6. Quantum computational complexity
(mostly reading)

NC 3.2 (less detailed), KLM 9.1 (more detailed)

Professor John Watrous Lecture notes
Quantum Computation spring 2006, lecture 22
https://johnwatrous.com/lecture-notes/
scroll to the very bottom!

Complexity Zoo by Scott Aaronson

Big-O notation (reading):

Choose a parameter n. Let $f, g : \mathbb{N} \to \mathbb{R}$.

$g(n) \in O(f(n))$ iff $\exists n_0, c\ \forall n > n_0\quad g(n) \leq c f(n)$

$g(n) \in \Omega(f(n))$ iff $\exists n_0, c\ \forall n > n_0\quad g(n) \geq c f(n)$

iff $f(n) \in O(g(n))$

$g(n) \in \Theta(f(n))$ iff $g(n) \in O(f(n)) \wedge g(n) \in \Omega(f(n))$

$g(n) \in o(f(n))$ iff $\lim\limits_{n \to \infty} \dfrac{g(n)}{f(n)} = 0$

$g(n) \in \mathrm{poly}(n)$ iff $g(n) \in O(n^c)$ for some fixed c.

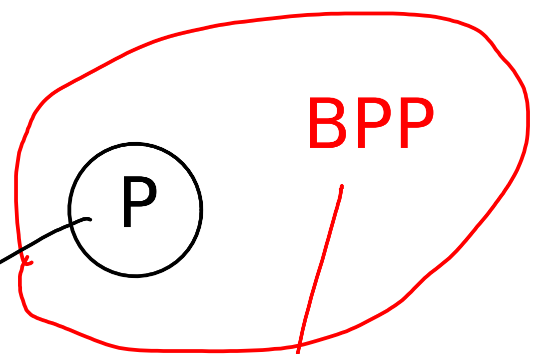
eg $5n^3 + 2n - 8 \in O(n^3),\ O(n^4),\ \Omega(n^3),\ \Omega(n^2)$

$C = 6 \quad C = 1 \quad C = 4 \quad C = 1$

$\in \Theta(n^3),\ \mathrm{poly}(n)$

Polynomial time classical computation --
Problems whose complexity increases "slowly enough" in the input size, and what's considered "feasible".

Suspected: P=BPP
ie randomness doesn't help



poly-time

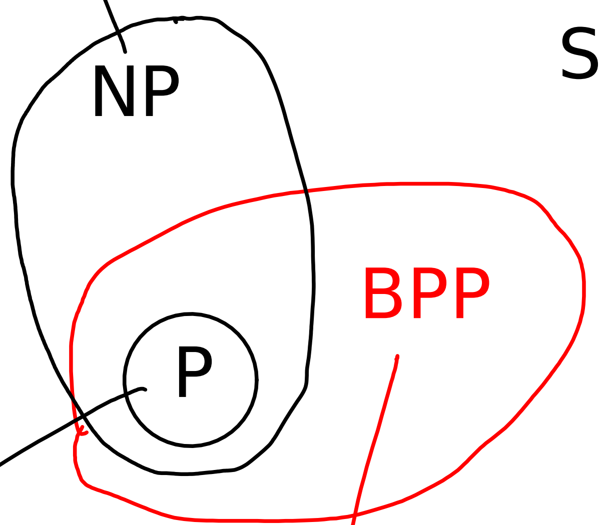
probabilistic poly time

Nondeterministic polynomial time problems: those, given the answer, can be verified in polynomial time

