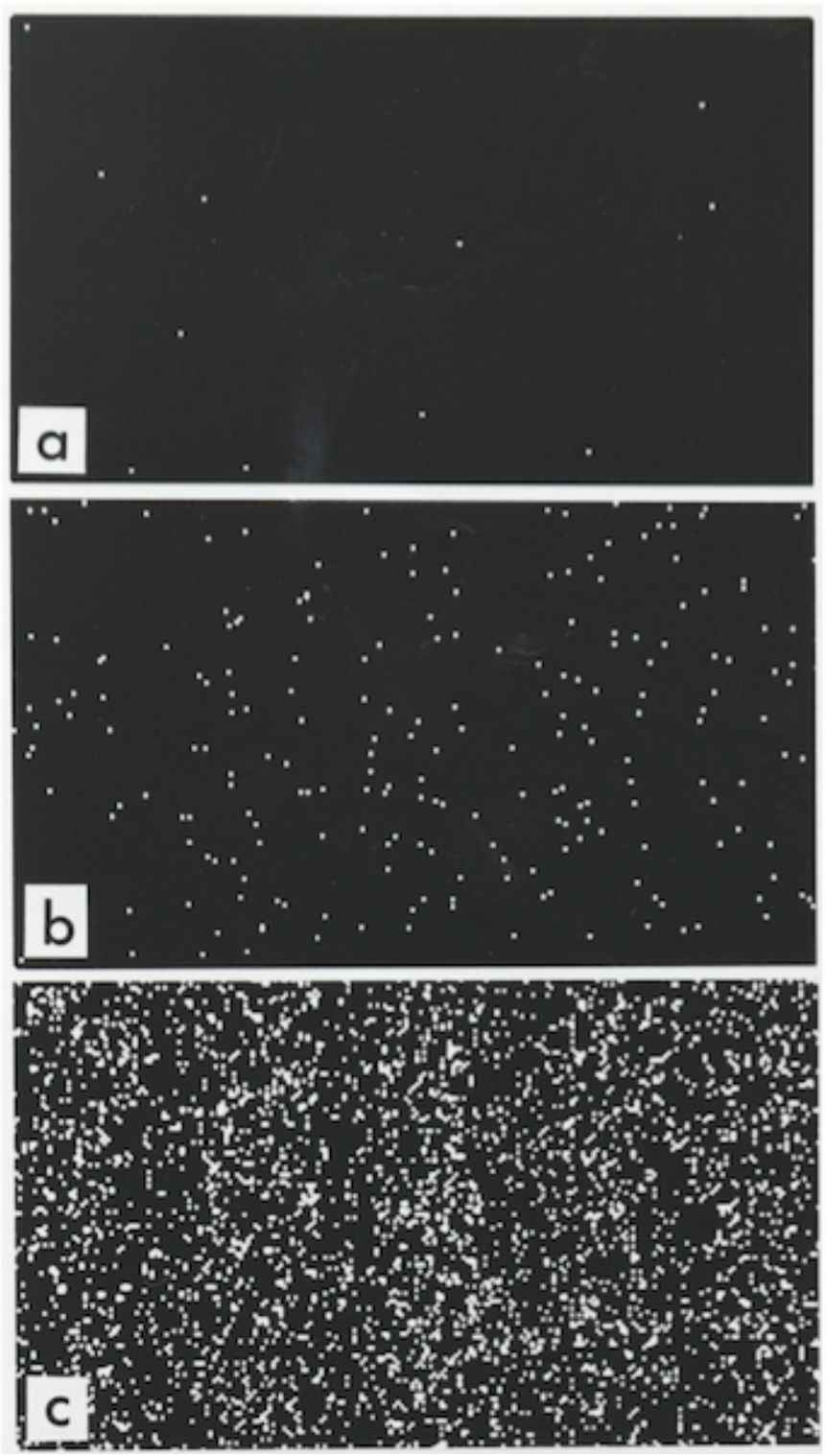
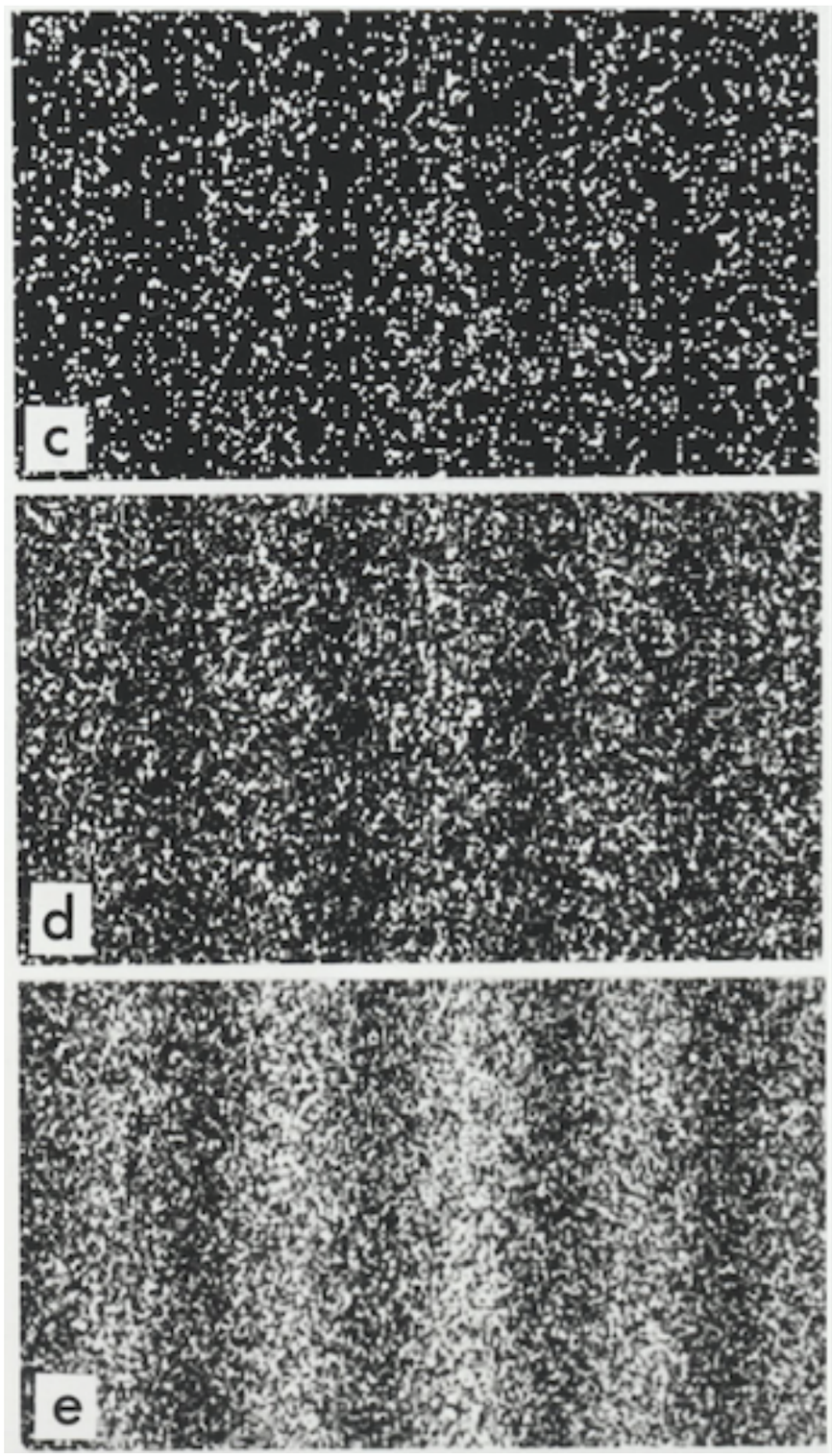
Quantum Computation & Cryptography
Day 1

What's quantum about quantum computing?

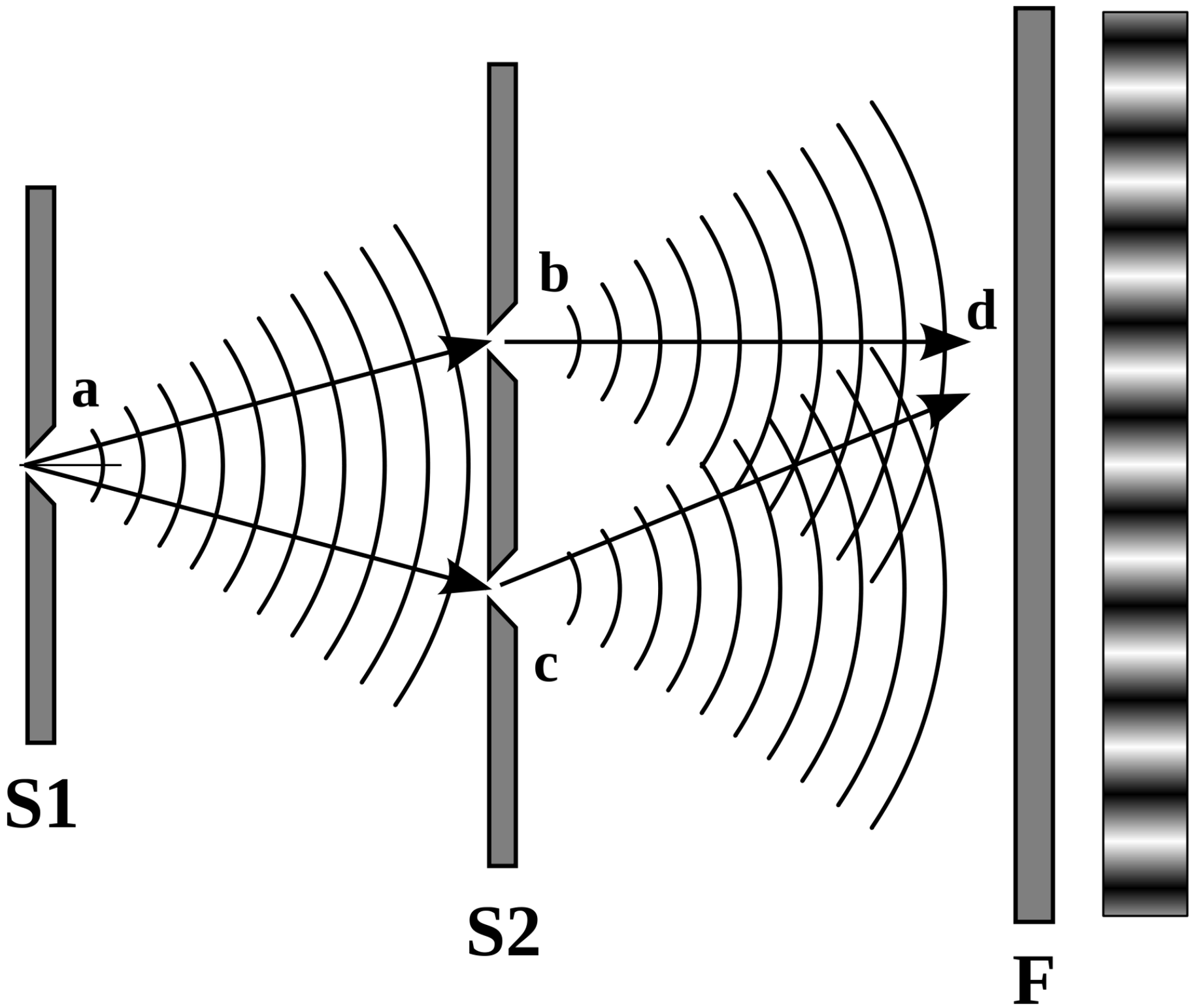
Andru Gheorghiu

The answer is **interference**





The answer is **interference**



"*We choose to examine a phenomenon which is impossible, absolutely impossible, to explain in any classical way, and which has in it the heart of quantum mechanics. In reality it contains the only mystery.*"

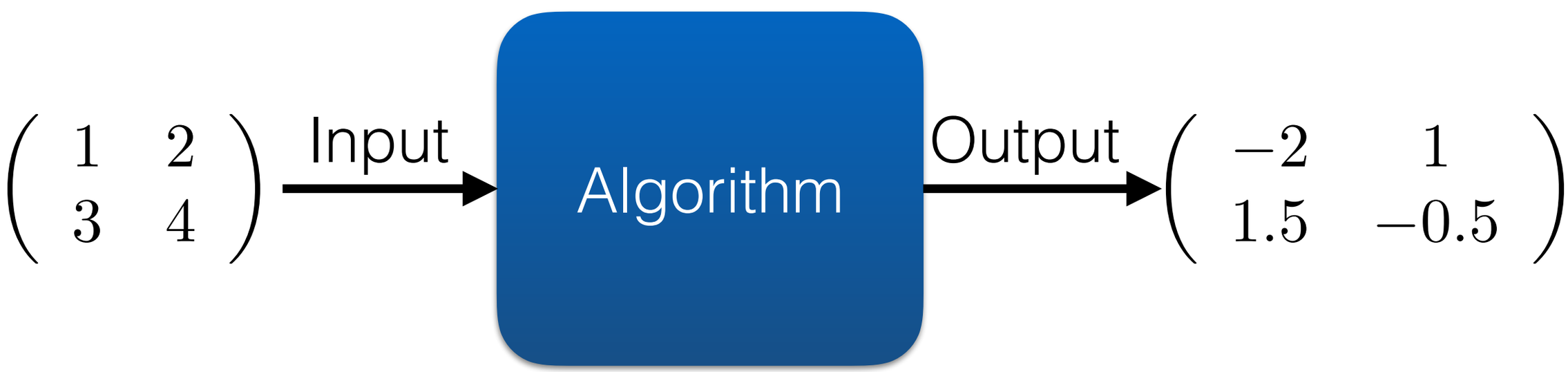
Computation

Computation

Sorting

Computation

Matrix inversion



Computation

Primality testing

Efficient computation

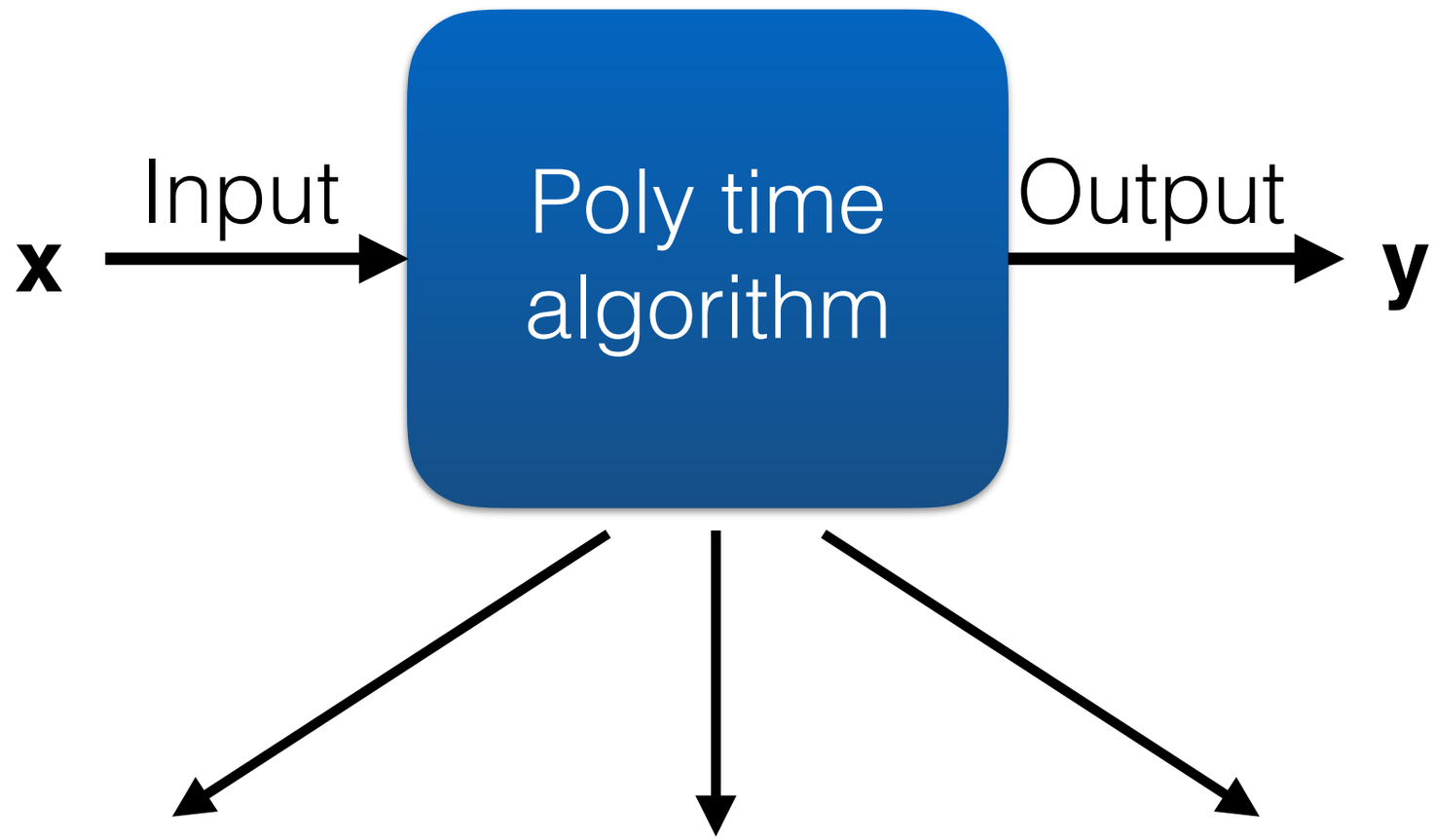
What does efficient algorithm mean?

