Programming Paradigms

Quantum Programming

Prof. Dr. Michael Pradel

Software Lab, University of Stuttgart

Summer 2022

Overview

- **Basics**
- **From Classical to Quantum Systems**
- **Describing Computations with Gates**
- **Algorithms**

Quantum Computing

- **Classical computers are reaching their limits, but demand for computation keeps increasing**
- **Quantum computers: New kind of hardware**
  - Builds on results in quantum physics
  - Computing power scales exponentially with number of qubits

Bits vs. Qubits

- **Classical computing**
  - Basic unit of information: Bit
  - Either $[1, 0]^T$ (i.e., zero) or $[0, 1]^T$ (i.e., one)
- **Quantum computing**
  - Basic unit of information: Qubit
  - Vector of two complex numbers $[c_0, c_1]^T$ where $|c_0|^2 + |c_1|^2 = 1$

Reminder: Complex Numbers

- **Complex number** $c = a + b \cdot i$ **consists of real part** a **and imaginary part** b
- **Imaginary number** $i = \sqrt{-1}$**,**
  i.e., $i^2 = -1$
- **Modulus of complex number:**
  $|c| = |a + b \cdot i| = \sqrt{a^2 + b^2}$

Quiz

Which of the following statements is correct?

- The modulus of $c = 1 - i$ is 0.5.
- The modulus of $c = 1 - i$ is 1.
- The modulus of $c = 1 - i$ is $\sqrt{2}$.
- $[\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}]^T$ is a qubit.
- $[\frac{1}{2} + i, -i]^T$ is a qubit.

Quiz

Which of the following statements is correct?

- ~~The modulus of $c = 1 - i$ is 0.5.~~
- ~~The modulus of $c = 1 - i$ is 1.~~
- The modulus of $c = 1 - i$ is $\sqrt{2}$.
- $[\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}]^T$ is a qubit.
- ~~$[\frac{1}{2} + i, -i]^T$ is a qubit.~~

Measuring a Qubit

$$[c_0, c_1]^T \xrightarrow{|c_0|^2} [1, 0]^T$$

$$[c_0, c_1]^T \xrightarrow{|c_1|^2} [0, 1]^T$$

- Qubit may be in two states at the same time
  → Superposition
- Measurement destroys the quantum state

$|c_0|^2$ ... probability that measurement produces zero

$|c_1|^2$ .. — " — one

Braket Notation

- **Different ways to denote qubits:**
  - $[c_0, c_1]^T$
  - $c_0 \cdot [1, 0]^T + c_1 \cdot [0, 1]^T$
  - $c_0 \cdot |0\rangle + c_1 \cdot |1\rangle$

Braket Notation

- **Different ways to denote qubits:**
  - $[c_0, c_1]^T$
  - $c_0 \cdot [1, 0]^T + c_1 \cdot [0, 1]^T$
  - $c_0 \cdot |0\rangle + c_1 \cdot |1\rangle$

Pronounced "ket zero" and "ket one"

Overview

- **Basics**
- **From Classical to Quantum Systems** ←
- **Describing Computations with Gates**
- **Algorithms**

From Classical to Quantum Systems

Same mathematical description for three kinds of systems (classical, probabilistic, quantum)

- State represented as vector X
- Dynamics represented as matrix M
- Compute next state: $X' = M \cdot X$
- Take multiple steps: $X^* = M^n \cdot X$

Example of Classical System

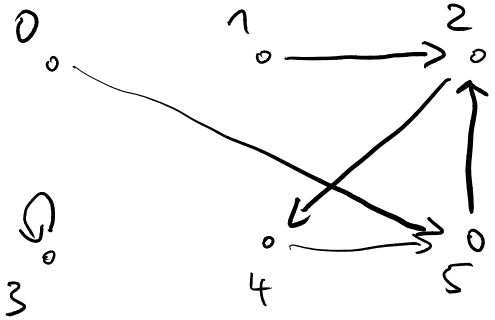
Intuition for state: Marbles placed on nodes in graph

State: 0 ∘ 6 1 ∘ 2 2 ∘ 1

3 ∘ 5 4 ∘ 3 5 ∘ 10

$$x = [6, 2, 1, 5, 3, 10]^T$$

Dynamics:



input

output

$$M = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$x' = M \cdot x = [0 \;\; 0 \;\; 12 \;\; 5 \;\; 1 \;\; 9]^T$$