verifiable in poly-time

May not know how to solve in poly-time, e.g., 3-SAT

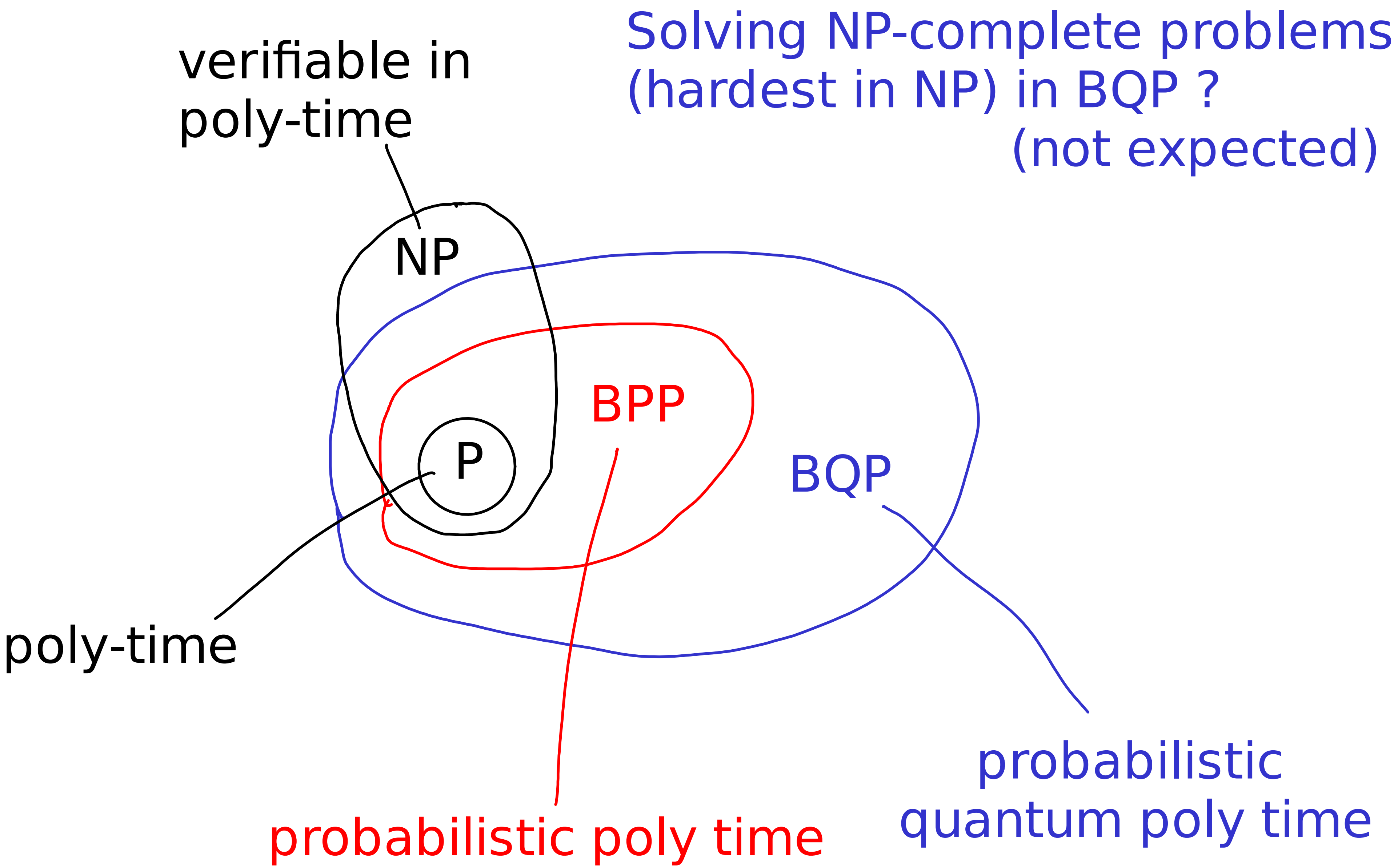
Suspected: P $\neq$ NP



poly-time

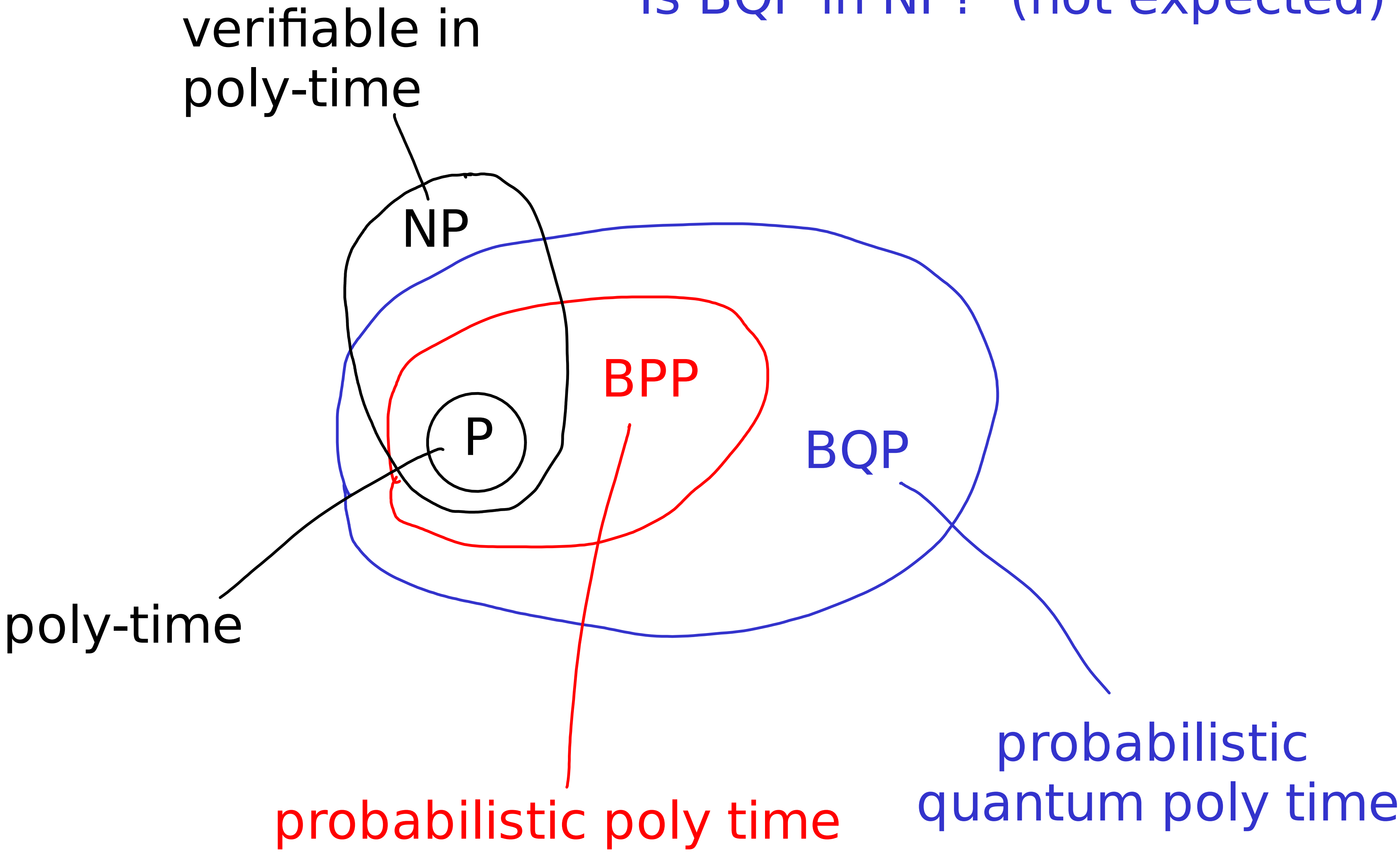
probabilistic poly time

Is a quantum computer much more powerful than
a classical computer?



Is a quantum computer much more powerful than a classical computer?

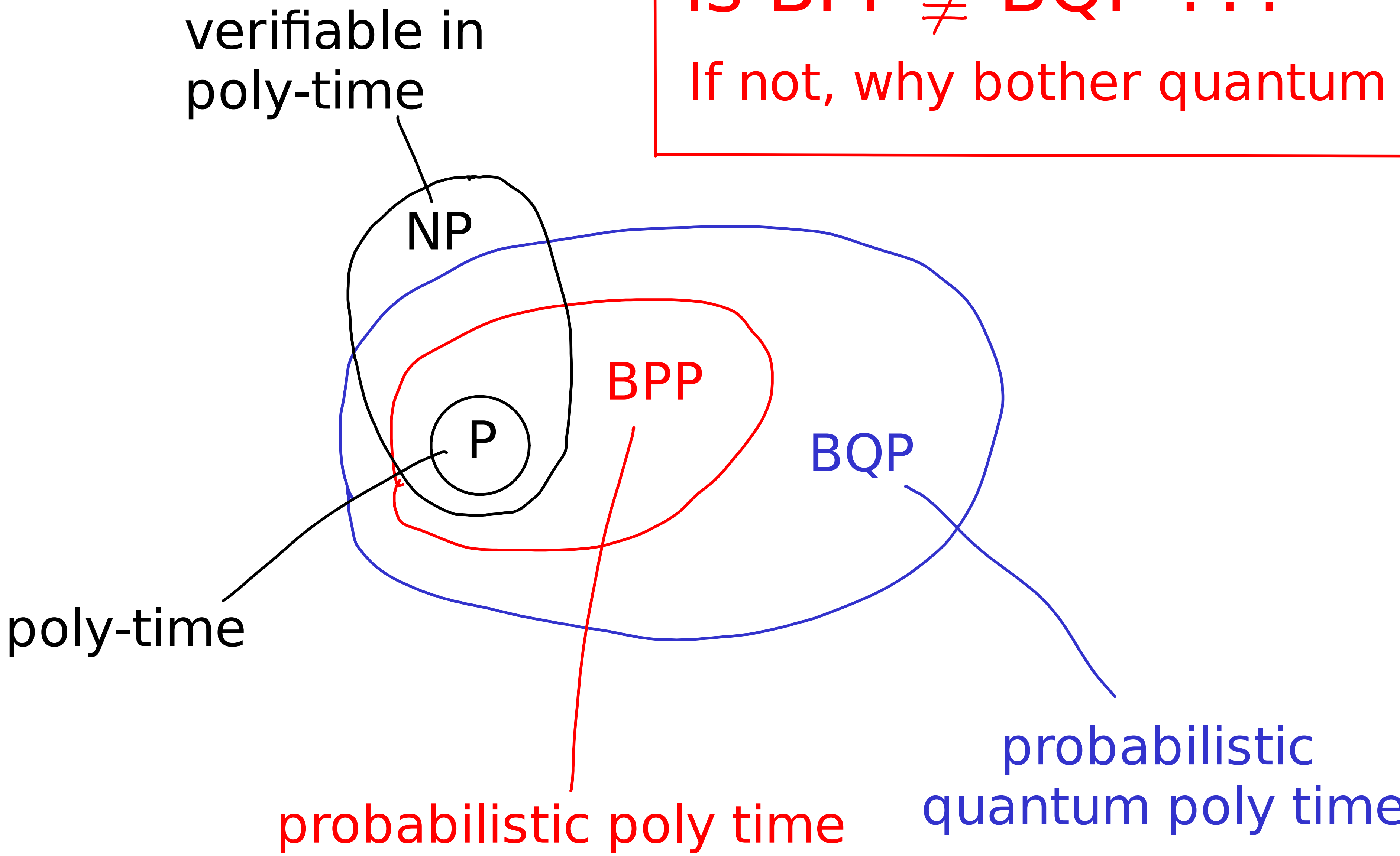
Is BQP in NP? (not expected)



Is a quantum computer much more powerful than a classical computer?

Is BPP $\subsetneq$ BQP ???

If not, why bother quantum ?