Number of instructions is a *polynomial* in length of input

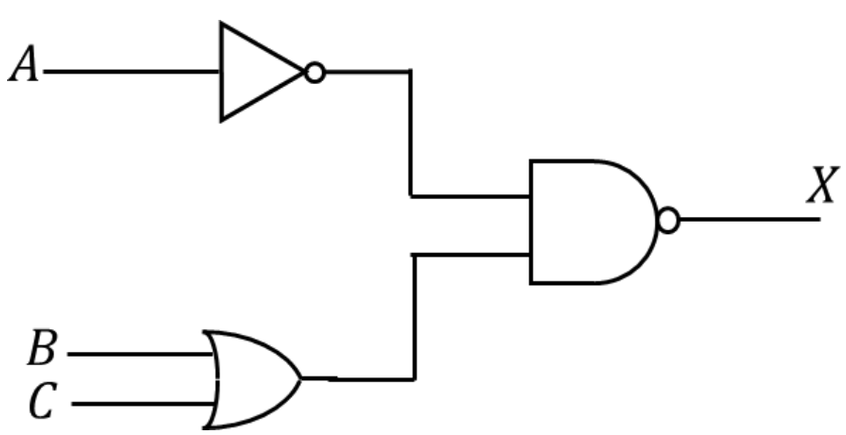
There is a constant d > 0 such that for all **x** running time is $O(|x|^d)$

There are caveats to this definition!

Types of computation

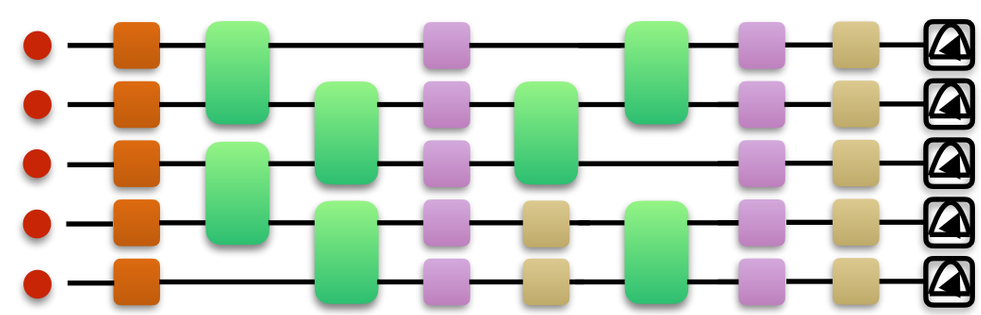


Deterministic



Probabilistic

Quantum



Deterministic computation

Write some code…

```
while (!toVisit.isEmpty()) {
    // remove queue head
    Pair<String, Integer> currentPair = toVisit.remove();
    String currentNode = currentPair.getFst();
    int distance = currentPair.getSnd();

    // reached destination?
    if (currentNode.equals(dest))
        return distance;

    // already visited this node, skip
    if (visited.contains(currentNode)) continue;

    // add neighbors to queue, to be visited later on
    for (String neighbor : neighbors.get(currentNode)) {
        // reached destination?
        if (neighbor.equals(dest))
            return distance + 1;
        toVisit.add(new Pair<String, Integer>(neighbor, distance + 1));
    }
    visited.add(currentNode);
}
```

Deterministic computation

Compile the code…

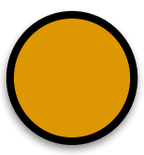
```
00000000    push    ebp
00000001    mov     ebp, esp
00000003    movzx   ecx, [ebp+arg_0]
00000007    pop     ebp
00000008    movzx   dx, cl
0000000C    lea     eax, [edx+edx]
0000000F    add     eax, edx
00000011    shl     eax, 2
00000014    add     eax, edx
00000016    shr     eax, 8
00000019    sub     cl, al
0000001B    shr     cl, 1
0000001D    add     al, cl
0000001F    shr     al, 5
00000022    movzx   eax, al
00000025    retn
```

Deterministic computation

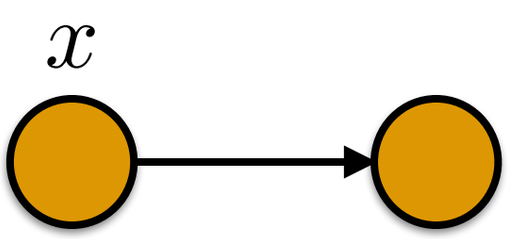
Run the code…

Deterministic computation

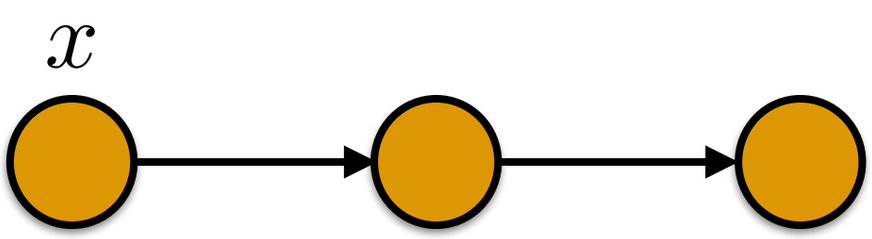
x



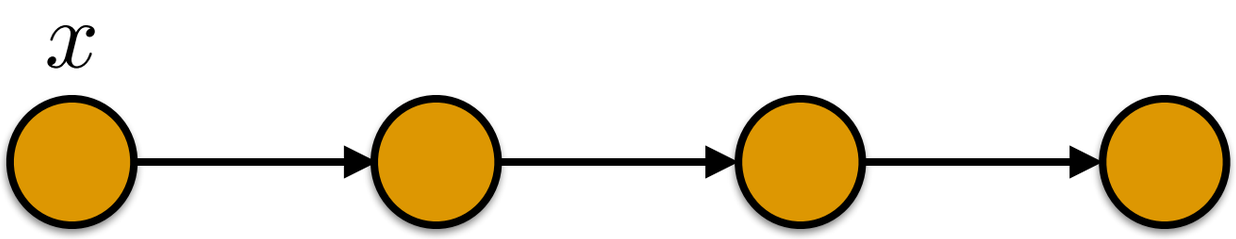
Deterministic computation



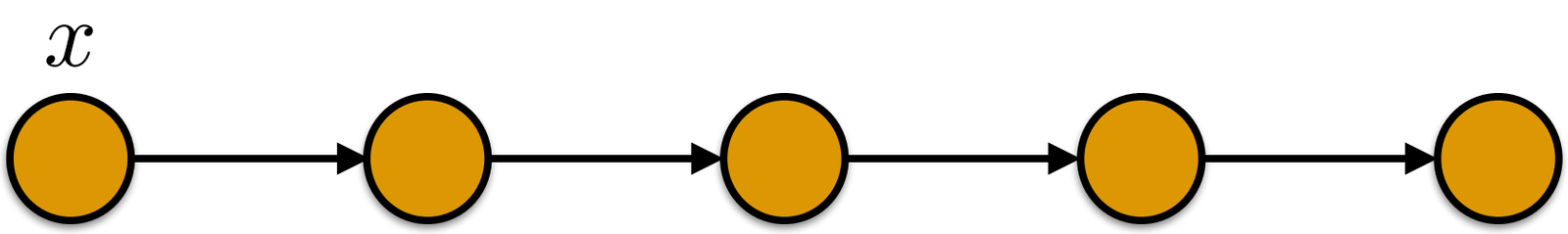
Deterministic computation



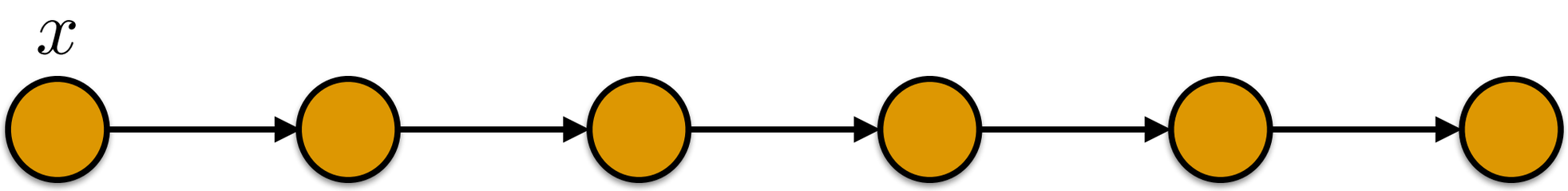
Deterministic computation



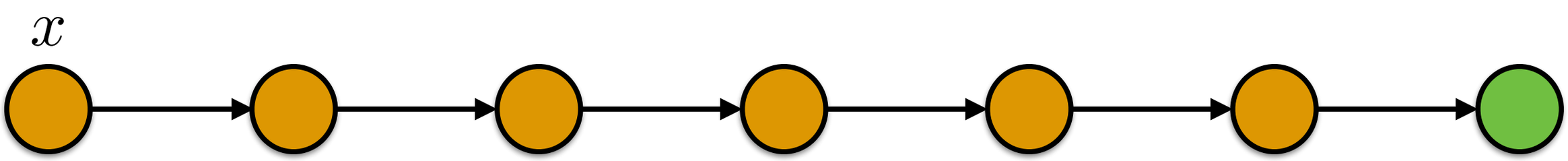
Deterministic computation



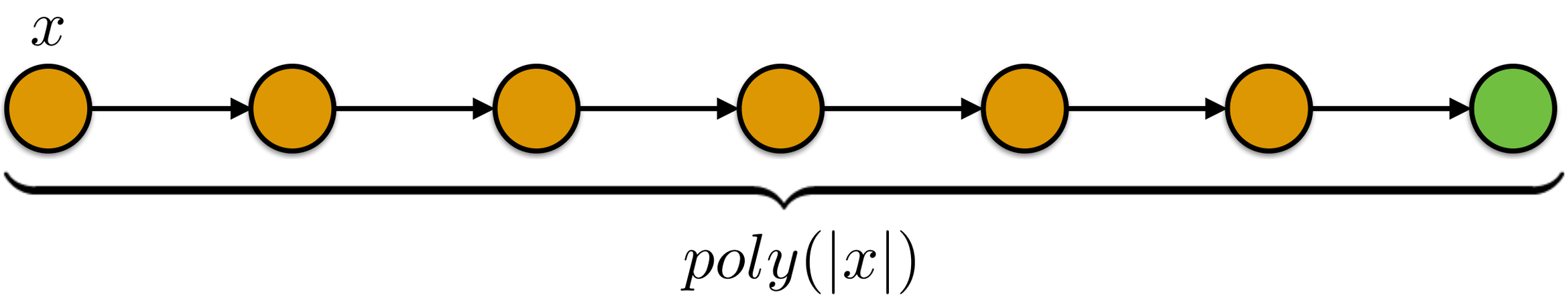
Deterministic computation



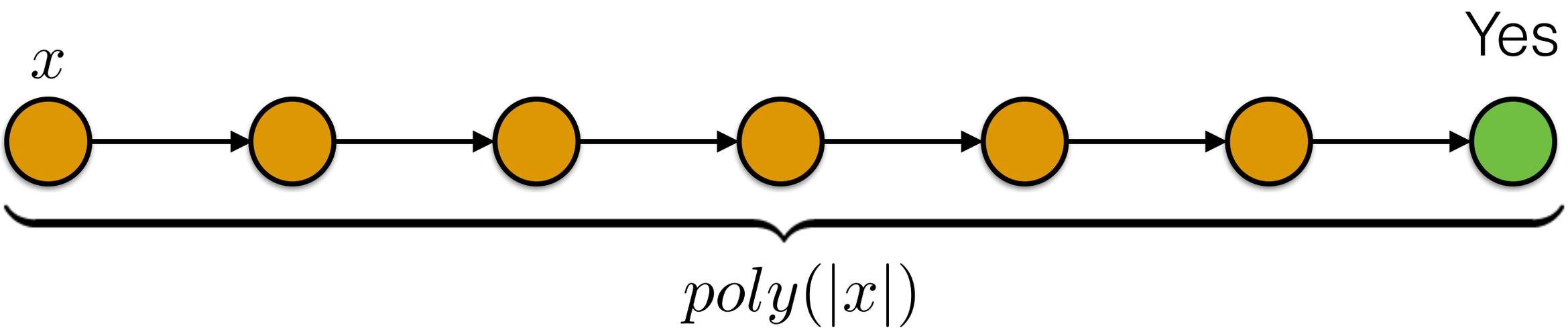
Deterministic computation



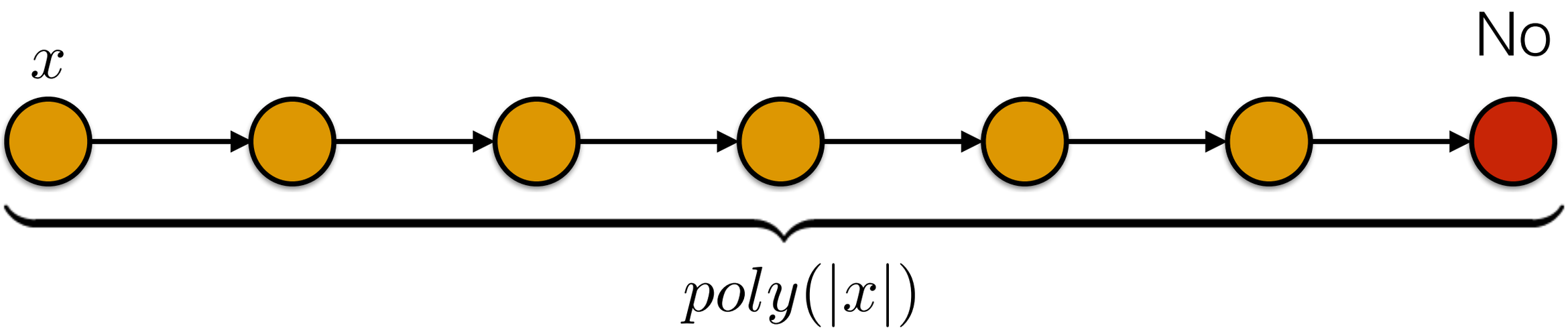
Deterministic computation



Deterministic computation



Deterministic computation



Probabilistic computation

Let’s add some randomness

Probabilistic computation

Let's add some randomness

As if computer flips a coin every now and then

Probabilistic computation

Let's add some randomness

As if computer flips a coin every now and then

We allow the algorithm to err some of the time

Probabilistic computation

Let's add some randomness

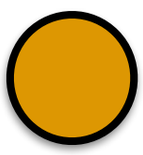
As if computer flips a coin every now and then

We allow the algorithm to err some of the time

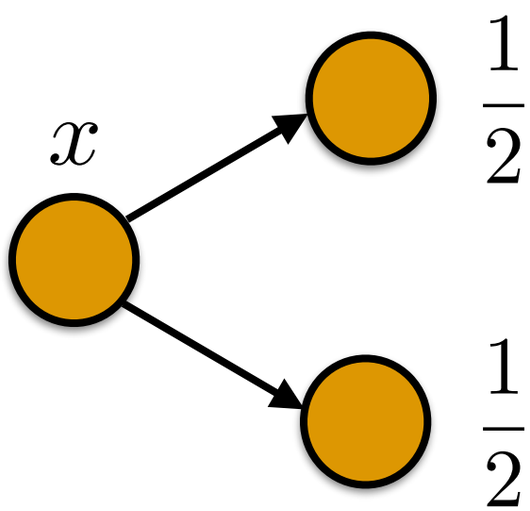
Should obtain correct answer <u>with high probability</u>