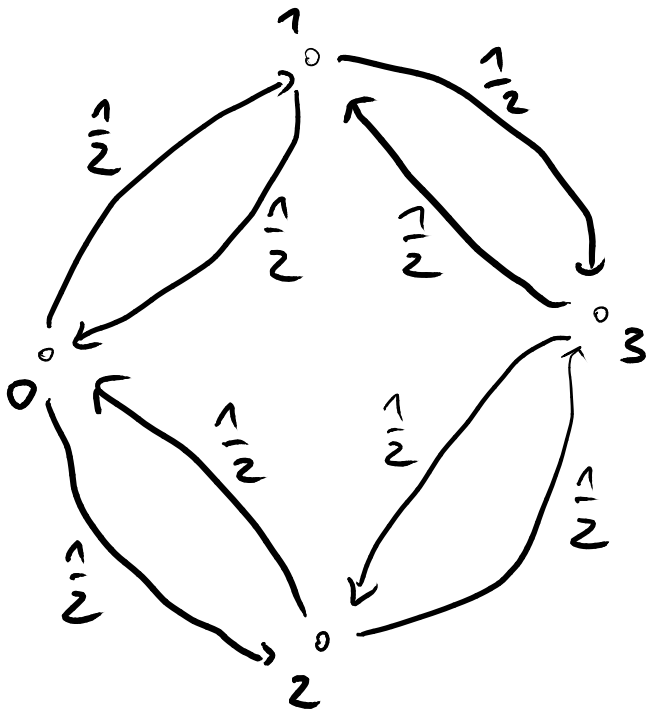
Probabilistic Systems

- **State vectors consist of probabilities**
  - Sum is one
- **Transition matrices consist of probabilities where**
  - sum of each row is one
  - sum of each column is one

  $\rightarrow$ "Doubly stochastic matrix"

Example: Stochastic Billiard Ball

M:



Initial state:

$$x = [1, 0, 0, 0]^T$$

One step:

$$x' = M \cdot x = [0, \tfrac{1}{2}, \tfrac{1}{2}, 0]^T$$

→ ball may be on two nodes

Two steps:

$$x'' = M \cdot x' = [\tfrac{1}{2}, 0, 0, \tfrac{1}{2}]^T$$

Three steps:

$$x''' = M \cdot x'' = x'$$

→ systems keeps flipping between 2 states

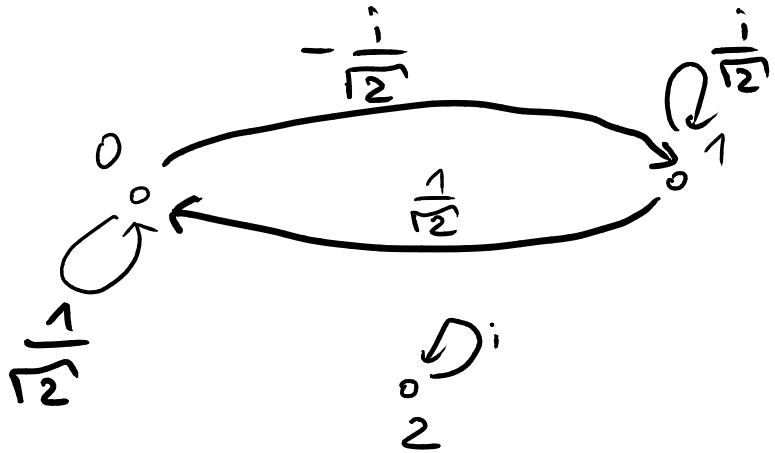
Quantum Systems

- **Entries in vectors and matrices: Complex numbers** c **with** $|c|^2 \in [0, 1]$
- **Unlike with probabilities,** $|c_1|^2 + |c_2|^2$ **may be smaller than** $|c_1 + c_2|^2$
  - Complex numbers may cancel each other out
  - Corresponds to interference in quantum physics