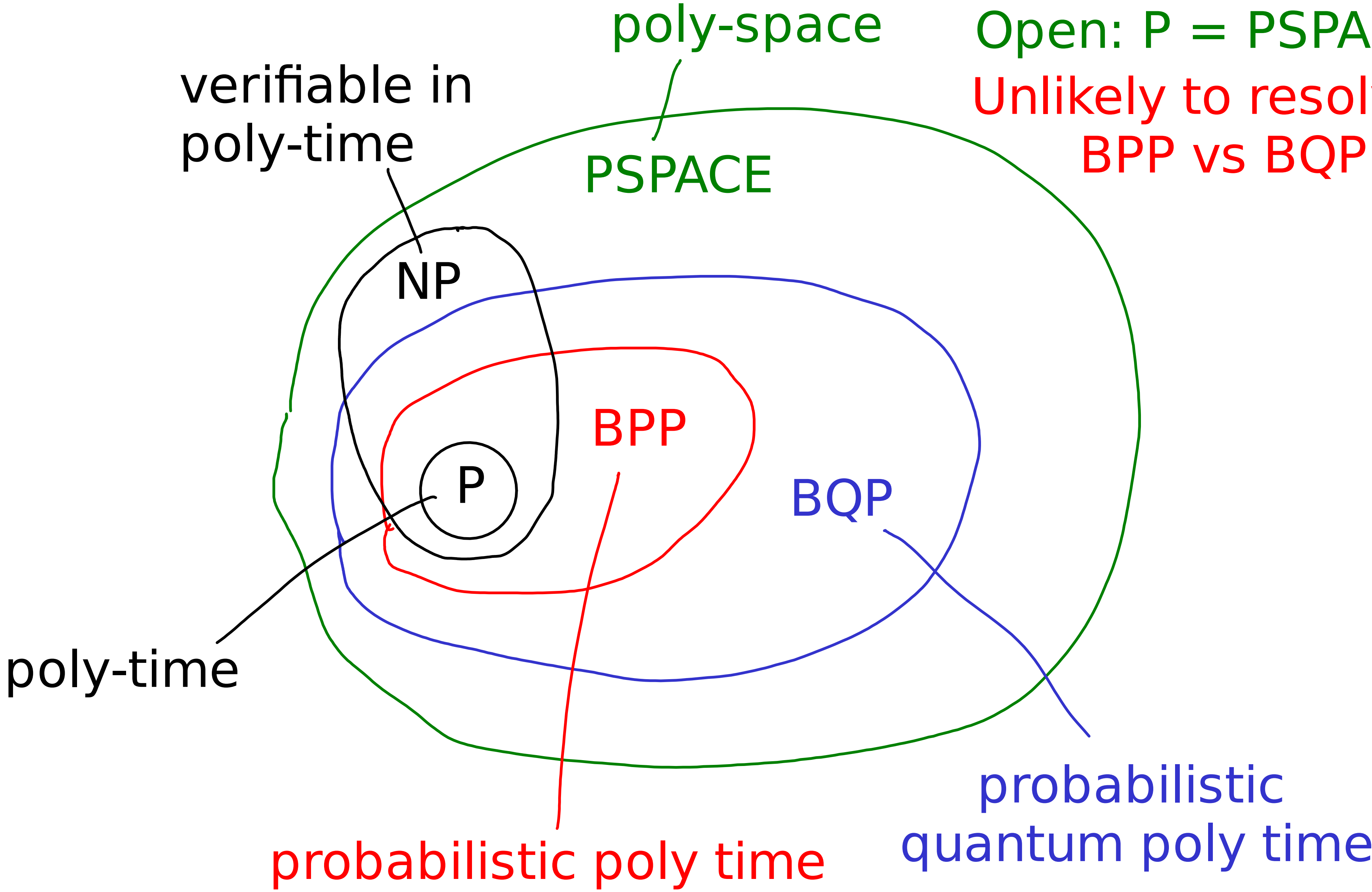


Is a quantum computer much more powerful than a classical computer?

$\mathrm{BQP} \subseteq \mathrm{PSPACE}$

Open: P = PSPACE

Unlikely to resolve BPP vs BQP ...



Surprisingly hard to show a problem is hard ...

The fact we can't find an efficient algorithm doesn't imply there is none ...

Idea: we turn to a different measure of complexity (not the circuit size).

7. Quantum algorithms (part 1)

(a) Quantum query complexity: (KLM 9.2*, 6.2*)
black box model, phase kick back

(d) Deutsch-Jozsa algorithm
(NC 1.4.2-1.4.5, KLM 6.3-6.4, M 2.2)

(e) Quantum fourier transform (I)
(NC 5.1, M 3.5, KLM p110-117)

(f) Simon's algorithm (M 2.5, KLM 6.5)

(g) Shor's factoring algorithm

(M 3.1-3.4,3.7-3.10, NC 5.3, 5.4.1-5.4.2,
KLM 7.1.2-7.1.3, 7.3.1-7.3.2, 7.3.4, 7.4)

(h) Hidden subgroup framework (NC 5.4.3, KLM 7.5)

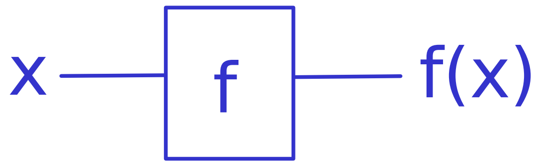
Black box model

Let f be a partially unknown function.
Goal: determine some properties of f.

Black box model

Let f be a partially unknown function.
Goal: determine some properties of f.

Allowed: "query" a blackbox for f :



If input is x (in domain of f), the blackbox outputs f(x).

Not allowed: open a blackbox and see what's inside.

Black box model

e.g. $f(x) = a\,x^2 + b\,x + c$, a polynomial over a field F.

Known: f polynomial of degree 2
Unknown: a,b,c.
(1) How many queries are needed to learn c?
(2) What about b?

Goal:
- Solve problem with few queries
- Check if solution is optimal

Black box model

e.g. $f(x) = a\,x^2 + b\,x + c$

(1) 1 query is necessary and sufficient to learn c:

$$0 \;—\; \boxed{f} \;—\; f(0)=c$$

NB inputs to queries should be optimized!

Black box model

e.g. $f(x) = a\,x^2 + b\,x + c$

(2) 2 queries are necessary and sufficient to learn b:

Black box model

e.g. $f(x) = a x^2 + b x + c$

(2) 2 queries are necessary and sufficient to learn b:

Sufficiency:

i.e., an algorithm to solve the problem,
giving an upper bound on the required # of queries

Black box model

e.g. $f(x) = a\,x^2 + b\,x + c$

(2) 2 queries are necessary and sufficient to learn b:

Sufficiency:

Pick $q \neq 0$, query q and $-q$; $b = (\underbrace{f(q) - f(-q)}_{\text{from the black box}})/2q$.

Black box model

e.g. $f(x) = a x^2 + b x + c$

(2) 2 queries are necessary and sufficient to learn b:

Necessity:

ie., proving lower-bound on the required # of queries
 -- useful for checking optimality of known solutions

Black box model

e.g. $f(x) = a\,x^2 + b\,x + c$

(2) 2 queries are necessary and sufficient to learn b:

Necessity:
Suppose, by contradiction, 1 query suffices.

Black box model

e.g. $f(x) = a x^2 + b x + c$

(2) 2 queries are necessary and sufficient to learn b:

Necessity:
Suppose, by contradiction, 1 query suffices.
Let q be the input for that query.

Black box model

e.g. $f(x) = a\,x^2 + b\,x + c$

(2) 2 queries are necessary and sufficient to learn b:

Necessity:
Suppose, by contradiction, 1 query suffices.
Let q be the input for that query.

Define another poly^{n}: $g(x) = a\,x^2 + (b+1)\,x + (c-q)$

Black box model

e.g. $f(x) = a\,x^2 + b\,x + c$

(2) 2 queries are necessary and sufficient to learn b:

Necessity:
Suppose, by contradiction, 1 query suffices.
Let q be the input for that query.

Define another poly^{n}: $g(x) = a\,x^2 + (b+1)\,x + (c-q)$

Now, $g(q) = a\,q^2 + (b+1)\,q + (c-q)$
$= a\,q^2 + b\,q + c = f(q)$.

Black box model

e.g. $f(x) = a\,x^2 + b\,x + c$

(2) 2 queries are necessary and sufficient to learn b:

Necessity:
Suppose, by contradiction, 1 query suffices.
Let q be the input for that query.

Define another poly^n: $g(x) = a\,x^2 + (b+1)\,x + (c\text{-}q)$

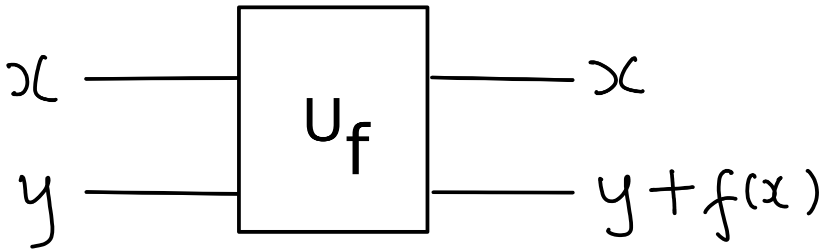
$$\begin{aligned} \text{Now, } g(q) &= a\,q^2 + (b+1)\,q + (c\text{-}q) \\ &= a\,q^2 + b\,q + c = f(q)\,. \end{aligned}$$

So, query q cannot distinguish f(x) from g(x) but they have different linear coefficients, a contradiction.

Student feedback from W2019:

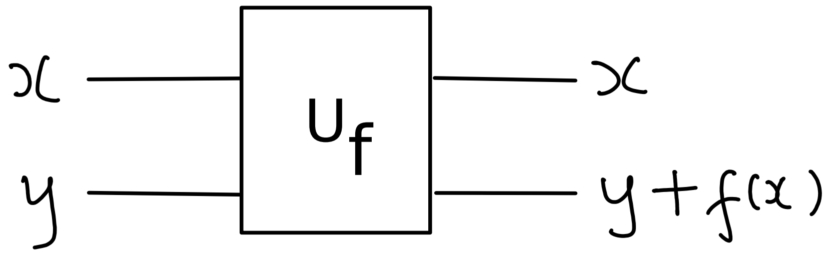
If the coefficients are real, but the input/output are allowed to be complex, then 1 query suffices. Note that this scenario breaks the condition that the polynomial is over a field (both inputs and coeffs are from the same field). In this scenario, $g(x)$ in the proof is not a valid polynomial (why there is no contradiction to our proven result).