Probabilistic computation

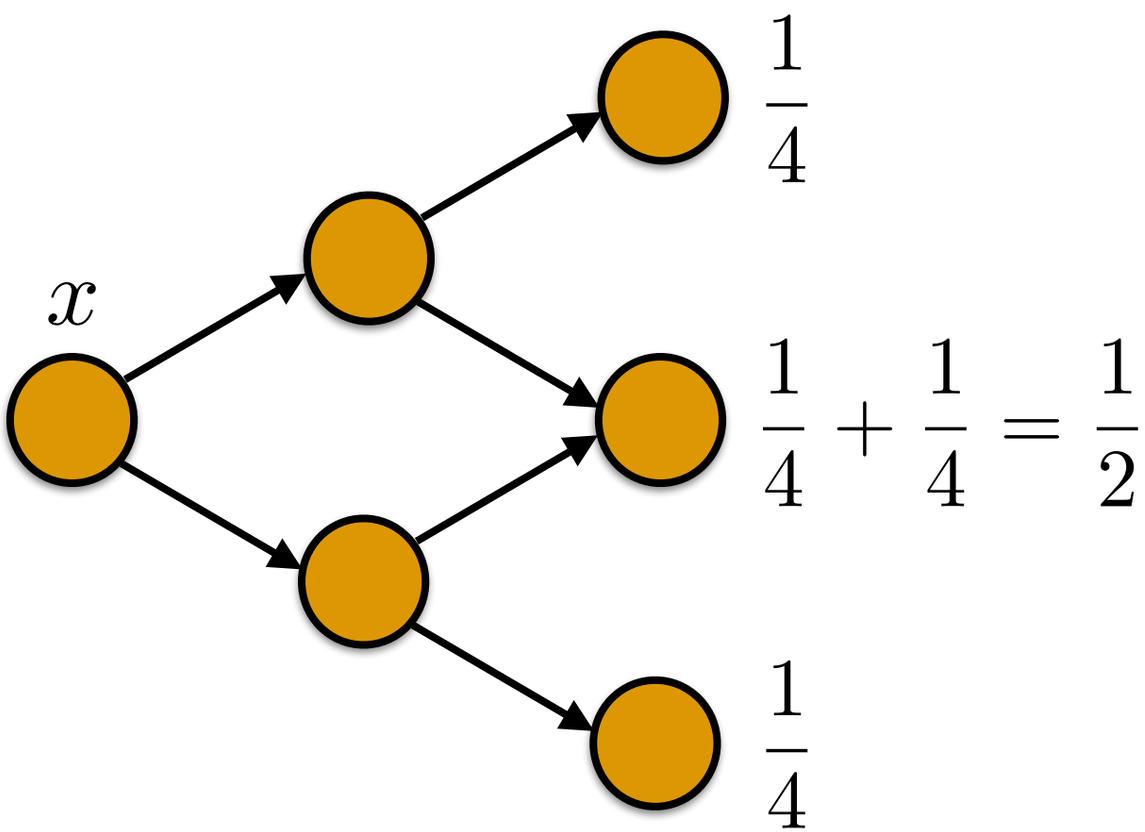
x



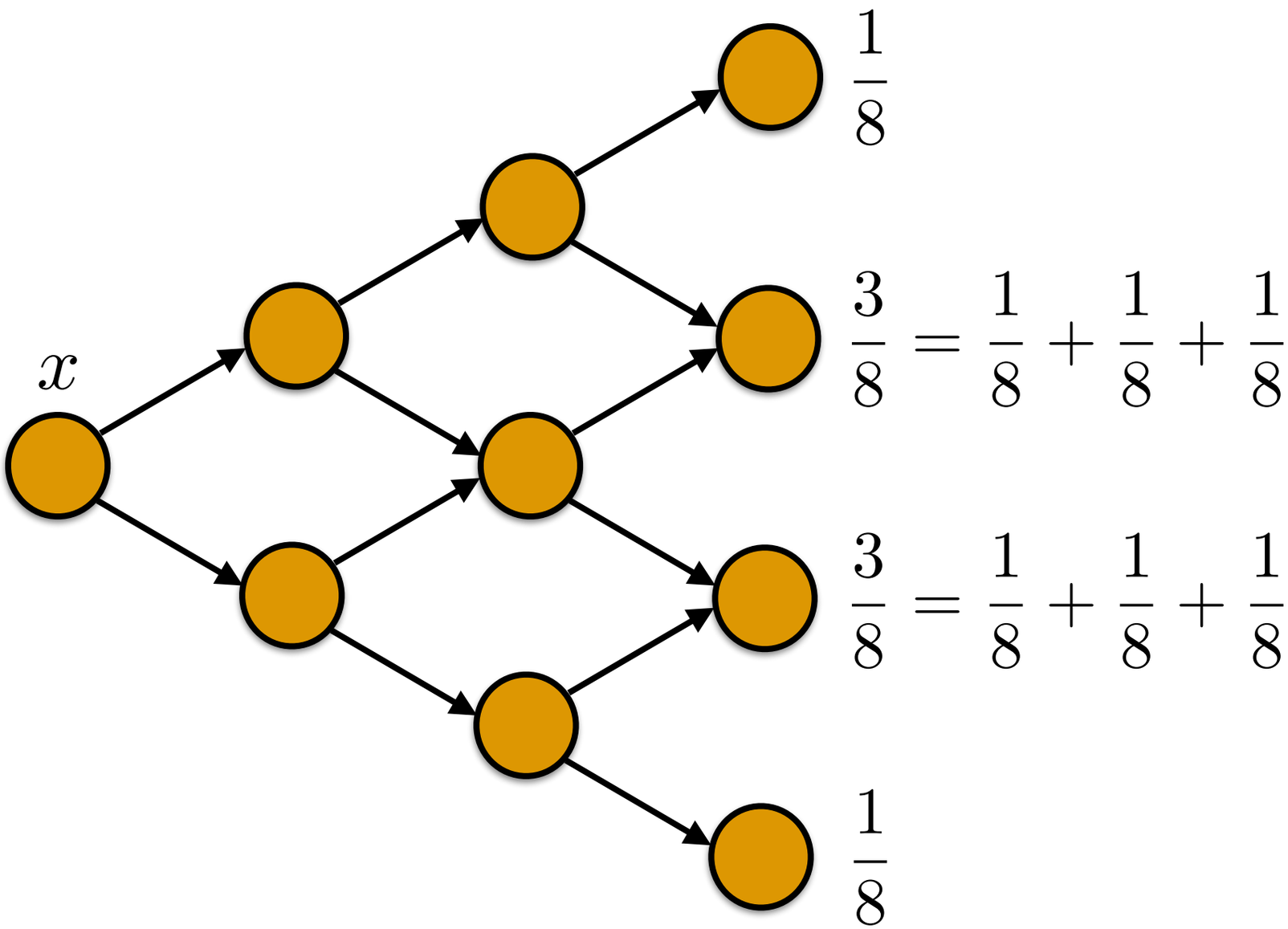
Probabilistic computation



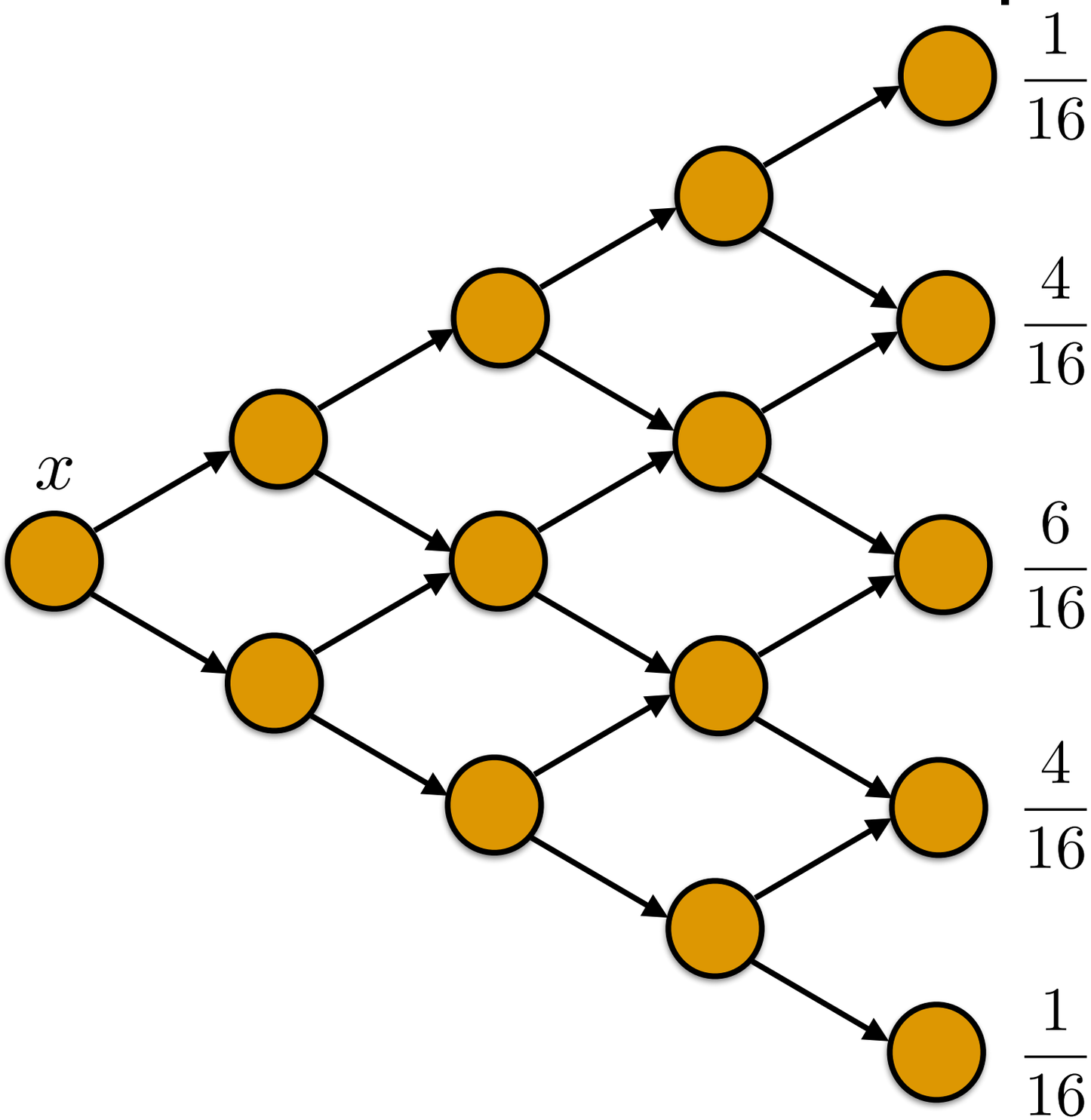
Probabilistic computation



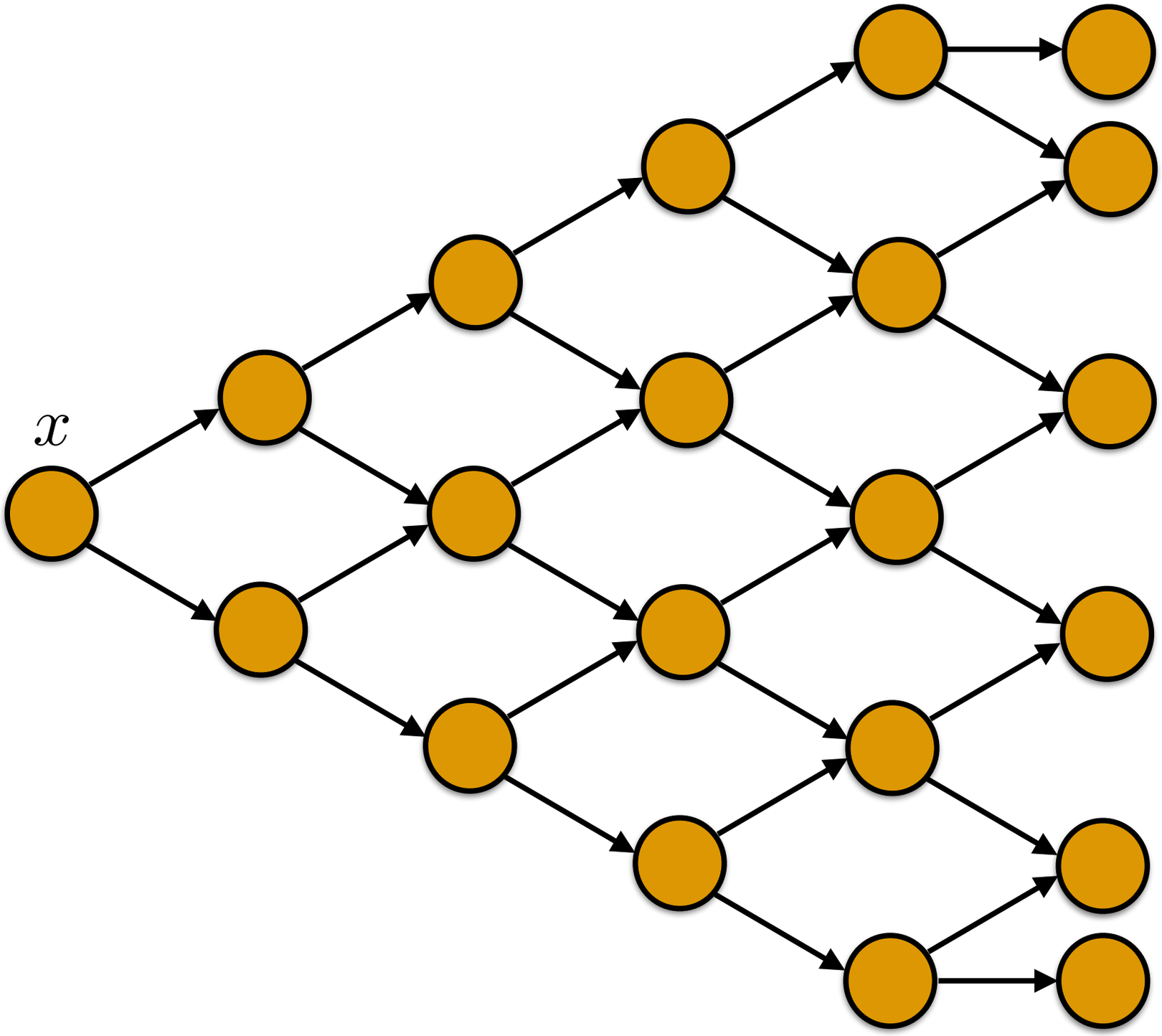
Probabilistic computation



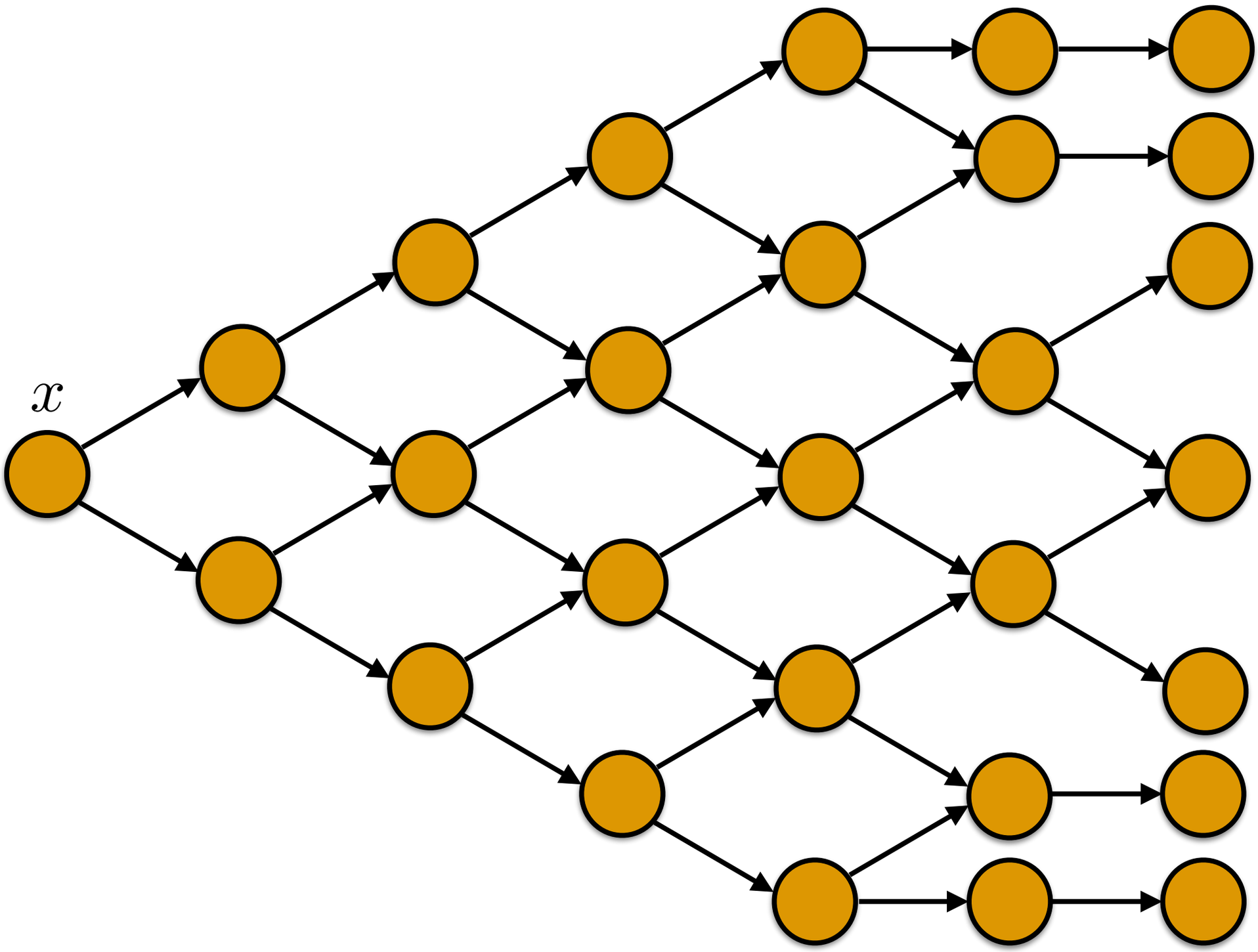
Probabilistic computation



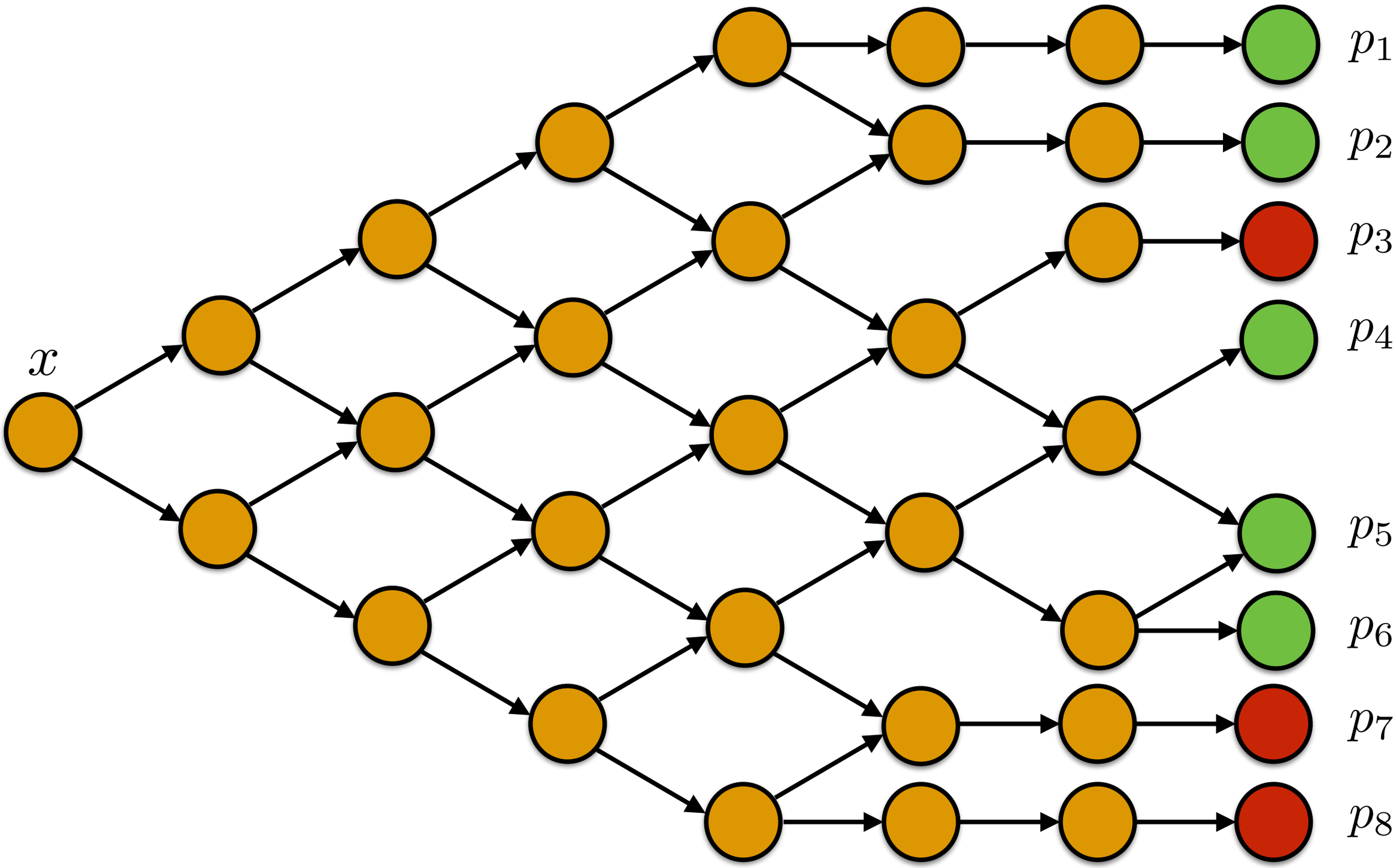
Probabilistic computation



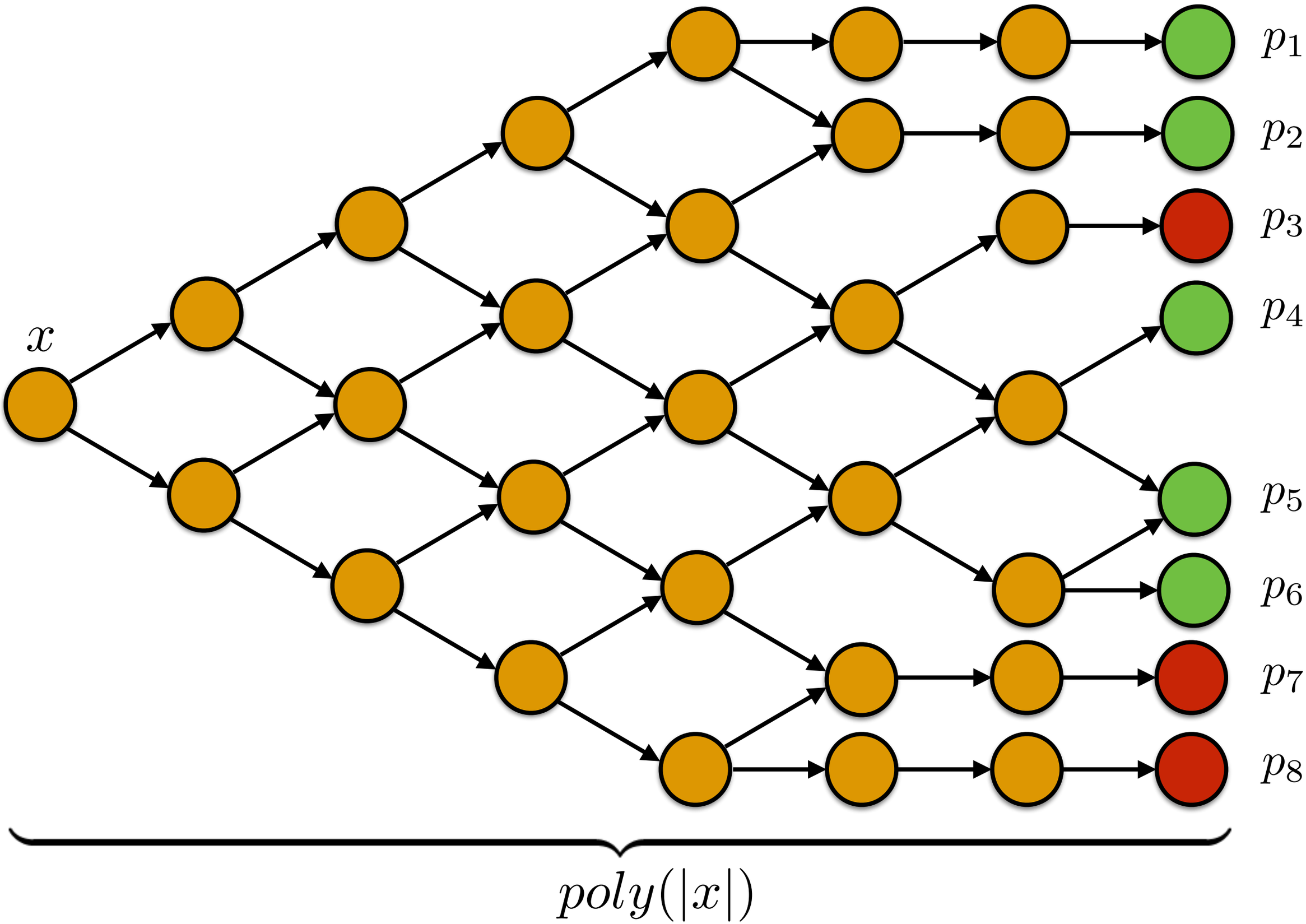
Probabilistic computation



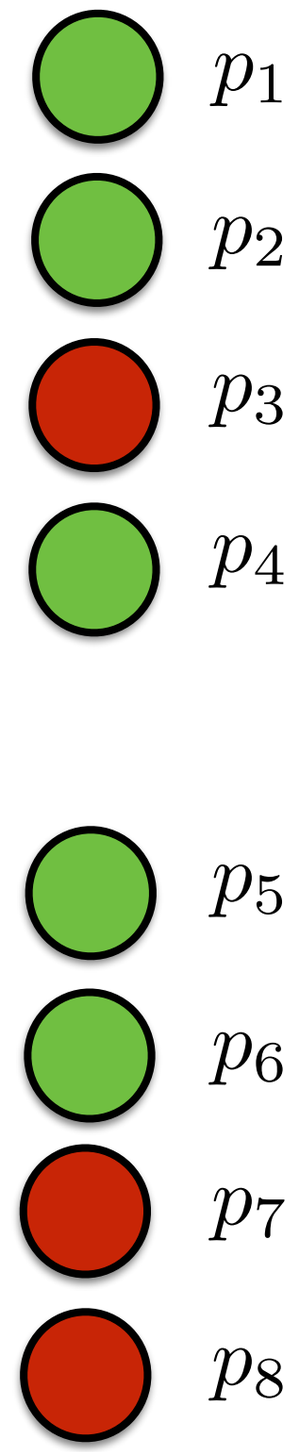
Probabilistic computation



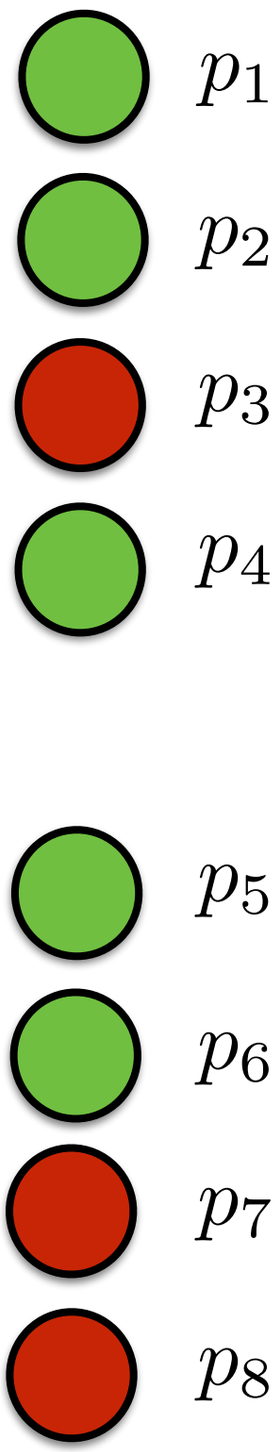