Example 1

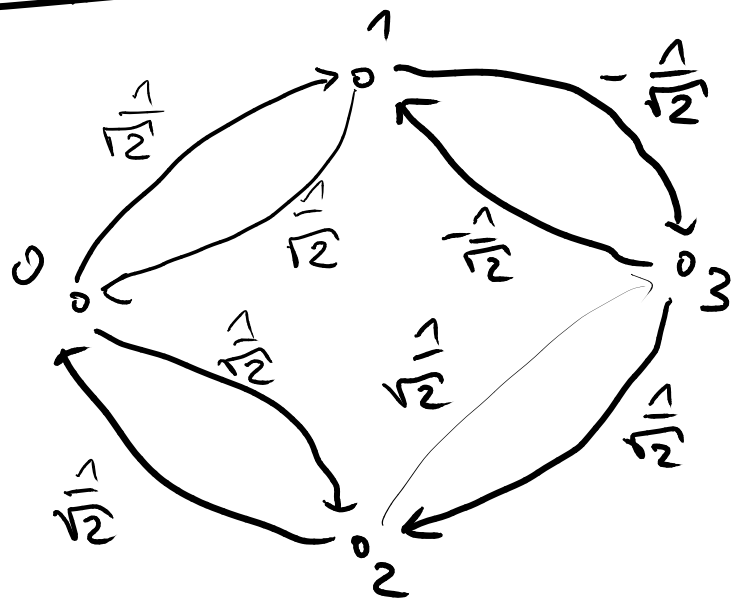
$$x : \left[\frac{1}{\sqrt{3}} , \frac{2 \cdot i}{\sqrt{15}} , \sqrt{\frac{2}{5}} \right]^T$$

M:



Example 2: Quantum Billiard Ball

M:



or as a matrix:

$$M = \begin{bmatrix} 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & 0 & 0 & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & 0 & 0 & \frac{1}{\sqrt{2}} \\ 0 & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \end{bmatrix}$$

Initial state:

$$x = [1, 0, 0, 0]^T$$

One step:

$$x' = M \cdot x = \left[0, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0\right]^T$$

Two steps:

$$x'' = M \cdot x' = [1, 0, 0, 0]^T$$

→ Ball goes back to initial state!

(because some paths cancel each other out → interference)

Unitary Matrices

- **Matrix U is unitary if its conjugate transpose $U^\dagger$ is also its inverse:**
  $U \cdot U^\dagger = I$
  - "Conjugate transpose" means to
    - compute the conjugate of each entry:
      $\overline{c} = \overline{a + b \cdot i} = a - b \cdot i$
    - transpose the matrix
- **Computation steps described by a unitary matrix can be reversed**

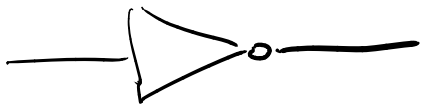
Overview

- **Basics**
- **From Classical to Quantum Systems**
- **Describing Computations with Gates** ←
- **Algorithms**

Describing Computation with Gates

- **Classical logic gate**
  - Manipulates one or more bits
- **Quantum gate**
  - Manipulates one or more qubits
  - Each gate corresponds to a unitary matrix

Classical Example: NOT



or as a matrix: $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

Using the gate:

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Quantum Examples

1) Pauli X-Gate:

Quantum equivalent of logical NOT

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

2) Hadamard Gate:

$$H = \frac{1}{\sqrt{2}} \cdot \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$$

Using H on $|0\rangle$: $H \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$ → Creates equal superposition of states