Reversible blackbox:



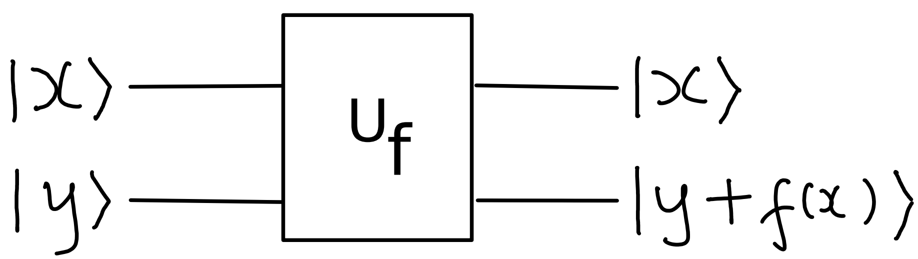
(mod 2 if bottom register is a bit)

Reversible blackbox:



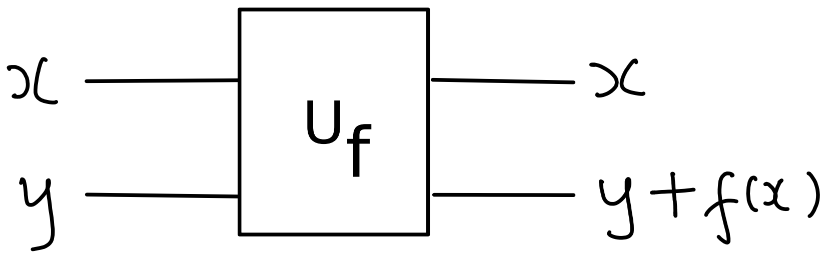
(mod 2 if bottom register is a bit)

Quantum blackbox:



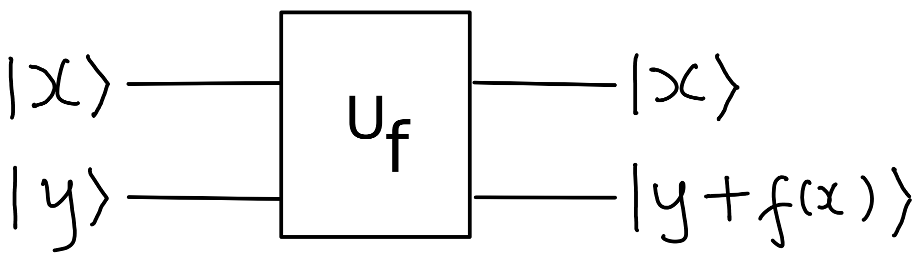
(mod 2 if bottom register is a bit)

Reversible blackbox:



(mod 2 if bottom register is a bit)

Quantum blackbox:

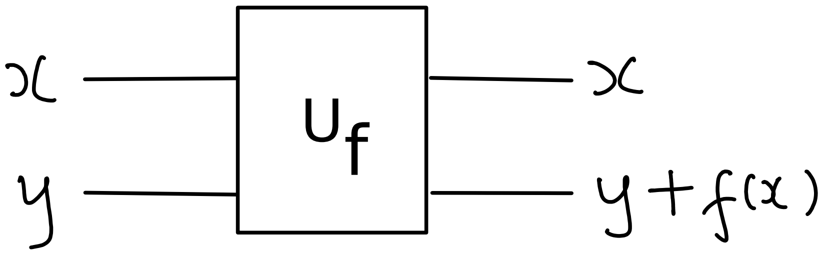


(mod 2 if bottom register is a bit)

It's like an "f(x)-controlled-NOT".

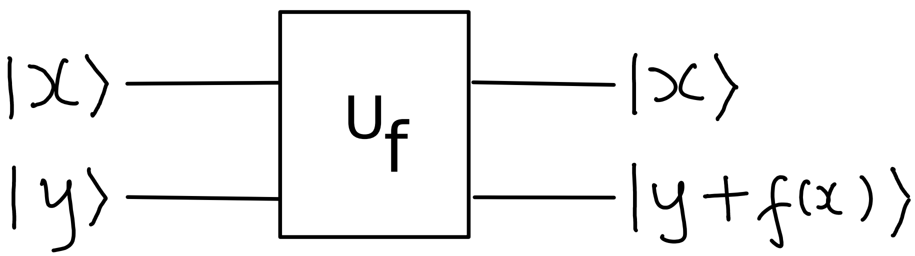
1. compute f(x) keeping x
2. CNOT from f(x) to target
3. uncompute f(x)

Reversible blackbox:



(mod 2 if bottom register is a bit)

Quantum blackbox:



(mod 2 if bottom register is a bit)

Qn: is quantum computation with quantum black boxes more powerful than classical computation with reversible classical blackboxes?

Quantum programming technique 1

Phase kick back

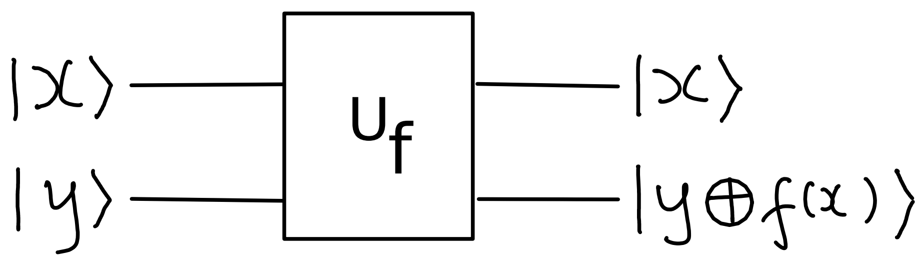
Phase kick back

For a Boolean function f (the range is {0,1}), the quantum blackbox of f can be modified to "answer in the phase".

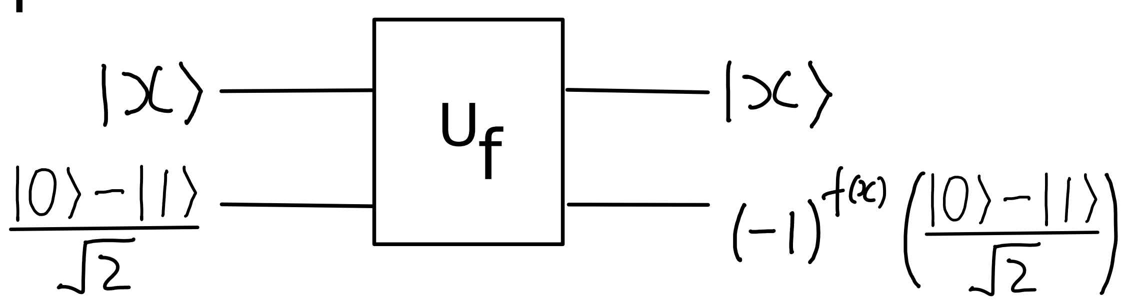
Phase kick back

For a Boolean function f (the range is {0,1}), the quantum blackbox of f can be modified to "answer in the phase".

i.e., If



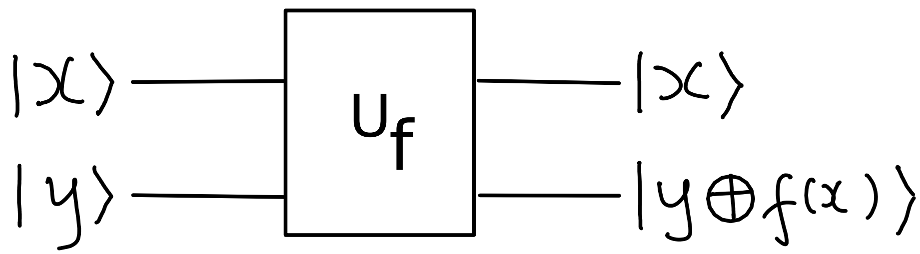
then



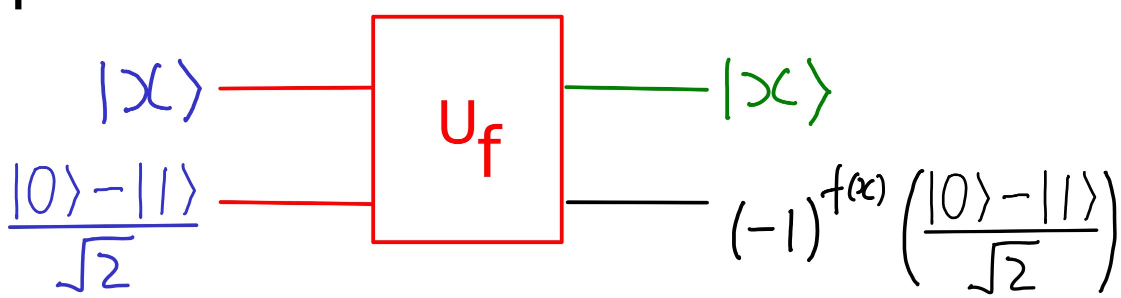
Phase kick back

For a Boolean function f (the range is {0,1}), the quantum blackbox of f can be modified to "answer in the phase".