Probabilistic computation



Probabilistic computation

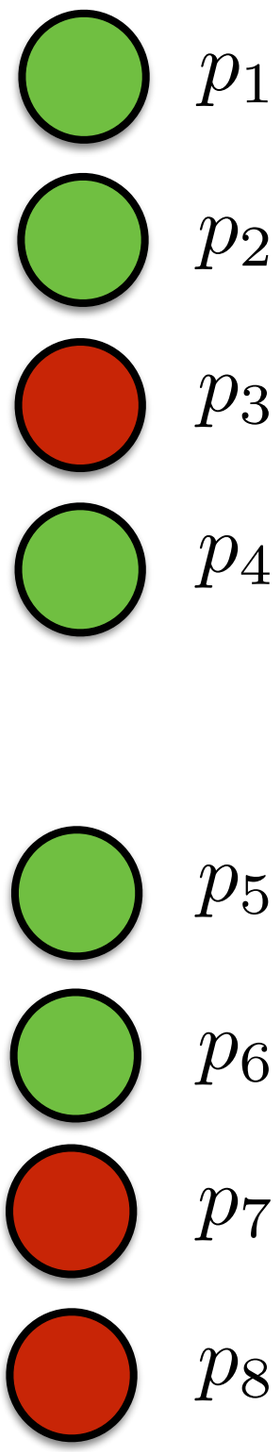


Probabilistic computation



$$Pr(YES) = p_1 + p_2 + p_4 + p_5 + p_6$$

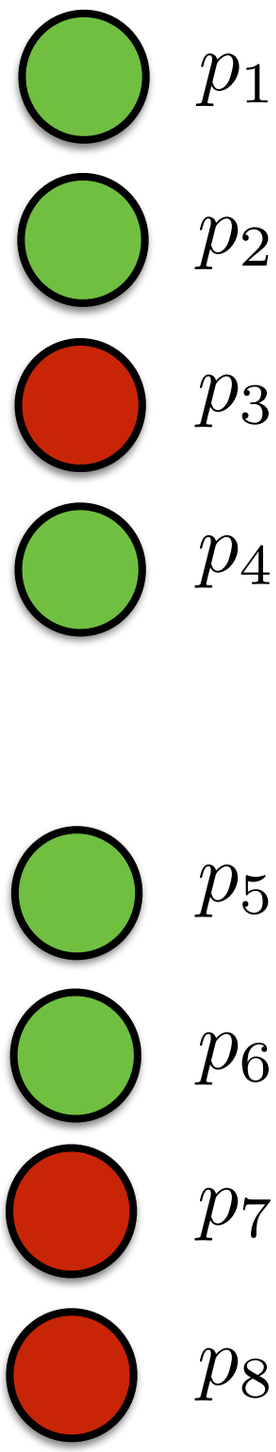
Probabilistic computation



$$Pr(YES) = p_1 + p_2 + p_4 + p_5 + p_6$$

$$Pr(NO) = p_3 + p_7 + p_8$$

Probabilistic computation

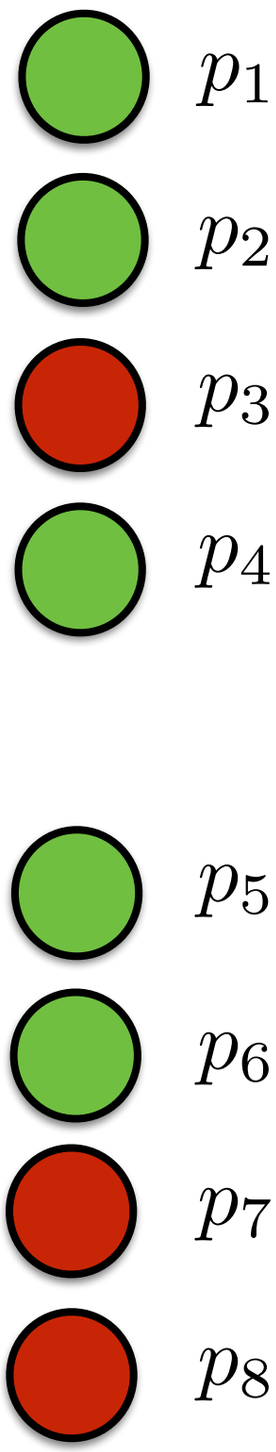


$$Pr(YES) = p_1 + p_2 + p_4 + p_5 + p_6$$

$$Pr(NO) = p_3 + p_7 + p_8$$

$$Pr(YES) + Pr(NO) = 1$$

Probabilistic computation



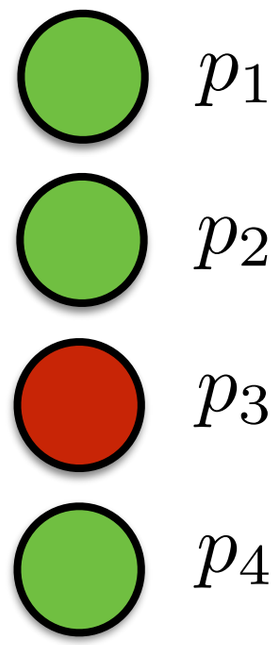
$$Pr(YES) = p_1 + p_2 + p_4 + p_5 + p_6$$