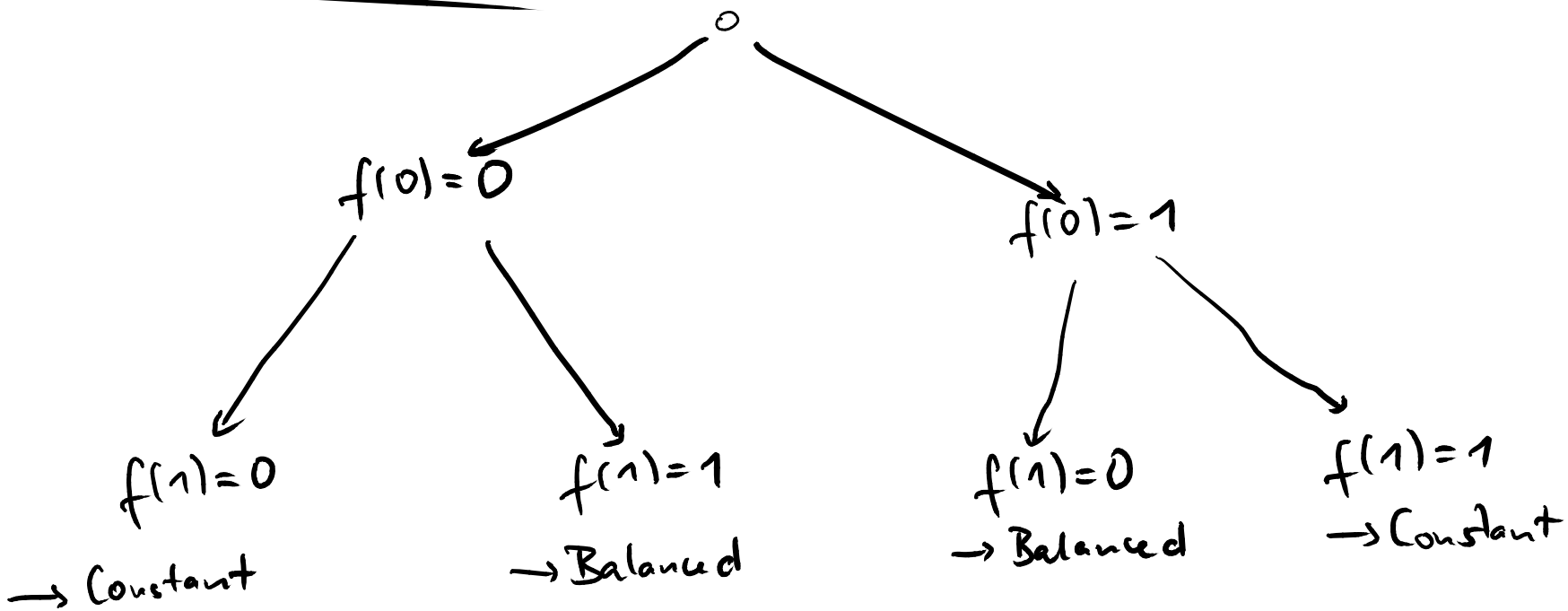
Overview

- **Basics**
- **From Classical to Quantum Systems**
- **Describing Computations with Gates**
- **Algorithms** ←

Deutsch's Algorithm

- **Named after British physicist David Deutsch**
- **Given**
  - Blackbox function $f : \{0, 1\} \rightarrow \{0, 1\}$
  - f is either "balanced" ($f(0) \neq f(1)$) or "constant" ($f(0) = f(1)$)
- **Goal: Find out whether f is balanced or constant**

Classical Algorithm



→ Must call f twice

Quantum Algorithm

- **Use superposition of two basic states to evaluate both inputs to f at once**
- **Quantum circuit that**
  - takes a blackbox function f as input,
  - "calls" f once,
  - yields a measured bit indicating whether f is balanced or constant

Function f as a Matrix

f is <u>one</u> of <u>four</u> possible matrices

→ Balanced 1: $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

→ Balanced 2: $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

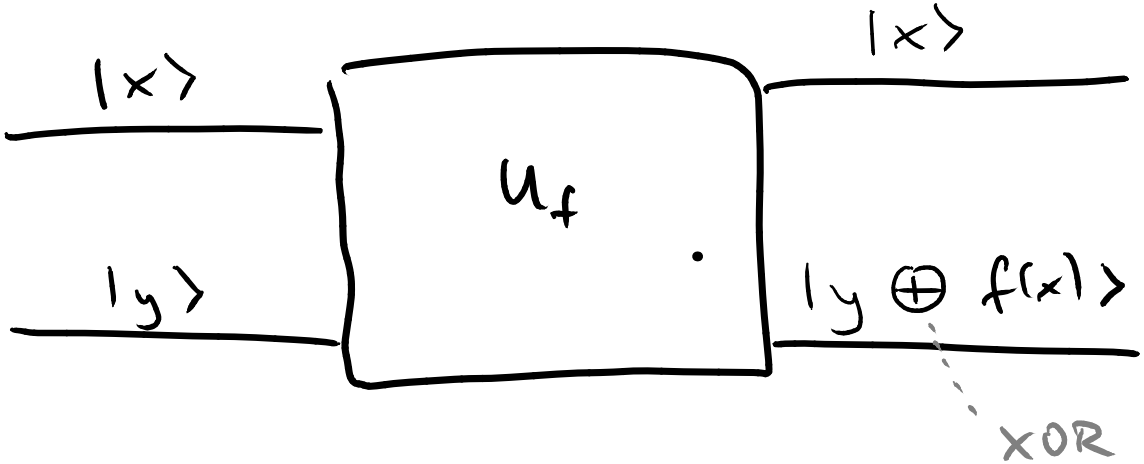
→ Constant 1: $\begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}$

→ Constant 2: $\begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}$

Not unitary

→ Can't use in quantum algor.!