i.e., If



then



Proof:

$$|x\rangle\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) \xrightarrow{U_f} \frac{1}{\sqrt{2}}\Big(|x\rangle|f(x)\rangle - |x\rangle|f(x)\oplus 1\rangle\Big)$$

$$|x\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right) \xrightarrow{U_f} \frac{1}{\sqrt{2}}\left(|x\rangle|f(x)\rangle - |x\rangle|f(x)\oplus 1\rangle\right)$$

$$|x\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right) \xrightarrow{U_f} \frac{1}{\sqrt{2}}\Big(|x\rangle|f(x)\rangle - |x\rangle|f(x)\oplus 1\rangle\Big)$$

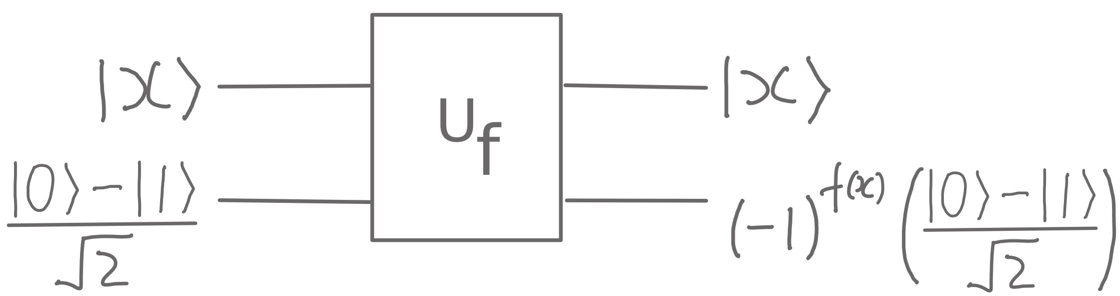
$$= \frac{1}{\sqrt{2}}|x\rangle \otimes \begin{cases} |0\rangle-|1\rangle & \text{if } f(x) = 0 \\ |1\rangle-|0\rangle & \text{if } f(x) = 1 \end{cases}$$

$$|x\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right) \xrightarrow{U_f} \frac{1}{\sqrt{2}}\Big(|x\rangle|f(x)\rangle - |x\rangle|f(x)\oplus 1\rangle\Big)$$

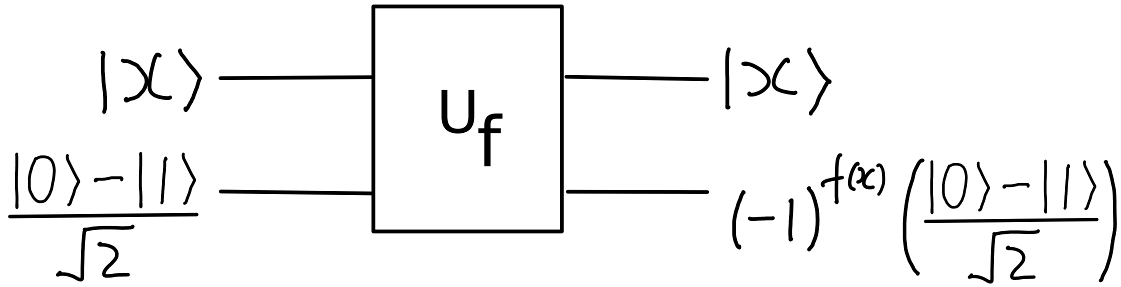
$$= \frac{1}{\sqrt{2}}\,|x\rangle \otimes \begin{cases} |0\rangle-|1\rangle & \text{if } f(x)=0 \\ |1\rangle-|0\rangle & \text{if } f(x)=1 \end{cases}$$

$$= |x\rangle \otimes \frac{1}{\sqrt{2}}\big(|0\rangle-|1\rangle\big)(-1)^{f(x)} \qquad \square$$

which is what we seek to prove:

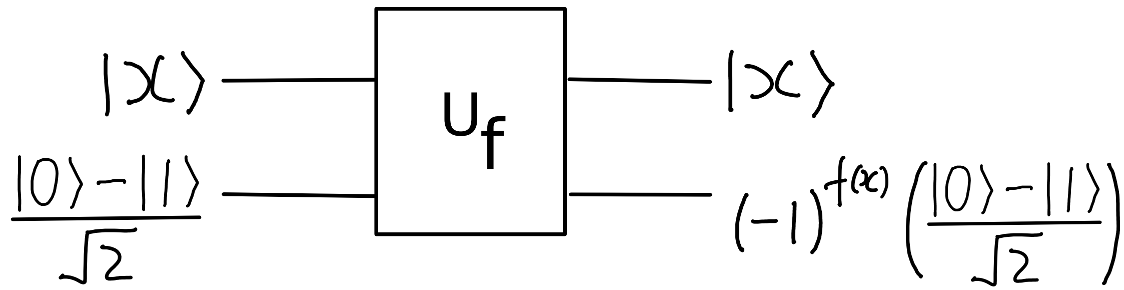


Phase kick back



For one x, the black box kicks back an overall phase,
for a superposition of inputs, the phase is relative !

Phase kick back



For one x, the black box kicks back an overall phase, for a superposition of inputs, the phase is relative !

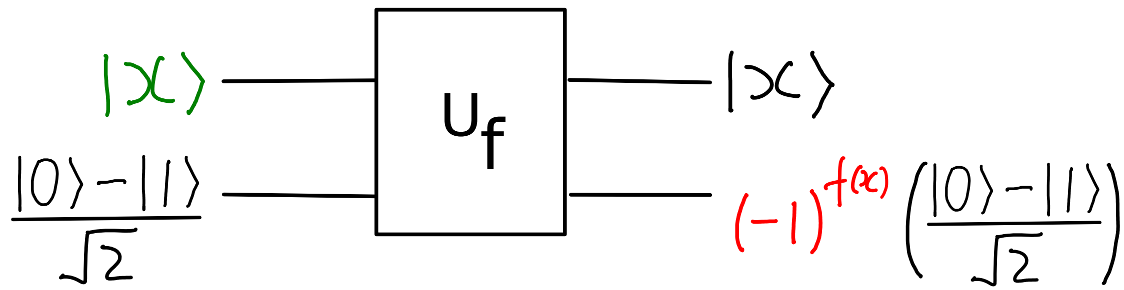
e.g., f(x) = x, x = 0,1

Input $\frac{1}{\sqrt{2}}\left(|0\rangle+|1\rangle\right)$ (superpose x=0 & x=1).

$$\frac{1}{\sqrt{2}}\left(|0\rangle+|1\rangle\right)\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right) \xrightarrow{U_f} \frac{1}{\sqrt{2}}\left(+|0\rangle-|1\rangle\right)\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)$$

So, kick back is NOT an overall phase for U_f.

Phase kick back



In fact, the black box is like a controlled-gate:

$(-1)^{f(x)}$ I is applied to the target if the control is $|x\rangle$.

with target input is fixed to $\frac{|0\rangle - |1\rangle}{\sqrt{2}}$.

Recall: control qubit can change.
Goal: use this change to compute !

7. Quantum algorithms (part 1)

(a) Quantum query complexity: (KLM 9.2*, 6.2*)
black box model, phase kick back

(d) Deutsch-Jozsa algorithm
(NC 1.4.2-1.4.5, KLM 6.3-6.4, M 2.2)

(e) Quantum fourier transform (I)
(NC 5.1, M 3.5, KLM p110-117)

(f) Simon's algorithm (M 2.5, KLM 6.5)

(g) Shor's factoring algorithm

(M 3.1-3.4,3.7-3.10, NC 5.3, 5.4.1-5.4.2,
KLM 7.1.2-7.1.3, 7.3.1-7.3.2, 7.3.4, 7.4)

(h) Hidden subgroup framework (NC 5.4.3, KLM 7.5)

Deutsch-Josza algorithm

Deutsch's problem:

Given: a black box for a function $f:\{0,1\} \to \{0,1\}$

Problem: Is f constant ($f(0)=f(1)$)
or balanced ($f(0)\neq f(1)$) ?

i.e., find $f(0) \oplus f(1)$

Deutsch's problem:

Given: a black box for a function $f:\{0,1\}\text{->}\{0,1\}$

Problem: Is f constant ($f(0)=f(1)$)
or balanced ($f(0)\neq f(1)$) ?

i.e., find $f(0) \oplus f(1)$

Classically, 2 queries are needed.

Ex: for each query x=0 or 1, for each possible answer,
you have exactly 1 constant & 1 balanced function
that are possible ...

Deutsch's problem:

Given: a black box for a function $f:\{0,1\} \to \{0,1\}$

Problem: Is f constant ($f(0)=f(1)$)
or balanced ($f(0) \neq f(1)$) ?

i.e., find $f(0) \oplus f(1)$

Classically, 2 queries are needed.