$$Pr(NO) = p_3 + p_7 + p_8$$

$$Pr(YES) + Pr(NO) = 1$$

When answer is YES, Pr(YES) > 2/3

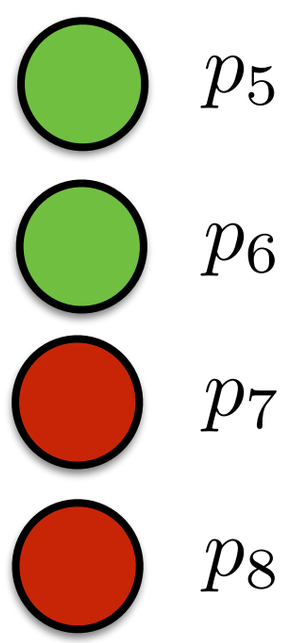
Probabilistic computation



$$Pr(YES) = p_1 + p_2 + p_4 + p_5 + p_6$$

$$Pr(NO) = p_3 + p_7 + p_8$$

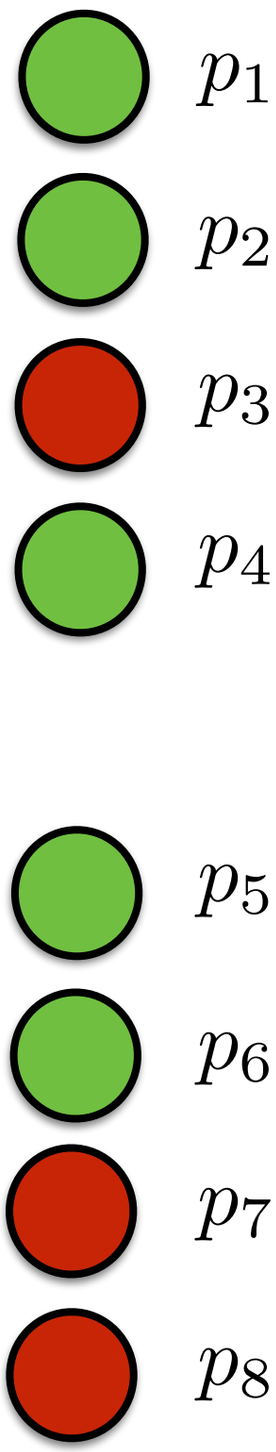
$$Pr(YES) + Pr(NO) = 1$$



When answer is YES, Pr(YES) > 2/3

When answer is NO, Pr(NO) > 2/3

Probabilistic computation



$$Pr(YES) = p_1 + p_2 + p_4 + p_5 + p_6$$

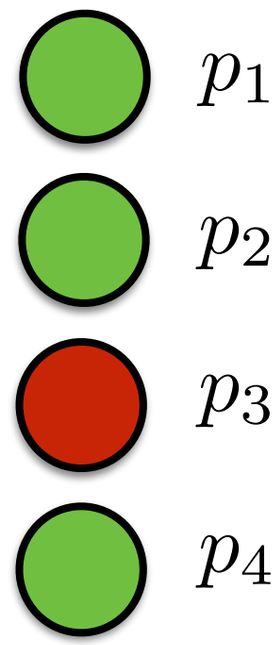
$$Pr(NO) = p_3 + p_7 + p_8$$

$$Pr(YES) + Pr(NO) = 1$$

When answer is YES, Pr(YES) > 2/3

When answer is NO, Pr(YES) < 1/3

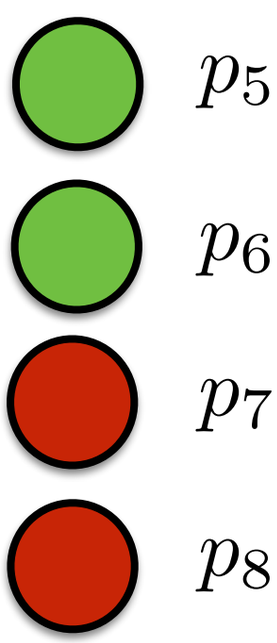
Probabilistic computation



$$Pr(YES) = p_1 + p_2 + p_4 + p_5 + p_6$$

$$Pr(NO) = p_3 + p_7 + p_8$$

$$Pr(YES) + Pr(NO) = 1$$



When answer is YES, Pr(YES) > 2/3

When answer is NO, Pr(YES) < 1/3

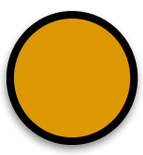
This defines a probabilistic poly time computation!

Quantum computation

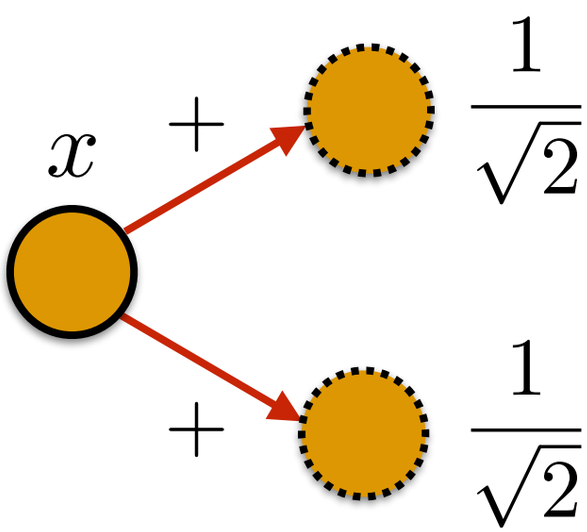
Quantum computers use “quantum probabilities”
(i.e. amplitudes)