Trick: "Wrap" f into a <u>unitary helper gate</u> U_f

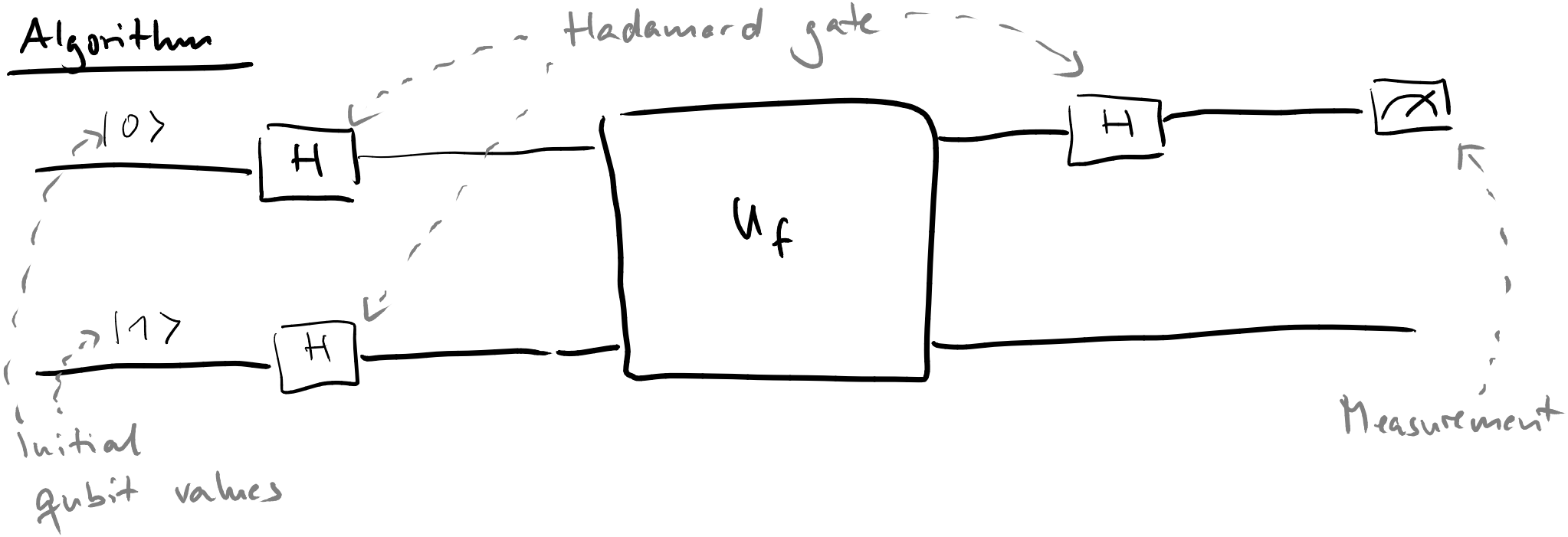


Example: If f is "Balanced 1"

then U_f is:

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Algorithm



or the same written as matrices:

$$(H \otimes I) \cdot U_f \cdot (H \otimes H) \cdot |0,1\rangle$$

identity matrix

tensor product

Qiskit Implementation

Demo

- Walk through main algorithm
- Quantum versions of four possible fs
- Histogram for each of the four fs
- Comment out last H gate and show histogram

Generalization

- **Deutsch's algorithm**
  - Input is $f : \{0, 1\} \to \{0, 1\}$
- **Deutsch-Jozsa algorithm**
  - Input is $f : \{0, 1\}^n \to \{0, 1\}$
  - I.e., string of n zeros and ones
  - "Balanced": Half of all inputs map to 0
  - "Constant": All inputs map to same output

Classical vs. Quantum

Must invoke f on more than half of its inputs

$$\frac{2^n}{2} + 1 = 2^{n-1} - 1$$

Invokes f only <u>once</u>

Exponential speedup

Other Algorithms

- **Shor's algorithm**
  - Find prime factors of an integer
- **Grover's search algorithm**
  - Find the unique input to f that produces a specific output

Other Algorithms

- **Shor's algorithm**
  - Find prime factors of an integer
- **Grover's search algorithm**
  - Find the unique input to f that produces a specific output

Commonality: Lower complexity than classical algorithms

Future of Quantum Computing

- **Still a very young field**
  - 11 quantum computers in 2018
  - Several thousands estimated for 2030
  - E.g., IBM Eagle has 127 qubits (released 11/2021)
- **Lots of opportunities**
  - Quantum programming languages
  - Applications of quantum computing

Overview

- **Basics**
- **From Classical to Quantum Systems**
- **Describing Computations with Gates**
- **Algorithms** ✔