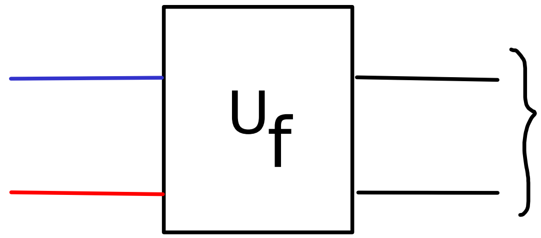
Quantumly, 1 query suffices!

What doesn't work:

query in
superposition

$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$

$|0\rangle$



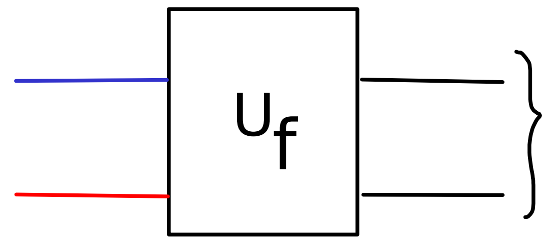
$$\frac{1}{\sqrt{2}}\left(|0\rangle|f(0)\rangle + |1\rangle|f(1)\rangle\right)$$

looks like we have both
f(0) & f(1) but measuring
2nd register gives one of
them at random ....

What doesn't work:

query in superposition $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$

$|0\rangle$



$\frac{1}{\sqrt{2}}(|0\rangle|f(0)\rangle + |1\rangle|f(1)\rangle)$

(IN)Distinguishability problem!

Possibility (1): f is constant, output is

$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|0\rangle$ or $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|1\rangle$

Possibility (2): f is balanced, output is

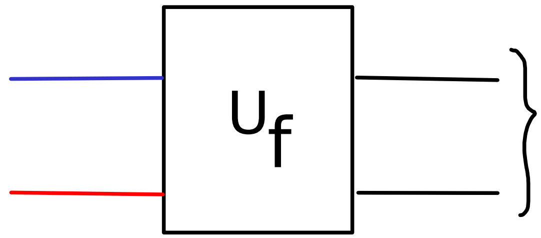
$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ or $\frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$

The two states within each possibility are mutually orthogonal, ...

What doesn't work:

query in superposition $\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)$

$|0\rangle$



$$\frac{1}{\sqrt{2}}\left(|0\rangle|f(0)\rangle+|1\rangle|f(1)\rangle\right)$$

(IN)Distinguishability problem!

Possibility (1): f is constant, output is

$$\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)|0\rangle \quad \text{or} \quad \frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)|1\rangle$$

Possibility (2): f is balanced, output is

$$\frac{1}{\sqrt{2}}(|00\rangle+|11\rangle) \quad \text{or} \quad \frac{1}{\sqrt{2}}(|01\rangle+|10\rangle)$$

The two states within each possibility are mutually orthogonal, BUT ... each state in possibility (1) is NOT orthogonal to each state in possibility (2).

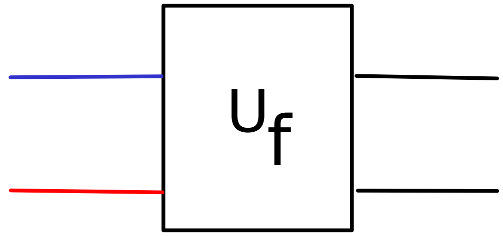
So, the two possibilities are NOT distinguishable !

What works:

query in superposition $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$

ancilla for phase kick back $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$

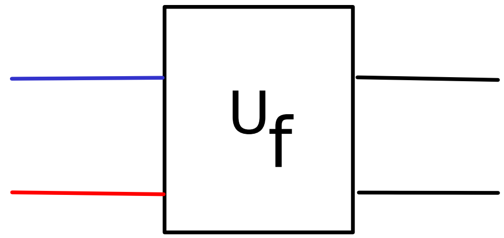
$$= |-\rangle$$



What works:

query in superposition $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$

ancilla for phase kick back $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |-\rangle$

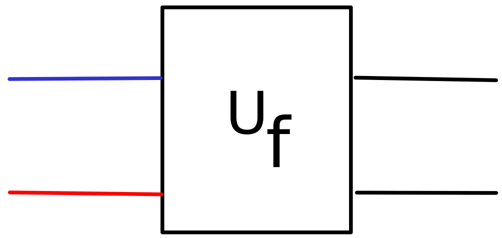


$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|-\rangle \xrightarrow{U_f} \left(\frac{1}{\sqrt{2}}|0\rangle(-1)^{f(0)} + \frac{1}{\sqrt{2}}|1\rangle(-1)^{f(1)}\right)|-\rangle$$

What works:

query in superposition $\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)$

ancilla for phase kick back $\frac{1}{\sqrt{2}}(|0\rangle-|1\rangle)$ $\parallel$ $|-\rangle$



$$\frac{1}{\sqrt{2}}\left(|0\rangle+|1\rangle\right)|-\rangle \xrightarrow{U_f} \left(\frac{1}{\sqrt{2}}|0\rangle(-1)^{f(0)}+\frac{1}{\sqrt{2}}|1\rangle(-1)^{f(1)}\right)|-\rangle$$

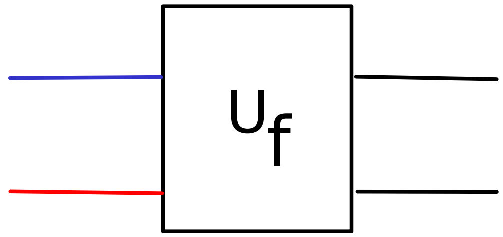
$$= (-1)^{f(0)}\frac{1}{\sqrt{2}}\left(|0\rangle+|1\rangle(-1)^{f(0)\oplus f(1)}\right)|-\rangle$$

overall phase relative phase

What works:

query in superposition $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$

ancilla for phase kick back $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |-\rangle$



$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|-\rangle \xrightarrow{U_f} \left(\frac{1}{\sqrt{2}}|0\rangle(-1)^{f(0)} + \frac{1}{\sqrt{2}}|1\rangle(-1)^{f(1)}\right)|-\rangle$$

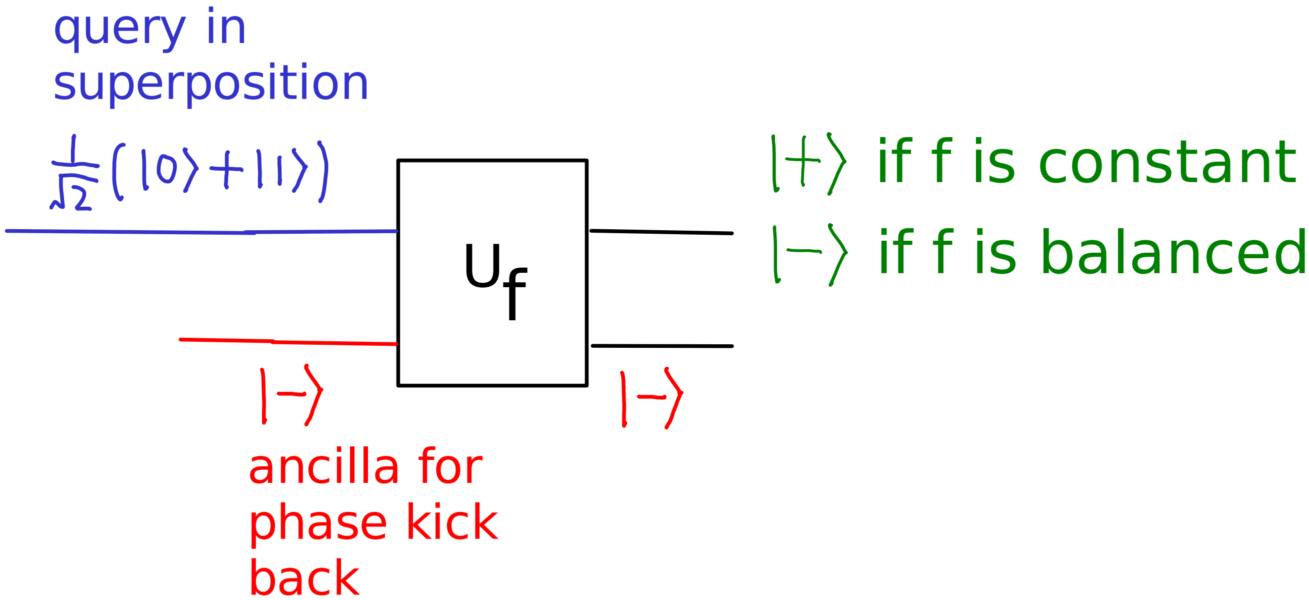
$$= \underset{\uparrow}{(-1)^{f(0)}} \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle \underset{\uparrow}{(-1)}^{f(0)\oplus f(1)}\right)|-\rangle$$