Quantum computation

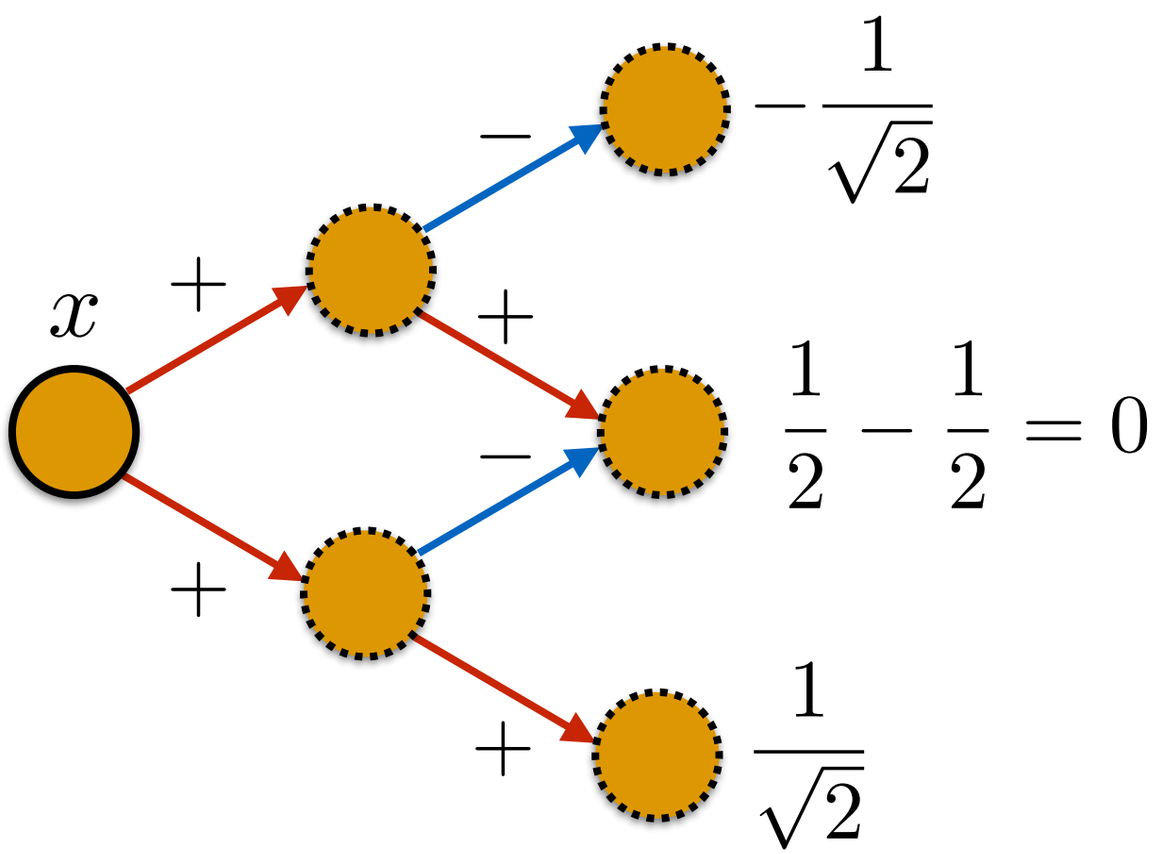
x



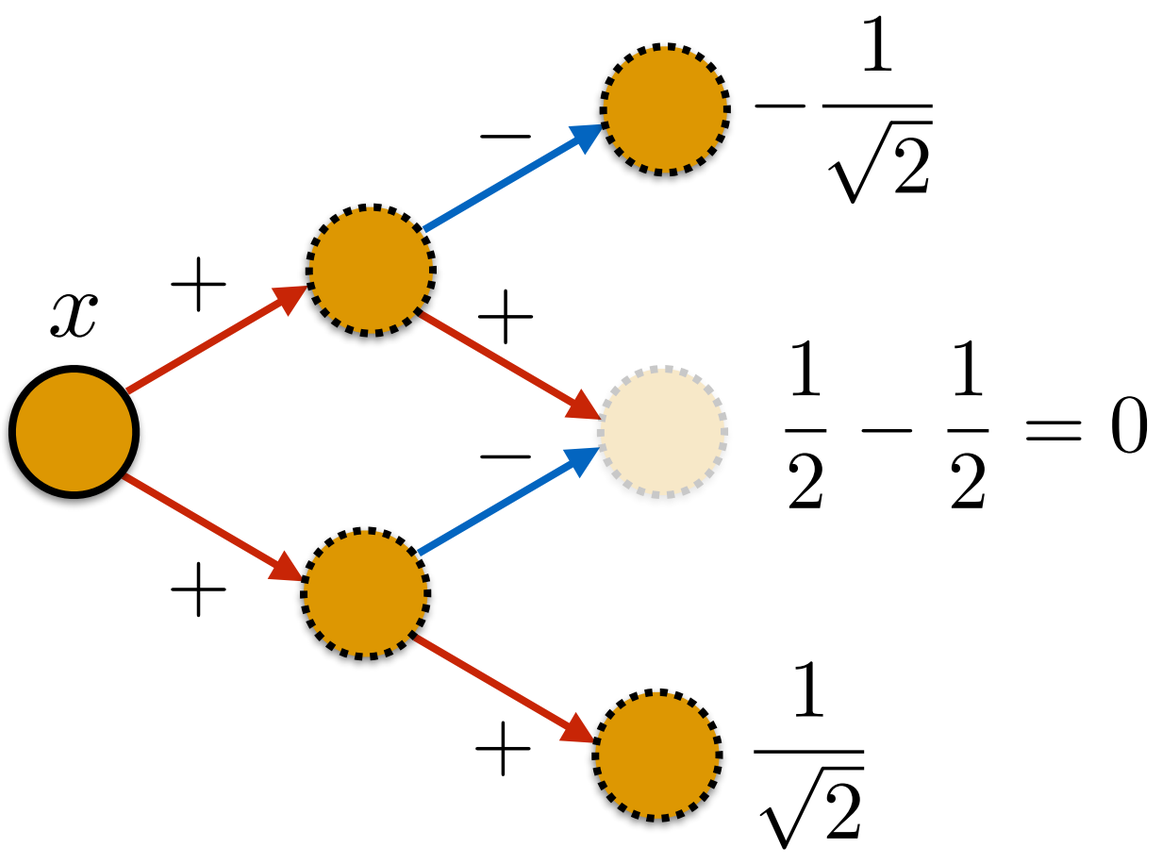
Quantum computation



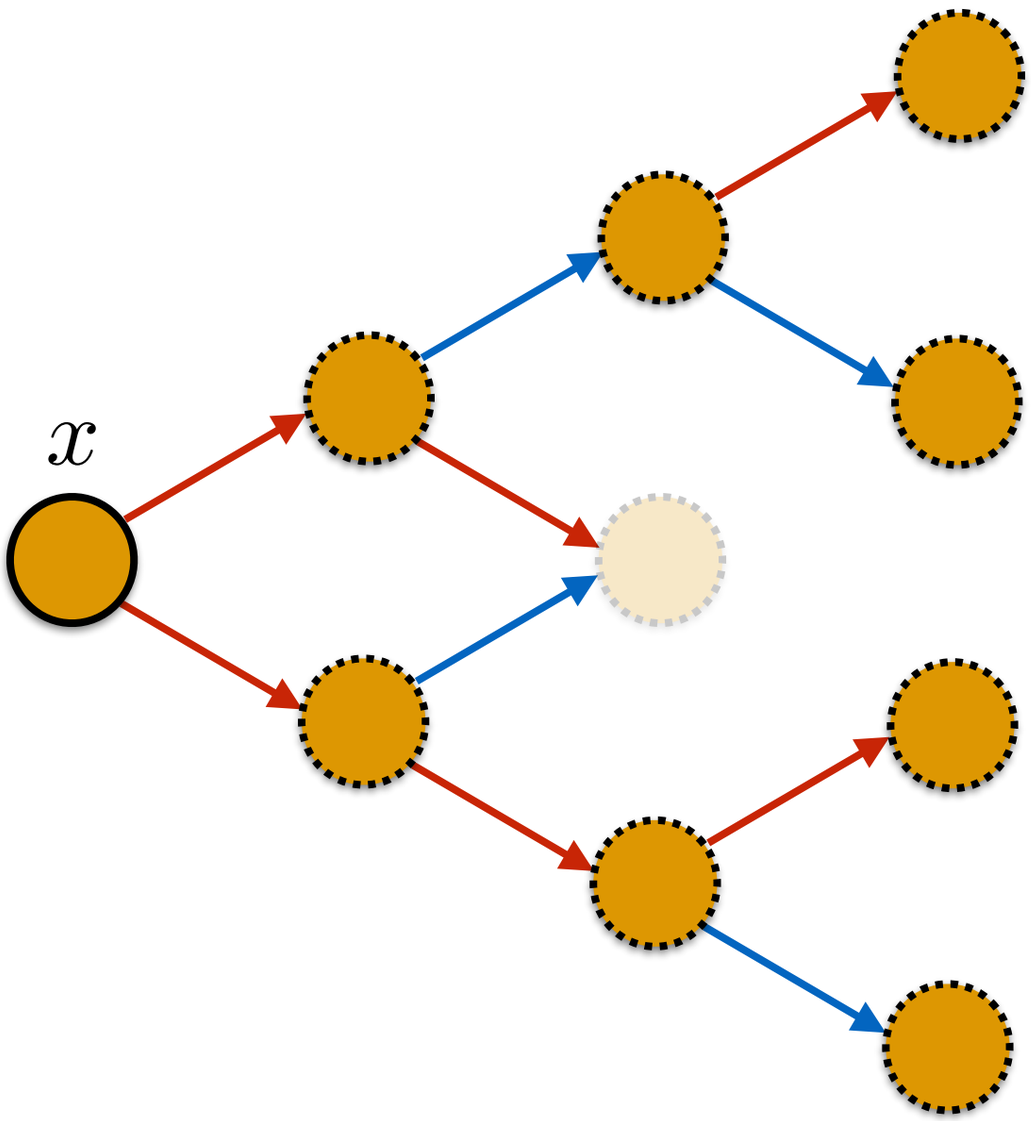
Quantum computation



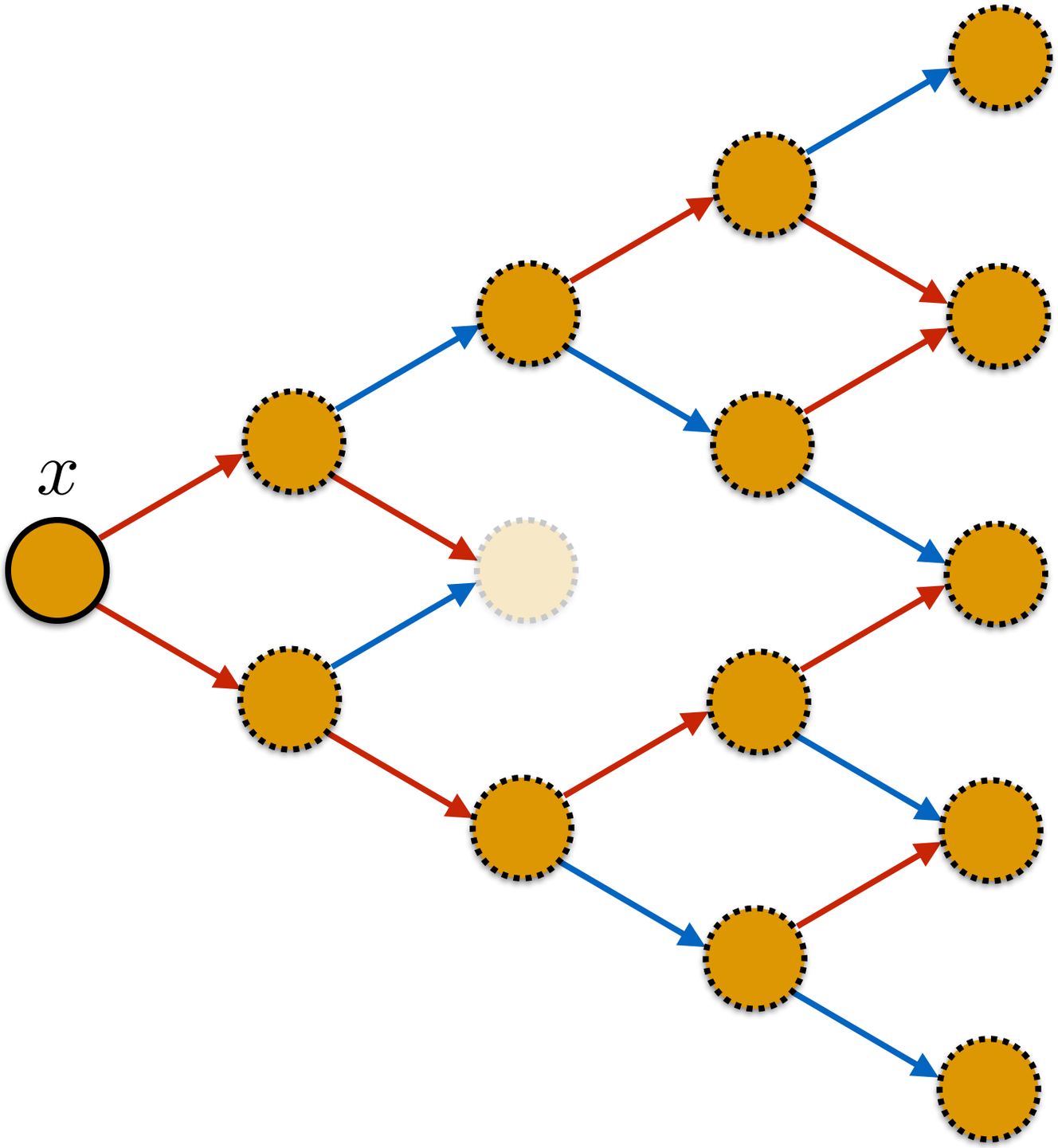
Quantum computation



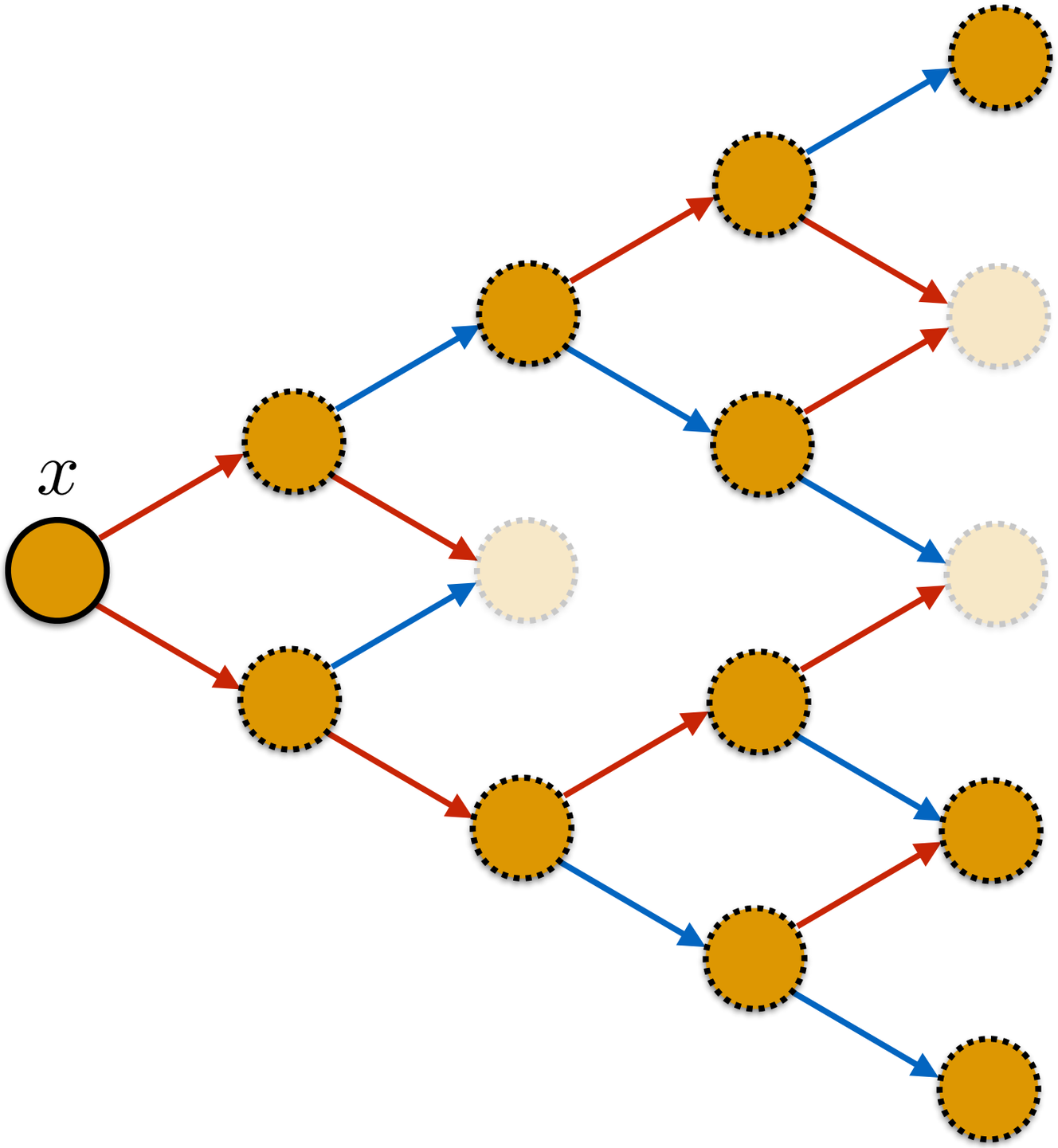
Quantum computation



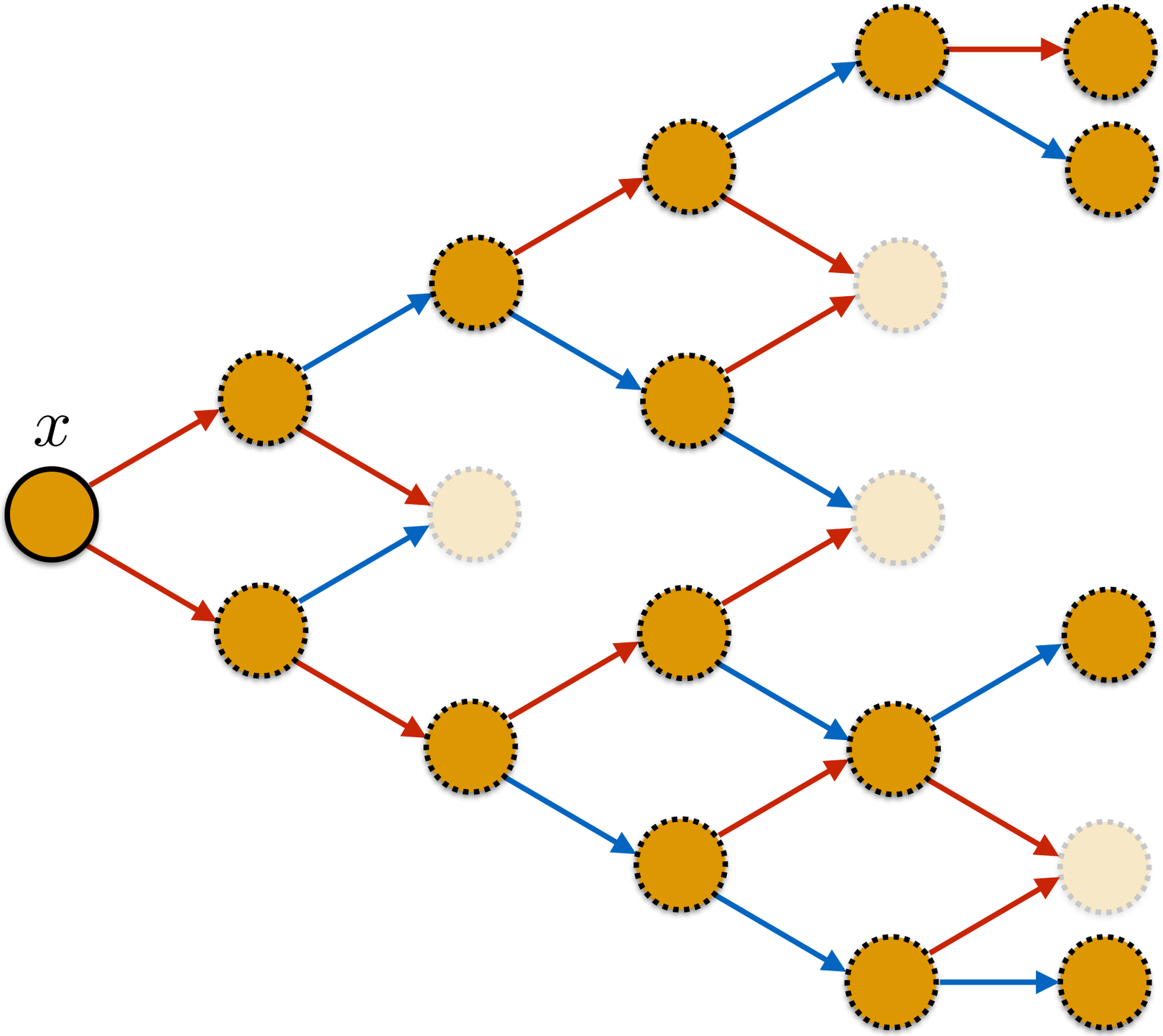
Quantum computation



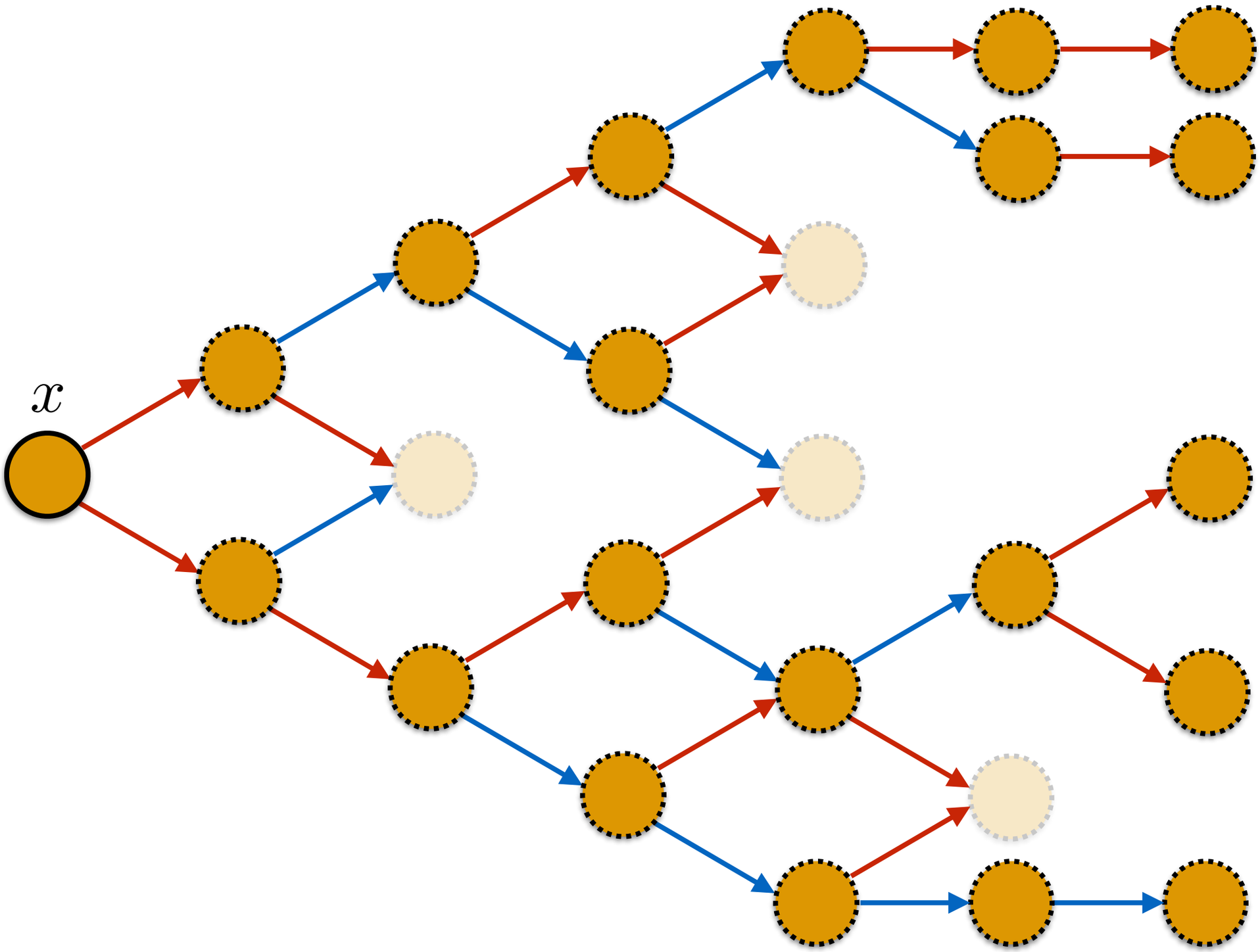
Quantum computation



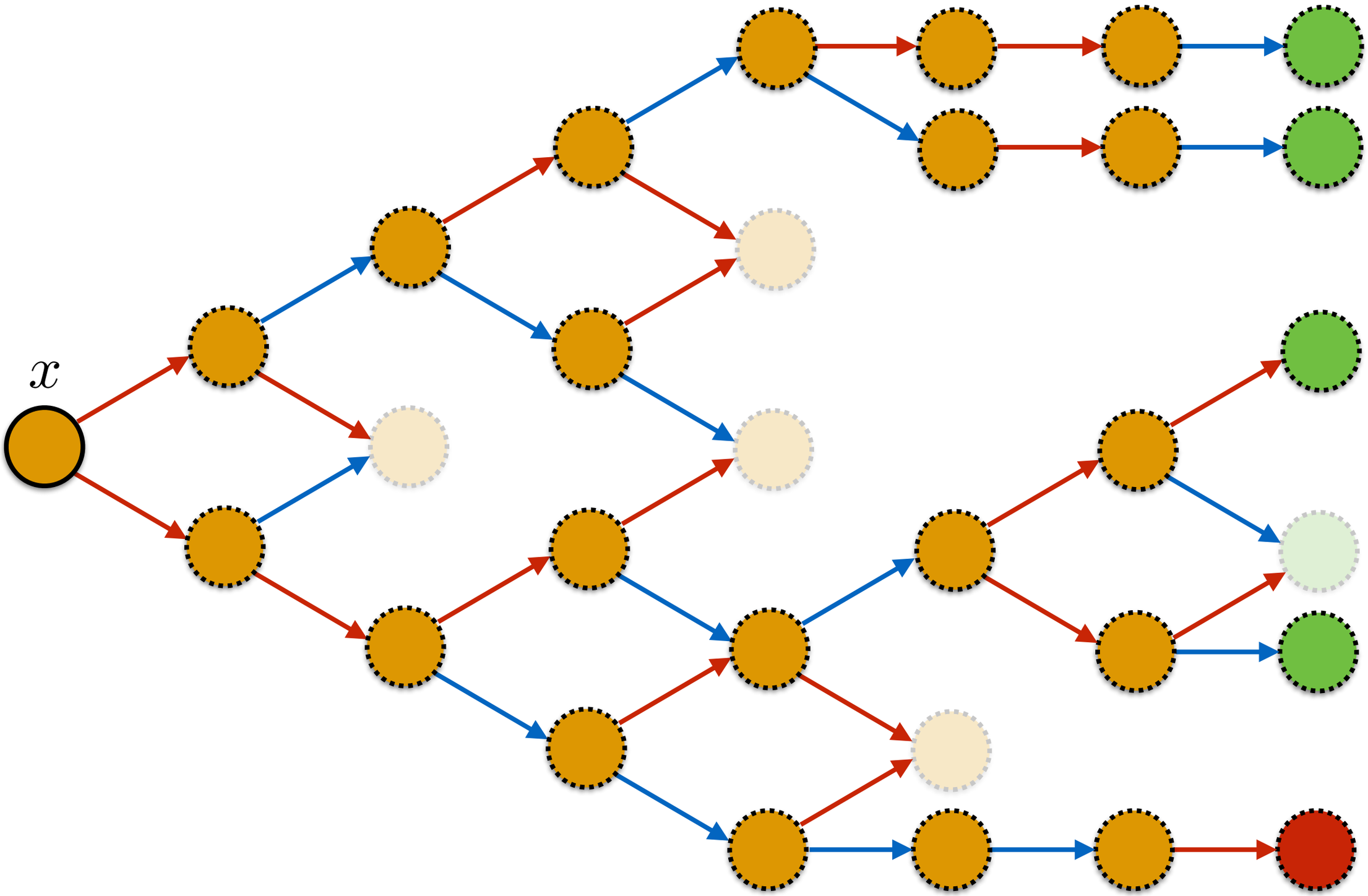
Quantum computation



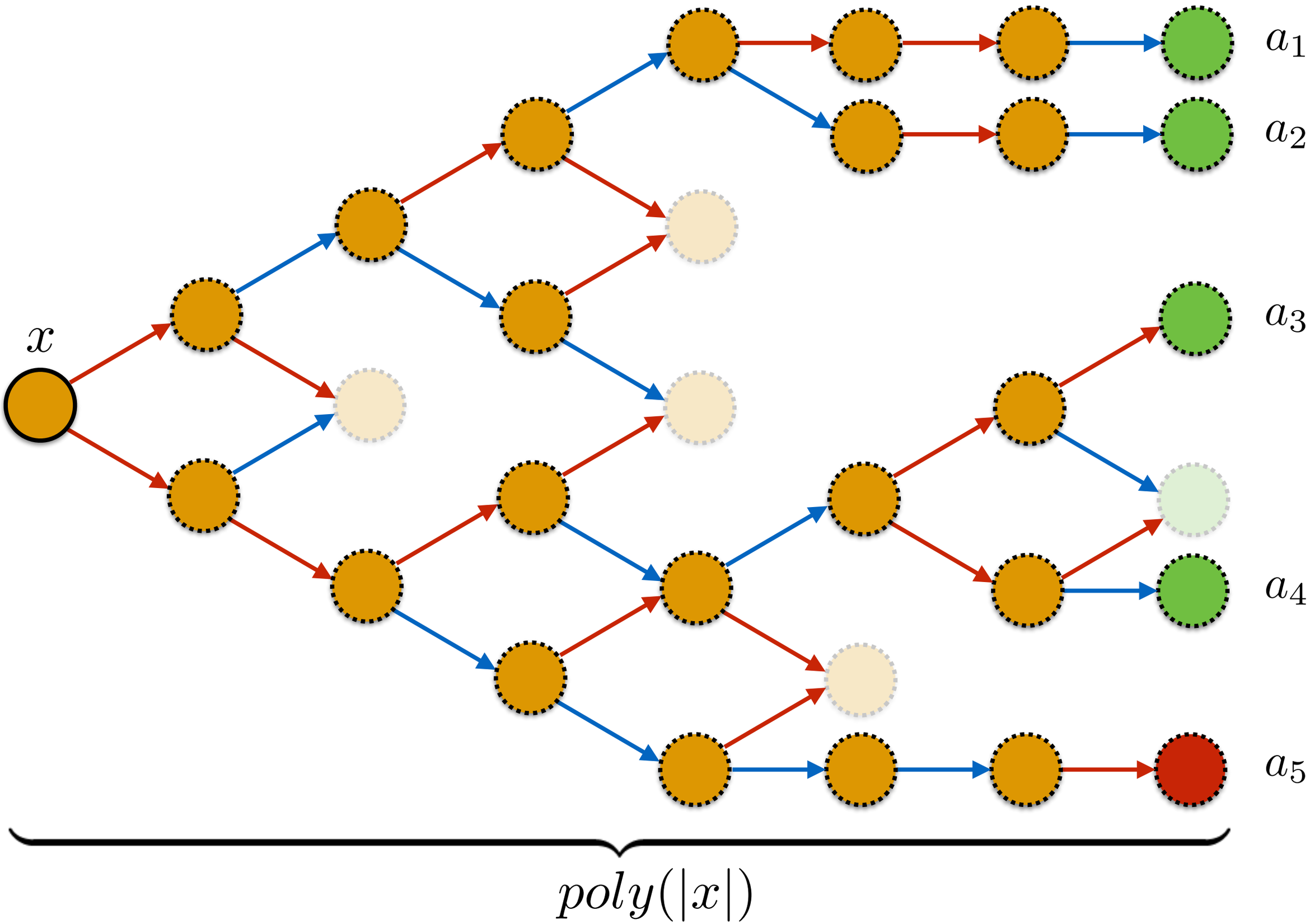
Quantum computation



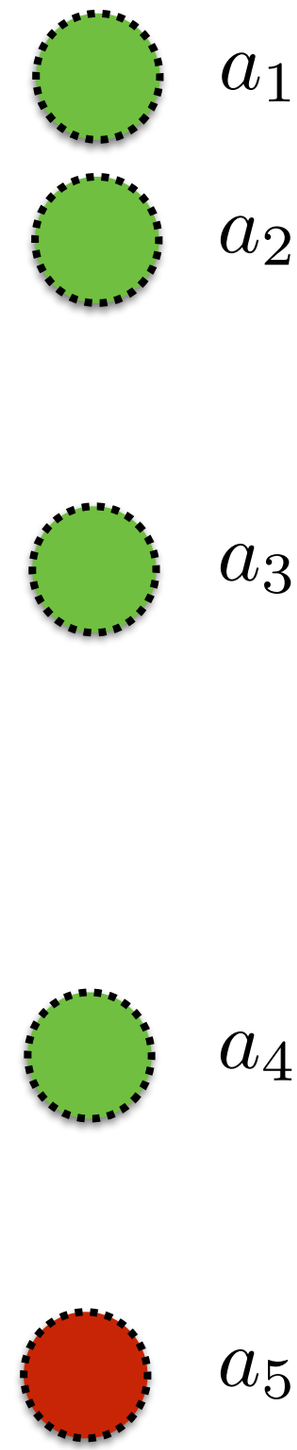
Quantum computation



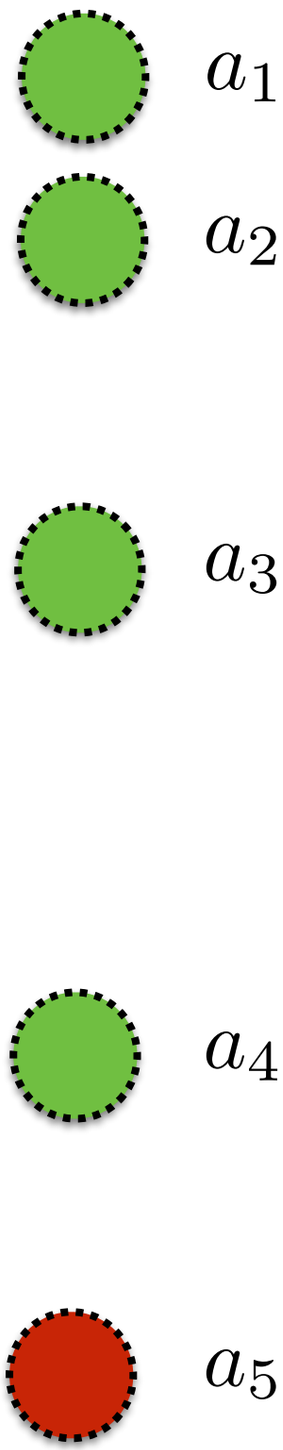
Quantum computation



Quantum computation

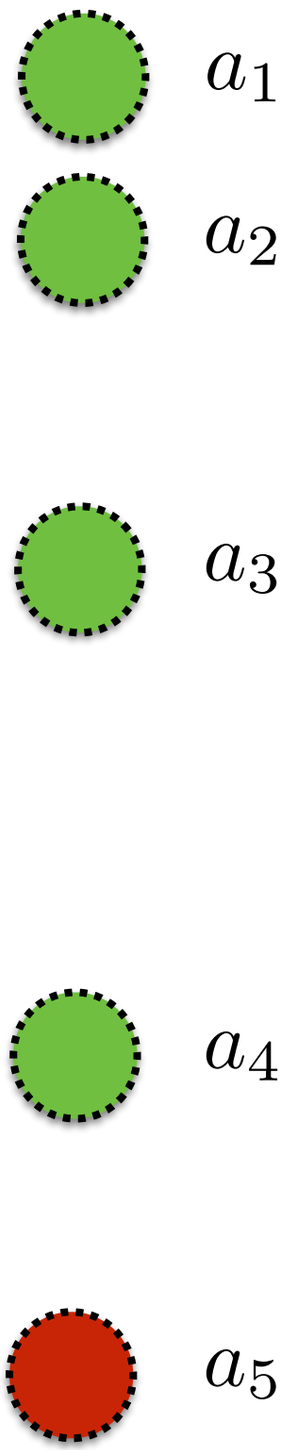


Quantum computation



$$Amp(YES) = a_1 + a_2 + a_3 + a_4$$

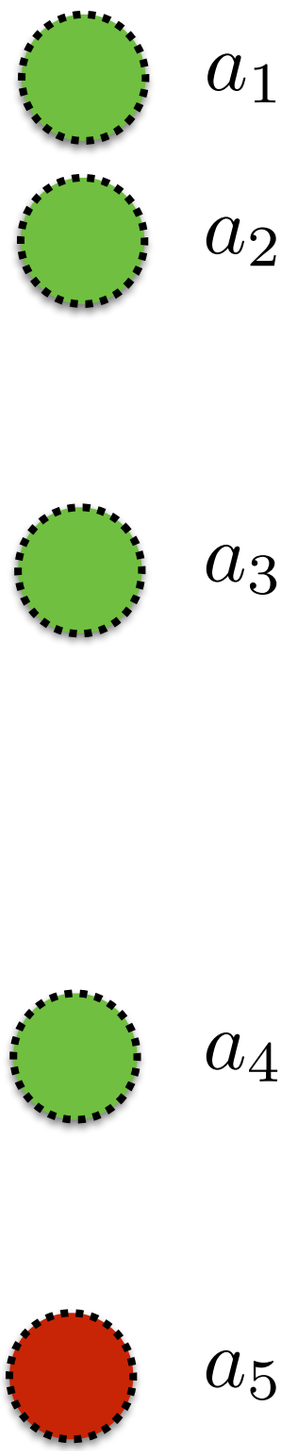
Quantum computation



