3}|0\rangle + \frac{2}{3}|1\rangle \right) \otimes \left( \frac{1}{2}|0\rangle + \frac{1}{2}|1\rangle \right)
\]

\[
= \frac{1}{3}|0\rangle \otimes \left( \frac{1}{2}|0\rangle + \frac{1}{2}|1\rangle \right) + \frac{2}{3}|1\rangle \otimes \left( \frac{1}{2}|0\rangle + \frac{1}{2}|1\rangle \right)
\]

\[
= \frac{1}{3}|0\rangle \otimes \frac{1}{2}|0\rangle + \frac{1}{3}|0\rangle \otimes \frac{1}{2}|1\rangle + \frac{2}{3}|1\rangle \otimes \frac{1}{2}|0\rangle + \frac{2}{3}|1\rangle \otimes \frac{1}{2}|1\rangle
\]

\[
= \frac{1}{6}|0,0\rangle + \frac{1}{6}|0,1\rangle + \frac{1}{6}|1,0\rangle + \frac{1}{6}|1,1\rangle
\]

\[
\text{moving coefficients}
\]

\[
\text{this form is not as clear about what is happening in individual bits}
\]

**Tensor Products of Operations**

*“apply F to first bit, G to second”* \( \rightarrow \) \( F \otimes G \)

\( (F|x\rangle \otimes G|y\rangle) = (F \otimes G)(|x\rangle \otimes |y\rangle) \)
Kronecker products

\[
\begin{bmatrix} a \\ b \end{bmatrix} \otimes \begin{bmatrix} c \\ d \end{bmatrix} = \begin{bmatrix} ac \\ ad \\ bc \\ bd \end{bmatrix} = \begin{bmatrix} a |y\rangle \\ b |y\rangle \end{bmatrix}
\]

Kronecker products of operations

\[
F = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad G = \begin{bmatrix} e & f \\ g & h \end{bmatrix}
\]

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

Example

\[
X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad F = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}
\]

\[(X \otimes F) |1,0\rangle = x |1\rangle \otimes F |0\rangle \]

\[
= |0\rangle \otimes |1\rangle = |0,1\rangle
\]

\[
\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad \text{corresponds to } |0,1\rangle
\]

\[
\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad \text{corresponds to } |1,0\rangle
\]

Size of Probabilistic 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:

\[
I \otimes \ldots \otimes I \otimes F \quad \text{99 times}
\]
Implementing Probabalistic Computing

Those numbers are super big, does this mean probabilistic 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 vals, 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