overall phase relative phase

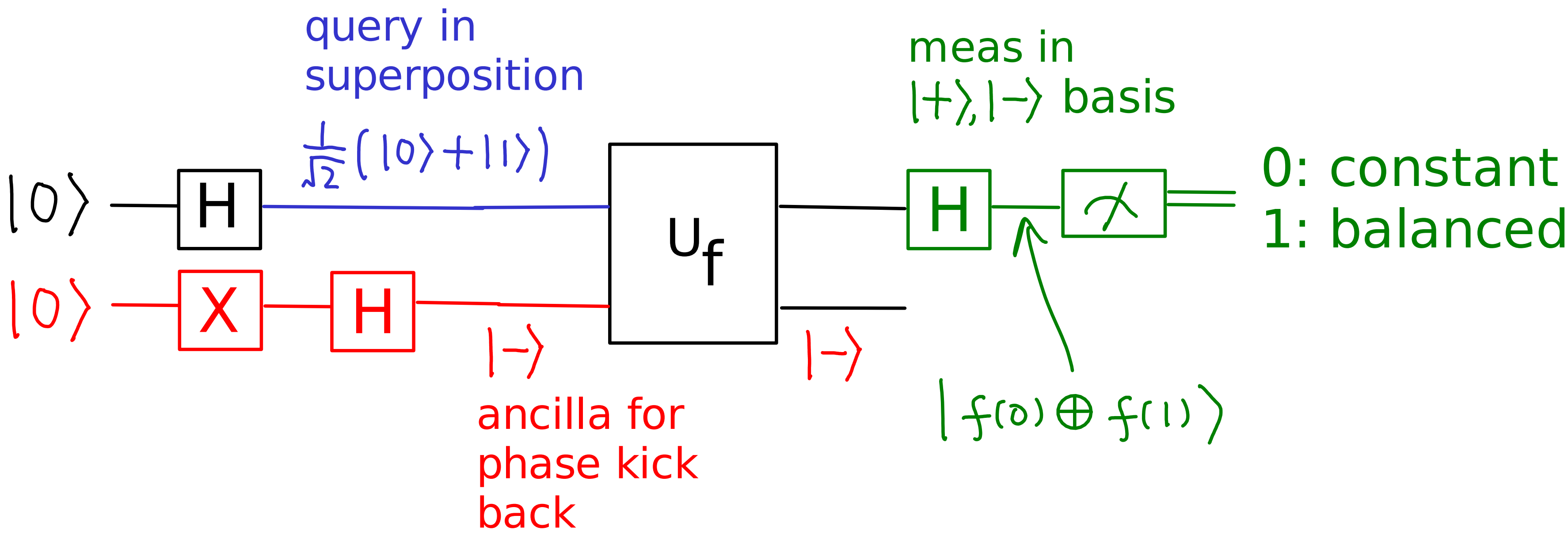
$$= \begin{cases} |+\rangle|-\rangle & \text{if f is constant} \\ |-\rangle|-\rangle & \text{if f is balanced} \end{cases}$$

perfectly distinguishable!

The black box:

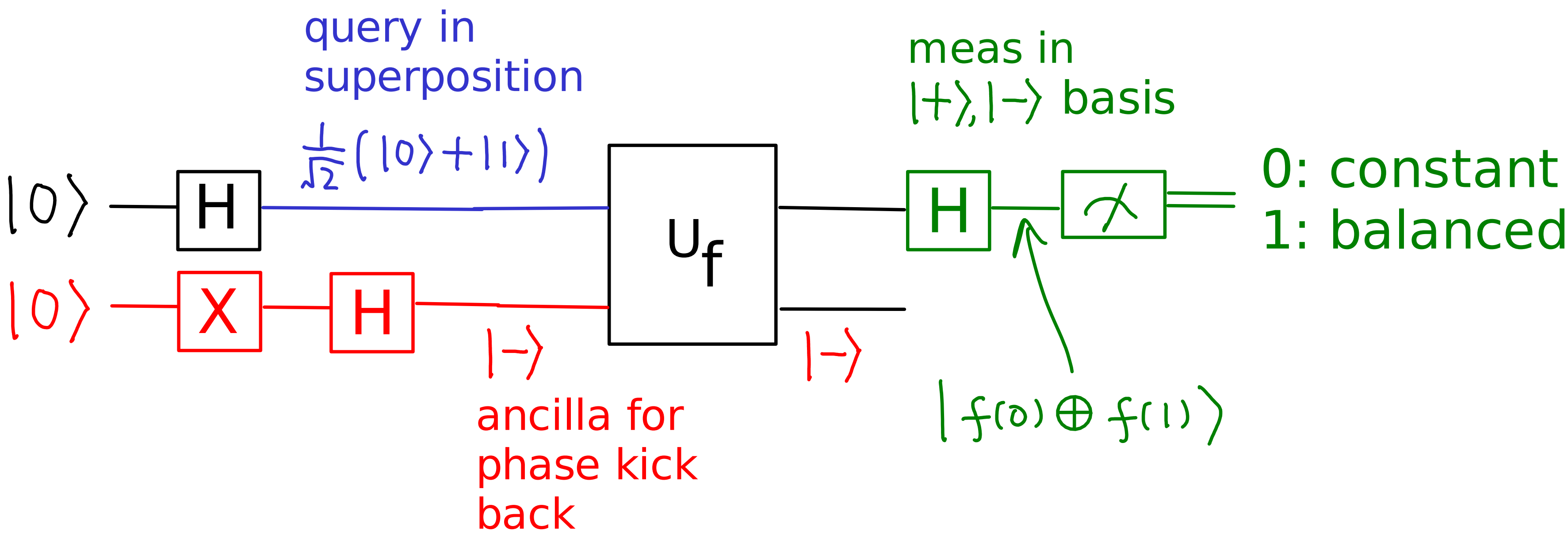


The complete circuit:



Recall: $H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$, $H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)$, $H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle-|1\rangle)$.

The complete circuit:



The relative phase (quantum interference) carries a global property of f which is mapped by the final H to something measurable in the computational basis !

Deutsch-Josza Problem

Given: a black box for a function f : $\{0,1\}^n \longrightarrow \{0,1\}$

Promise (partial information about f):
f is either constant
or balanced (half of the f(x)'s = 0)

Deutsch-Josza Problem

Given: a black box for a function f : $\{0,1\}^n \longrightarrow \{0,1\}$

Promise (partial information about f):
  f is either constant
    or balanced (half of the f(x)'s = 0)

e.g., n=3, a balanced function is

f(000) = 0	f(100)=1
f(001) = 0	f(101)=1
f(010) = 1	f(110)=1
f(011) = 0	f(111)=0

Deutsch-Josza Problem

Given: a black box for a function f : $\{0,1\}^n \longrightarrow \{0,1\}$

Promise (partial information about f):
f is either constant
or balanced (half of the f(x)'s = 0)

Problem: Is f constant or balanced?

Question:

Classically, how may queries are needed to solve the D-J problem for the worse f deterministically ?

(a) 2 (b) 2^{n-1} (c) $2^{n-1}+1$ (d) 2^n

Deutsch-Josza Problem

Given: a black box for a function f : $\{0,1\}^n \longrightarrow \{0,1\}$

Promise (partial information about f):
f is either constant
or balanced (half of the f(x)'s = 0)

Problem: Is f constant or balanced?

Classically, 2^{n-1} +1 queries are needed.

to solve the problem for the worst f deterministically

Deutsch-Josza Problem

Given: a black box for a function f : $\{0,1\}^n \longrightarrow \{0,1\}$

Promise (partial information about f):
f is either constant
or balanced (half of the f(x)'s = 0)

Problem: Is f constant or balanced?

Classically, 2^{n-1} +1 queries are needed.

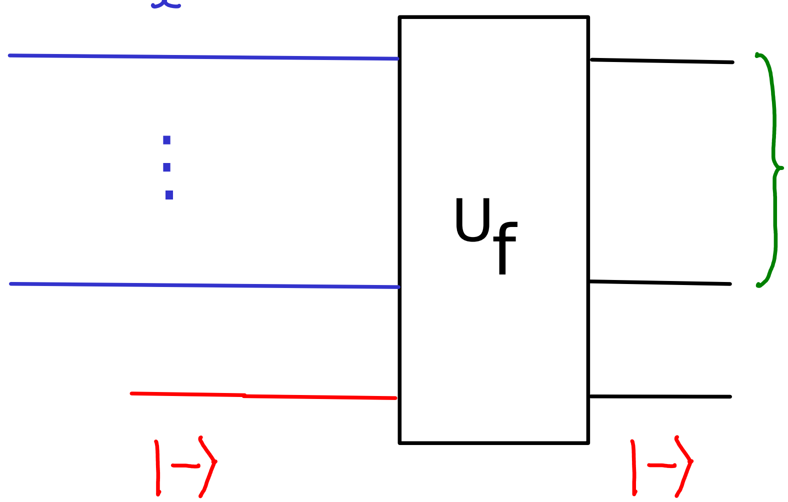
to solve the problem for the worst f deterministically

Quantumly, 1 query suffices!