$$Amp(YES) = a_1 + a_2 + a_3 + a_4$$

$$Pr(YES) = |a_1 + a_2 + a_3 + a_4|^2$$

Quantum computation

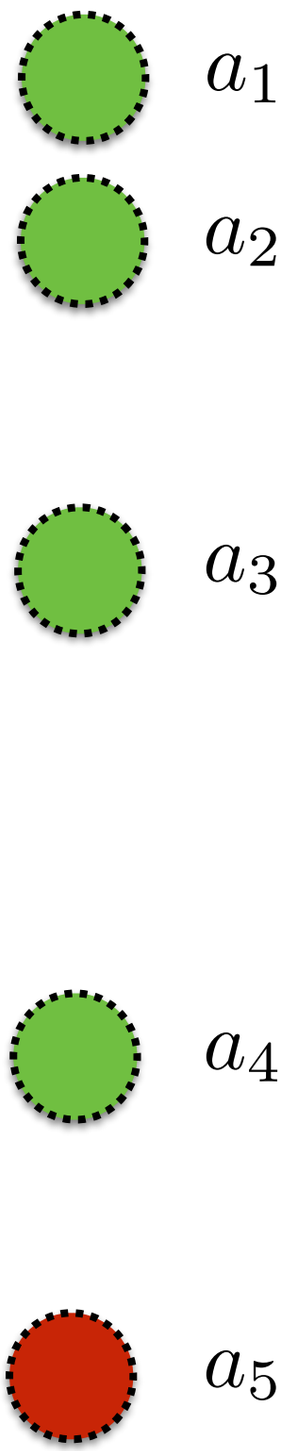


$$Amp(YES) = a_1 + a_2 + a_3 + a_4$$

$$Pr(YES) = |a_1 + a_2 + a_3 + a_4|^2$$

$$Pr(YES) = \left| \sum_{path \in YES-Paths} a_{path} \right|^2$$

Quantum computation



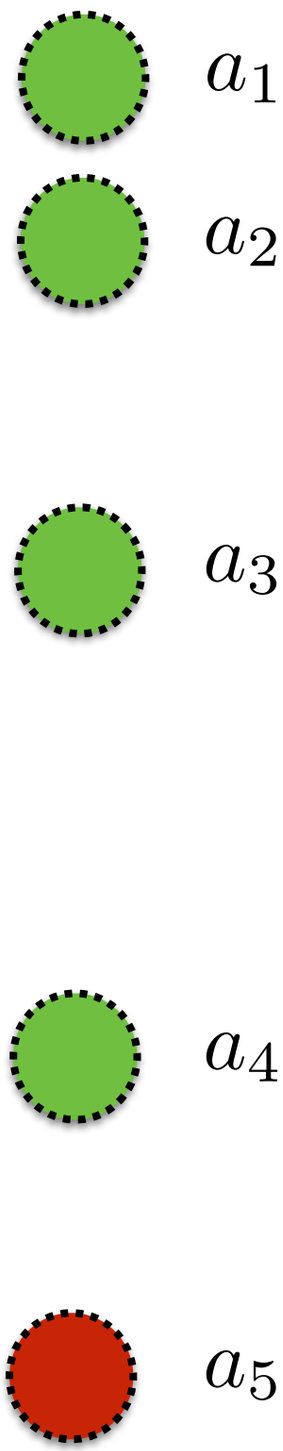
$$Amp(YES) = a_1 + a_2 + a_3 + a_4$$

$$Pr(YES) = |a_1 + a_2 + a_3 + a_4|^2$$

$$Pr(YES) = \left| \sum_{path \in YES-Paths} a_{path} \right|^2$$

$$Pr(NO) = \left| \sum_{path \in NO-Paths} a_{path} \right|^2$$

Quantum computation



$$Amp(YES) = a_1 + a_2 + a_3 + a_4$$

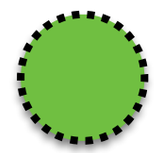
$$Pr(YES) = |a_1 + a_2 + a_3 + a_4|^2$$

$$Pr(YES) = \left| \sum_{path \in YES-Paths} a_{path} \right|^2$$

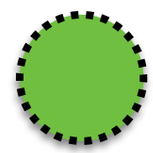
$$Pr(NO) = \left| \sum_{path \in NO-Paths} a_{path} \right|^2$$

$$Pr(YES) + Pr(NO) = 1$$

Quantum computation



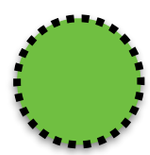
a_1



a_2

$$Amp(YES) = a_1 + a_2 + a_3 + a_4$$

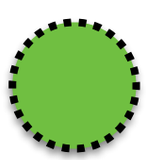
$$Pr(YES) = |a_1 + a_2 + a_3 + a_4|^2$$



a_3

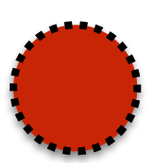
$$Pr(YES) = \left| \sum_{path \in YES-Paths} a_{path} \right|^2$$

$$Pr(NO) = \left| \sum_{path \in NO-Paths} a_{path} \right|^2$$



a_4

$$Pr(YES) + Pr(NO) = 1$$



a_5

When answer is YES, Pr(YES) > 2/3

When answer is NO, Pr(YES) < 1/3

Quantum computation

For simplicity, assume all amplitudes are real numbers

Quantum computation

For simplicity, assume all amplitudes are real numbers

$$\left(\sum_i a_i\right)^2 = \sum_i a_i^2 + \sum_{i \neq j} a_i a_j$$

Quantum computation

For simplicity, assume all amplitudes are real numbers

$$\left(\sum_i a_i\right)^2 = \sum_i a_i^2 + \boxed{\sum_{i \neq j} a_i a_j}$$

interference term

Quantum computation

For simplicity, assume all amplitudes are real numbers

$$\left(\sum_i a_i\right)^2 = \sum_i a_i^2 + \boxed{\sum_{i \neq j} a_i a_j}$$

interference term

Interference term close to 0 → classical computation

Quantum computation

For simplicity, assume all amplitudes are real numbers

$$\left(\sum_i a_i\right)^2 = \sum_i a_i^2 + \boxed{\sum_{i \neq j} a_i a_j}$$

interference term

Interference term close to 0 → classical computation

For everyday phenomena, amplitudes are fairly random

Quantum computation

For simplicity, assume all amplitudes are real numbers

$$\left(\sum_i a_i\right)^2 = \sum_i a_i^2 + \boxed{\sum_{i \neq j} a_i a_j}$$

interference term

Interference term close to 0 → classical computation

For everyday phenomena, amplitudes are fairly random

Interference term is close to 0

Quantum computation

For simplicity, assume all amplitudes are real numbers

$$\left(\sum_i a_i\right)^2 = \sum_i a_i^2 + \underbrace{\boxed{\sum_{i \neq j} a_i a_j}}_{\text{interference term}}$$

Interference term close to 0 → classical computation

For everyday phenomena, amplitudes are fairly random

Interference term is close to 0

This is why we don't see quantum weirdness around us!

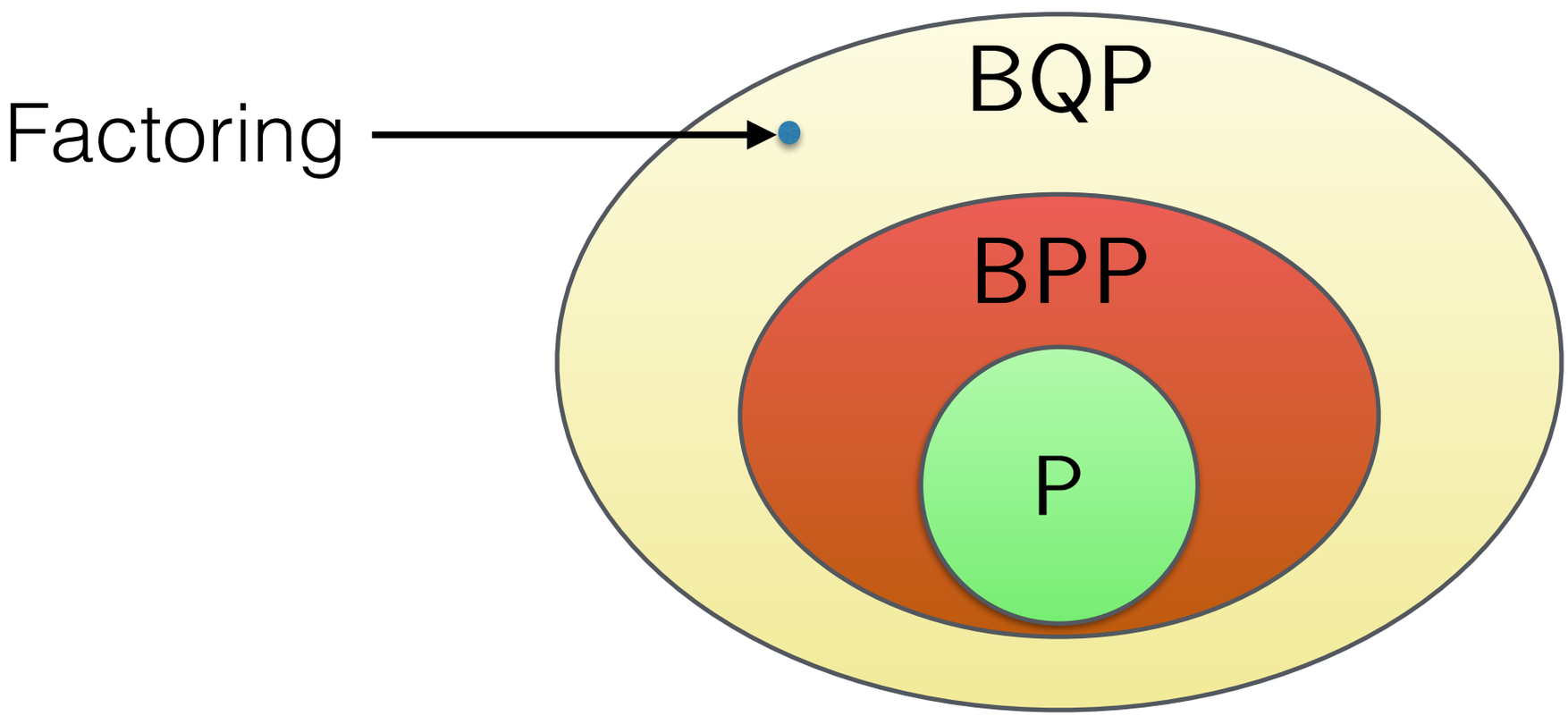
Classes of problems

Sets of problems that can be solved in polynomial time

Deterministic computation: **P**

Probabilistic computation: **BPP**

Quantum computation: **BQP**



Interference computation

Pick your programming language of choice

Imagine that in that language you had the function:

$$interf(f :: Set \to \{-1, +1\}, D :: Set) \to Bool$$

YES $$\frac{1}{|D|}\left|\sum_{x \in D} f(x)\right| \geq 2/3$$

NO $$\frac{1}{|D|}\left|\sum_{x \in D} f(x)\right| \leq 1/3$$

Otherwise, output is random

Interference computation

$$interf(f :: Set \to \{-1, +1\}, D :: Set) \to Bool$$

D can be specified **compactly**

E.g. $D = \{0, 1\}^n$

Would require exponential space to store D

f specified as pointer to some function

What can we solve with *interf*?

Deutsch-Josza problem

Suppose you can query the following function

$$f :: \{0,1\}^n \rightarrow \{0,1\}$$

The function is either **constant** or **balanced**

Constant: same value for all inputs

Balanced: 0 for half of inputs, 1 for other half

Constant functions:

$$\forall x \in \{0,1\}^n \; f(x) = 0 \qquad \forall x \in \{0,1\}^n \; f(x) = 1$$

Some balanced functions:

$$f(x) = parity(x)$$

$$f(x) = x_1$$

Deutsch-Josza problem

Suppose you can query the following function

$$f :: \{0,1\}^n \to \{0,1\}$$

The function is either **constant** or **balanced**

How many queries to determine type of function?

$$2^{n-1} + 1$$

Query function on half of domain + one extra element

What if we can use *interf*?

Deutsch-Josza problem

Let us define the following function:

$$g :: \{0,1\}^n \to \{-1,+1\} \quad g(x) = (-1)^{f(x)}$$

Call $interf(g, \{0,1\}^n)$

When f is constant

$$2^{-n} \left| \sum_{x \in \{0,1\}^n} g(x) \right| = 1 \to \mathbf{YES}$$

When f is balanced

$$2^{-n} \left| \sum_{x \in \{0,1\}^n} g(x) \right| = 0 \to \mathbf{NO}$$

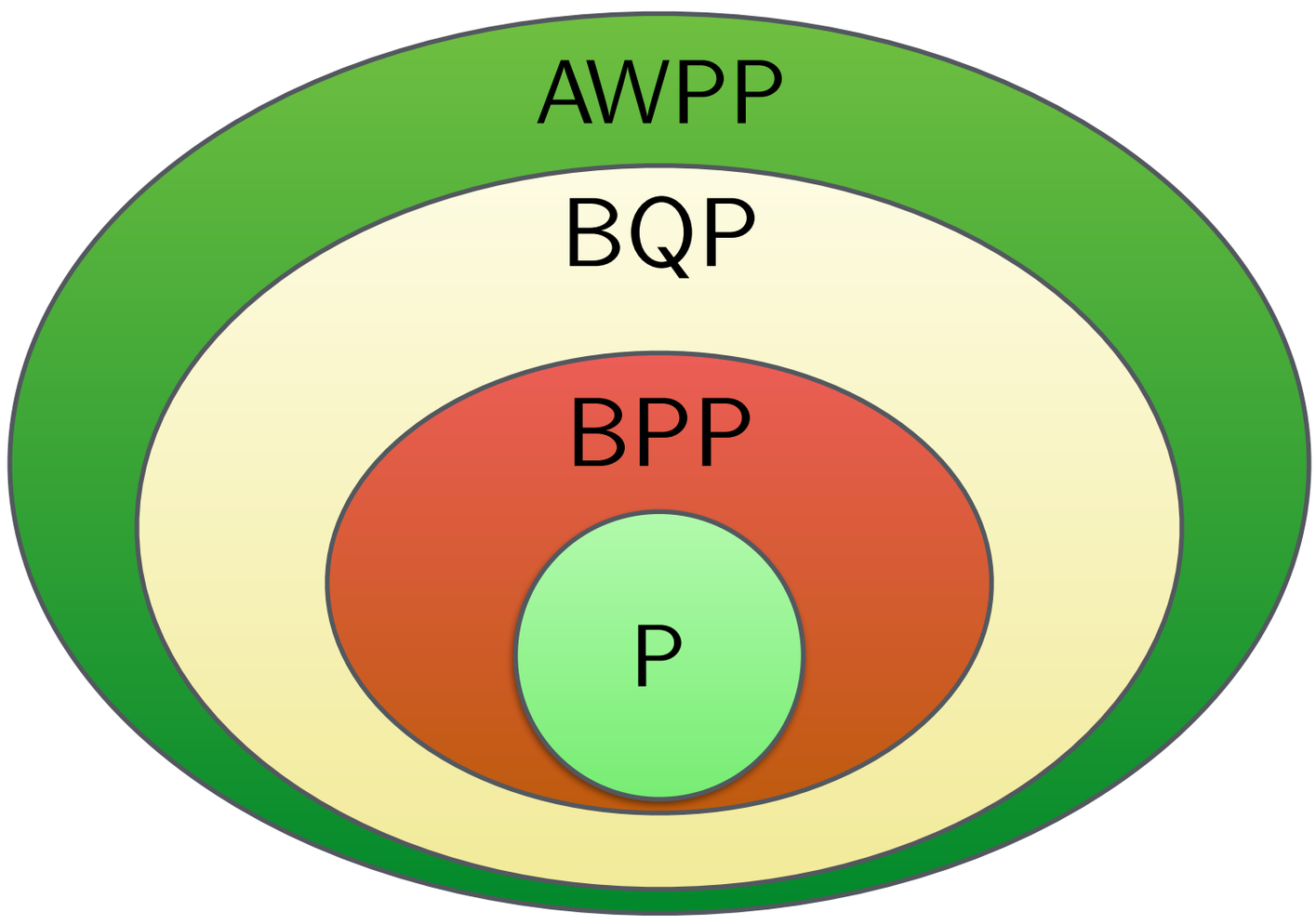
Interference computation

Quantum algorithms are mostly about implementing

$$interf(f :: Set \to \{-1, +1\}, D :: Set) \to Bool$$

Does classical computation + interf = quantum computation?

Probably not



The complexity zoo

QMA

coNP

NC

P/poly

NEXP

QCMA

ACC

L

PH

ONP

AM

BQP/qpoly

QIP

PostBPP

IP

MA

MIP

SZK

#P

PP

PostBQP

HVQSZK

QMA(2)

Favourite complexity class

YQP

Yoda Quantum Polynomial-time

https://arxiv.org/pdf/quant-ph/0608142.pdf

References and resources

Complexity Zoo
https://complexityzoo.uwaterloo.ca/Complexity_Zoo

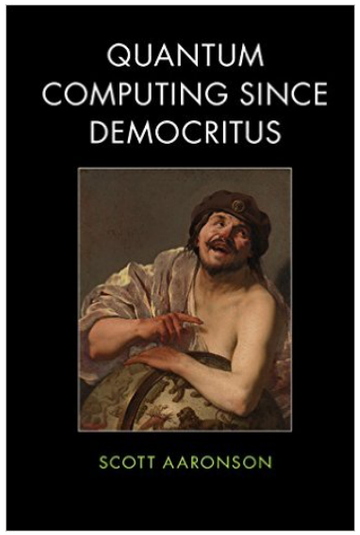
Complexity theory and quantum computation
http://stellar.mit.edu/S/course/6/fa14/6.845/materials.html

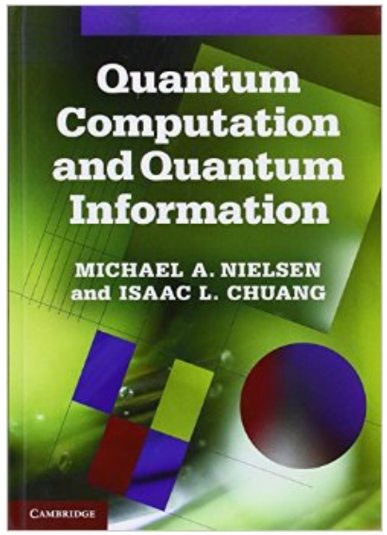
Complexity theory textbook
http://theory.cs.princeton.edu/complexity/book.pdf

Another great textbook
https://www.amazon.co.uk/Computational-Complexity-Christos-H-Papadimitriou/dp/0201530821

References and resources

For everything else…





References and resources

Images on slides 2,3,4,10

https://upload.wikimedia.org/wikipedia/commons/7/7e/Double-slit_experiment_results_Tanamura_2.jpg

https://upload.wikimedia.org/wikipedia/commons/thumb/8/86/Ebohr1_IP.svg/2000px-Ebohr1_IP.svg.png

https://upload.wikimedia.org/wikipedia/en/4/42/Richard_Feynman_Nobel.jpg

https://www.doorsteptutor.com/Submit-Question/GATE/Physics/posts/f2/f207e2d384a80dbd626ac8705abadba63206c18e84113ff0632d15cb293fd1de/Digital-circuit-of-three-different-gates.png

http://clipart.coolclips.com/480/vectors/tf05339/CoolClips_vc089665.png

References and resources

Images on slides 12,13,25,79

http://www.thesoftwarecomplex.com/wp-content/uploads/2015/05/Runcomputerfaster.jpg

http://www.hexblog.com/decompilation/pix/division.gif

https://vignette.wikia.nocookie.net/starwars/images/d/d6/Yoda_SWSB.png/revision/latest?cb=20150206140125

https://vecto.rs/600/vector-of-a-cartoon-happy-businesswoman-flipping-a-coin-coloring-page-outline-by-toonaday-20324.jpg