The black box:

query in superposition

$\frac{1}{\sqrt{2^n}}\sum_x |x\rangle$



$|-\rangle$

ancilla for phase kick back

$|-\rangle$

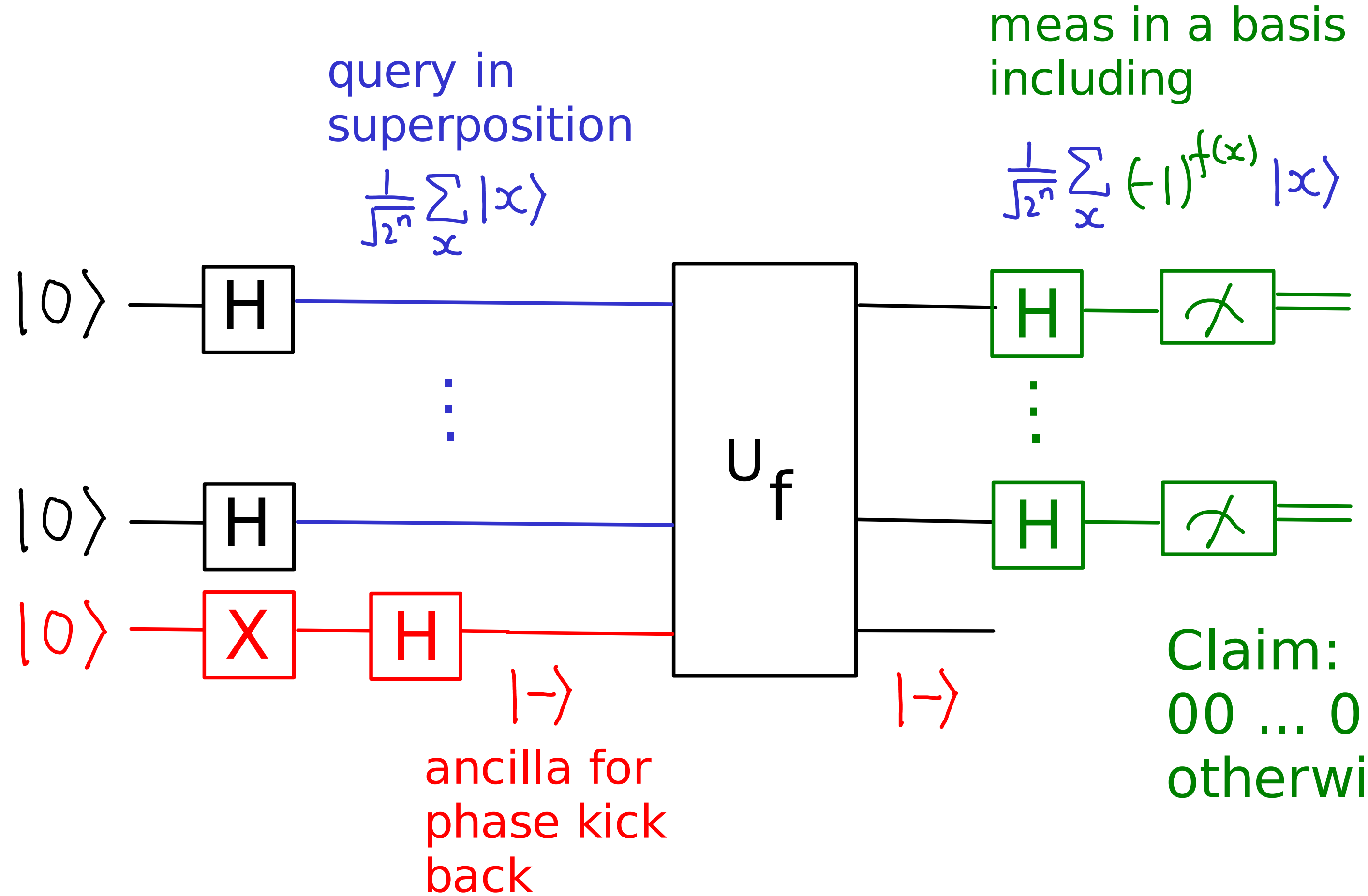
$\frac{1}{\sqrt{2^n}}\sum_x (-1)^{f(x)}|x\rangle$

if f constant, $= \pm\frac{1}{\sqrt{2^n}}\sum_x |x\rangle$

if f balanced, orthogonal to $\frac{1}{\sqrt{2^n}}\sum_x |x\rangle$

Suffices to prepare query and ancilla, and design a measurement to distinguish the two possibilities.

The complete circuit:



Claim:
00 … 0 : constant
otherwise: balanced

Analysis: (checking that the circuit works)

1. Initialize input in superposition & ancilla :

$$|0\rangle^{\otimes n}|0\rangle \xrightarrow{H^{\otimes n} \otimes HX} \frac{1}{\sqrt{2^n}}\sum_x |x\rangle|-\rangle$$

Analysis: (checking that the circuit works)

1. Initialize input in superposition & ancilla :

$$|0\rangle^{\otimes n}|0\rangle \xrightarrow{H^{\otimes n} \otimes HX} \frac{1}{\sqrt{2^n}} \sum_x |x\rangle |-\rangle$$

2. Apply blackbox with phase kick back:

$$\frac{1}{\sqrt{2^n}} \sum_x |x\rangle |-\rangle \xrightarrow{U_f} \frac{1}{\sqrt{2^n}} \sum_x |x\rangle (-1)^{f(x)} |-\rangle$$

Analysis: (checking that the circuit works)

1. Initialize input in superposition & ancilla :

$$|0\rangle^{\otimes n}|0\rangle \xrightarrow{H^{\otimes n} \otimes HX} \frac{1}{\sqrt{2^n}}\sum_x |x\rangle|-\rangle$$

2. Apply blackbox with phase kick back:

$$\frac{1}{\sqrt{2^n}}\sum_x |x\rangle|-\rangle \xrightarrow{U_f} \frac{1}{\sqrt{2^n}}\sum_x |x\rangle(-1)^{f(x)}|-\rangle$$

3. Apply Hadamard to "first register" (first n qubits):

$$\frac{1}{\sqrt{2^n}}\sum_x |x\rangle(-1)^{f(x)} \xrightarrow{H^{\otimes n}} \frac{1}{2^n}\sum_x \sum_y |y\rangle(-1)^{x\cdot y}(-1)^{f(x)}$$

derived next page

The Fourier transform:

For 1 qubit: $H|x\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + (-1)^x|1\rangle\right) = \frac{1}{\sqrt{2}}\sum_{y\in\{0,1\}}(-1)^{x\cdot y}|y\rangle$

The Fourier transform:

For 1 qubit: $H|x\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + (-1)^x |1\rangle\right) = \frac{1}{\sqrt{2}} \sum_{y \in \{0,1\}} (-1)^{x \cdot y} |y\rangle$

For n qubits: $H^{\otimes n} |x_1 x_2 \cdots x_n\rangle$

$$= (H|x_1\rangle) \otimes (H|x_2\rangle) \cdots \otimes (H|x_n\rangle)$$

The Fourier transform:

For 1 qubit: $H|x\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + (-1)^{x}|1\rangle\right) = \frac{1}{\sqrt{2}}\sum_{y\in\{0,1\}} (-1)^{x\cdot y}|y\rangle$

For n qubits: $H^{\otimes n}|x_1 x_2 \cdots x_n\rangle$

$$= (H|x_1\rangle) \otimes (H|x_2\rangle) \cdots \otimes (H|x_n\rangle)$$

$$= \left(\frac{1}{\sqrt{2}}\sum_{y_1\in\{0,1\}} (-1)^{x_1\cdot y_1}|y_1\rangle\right) \otimes \left(\frac{1}{\sqrt{2}}\sum_{y_2\in\{0,1\}} (-1)^{x_2\cdot y_2}|y_2\rangle\right) \otimes \cdots$$

$$\cdots \otimes \left(\frac{1}{\sqrt{2}}\sum_{y_n\in\{0,1\}} (-1)^{x_n\cdot y_n}|y_n\rangle\right)$$

The Fourier transform:

For 1 qubit: $H|x\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + (-1)^x|1\rangle\right) = \frac{1}{\sqrt{2}}\sum_{y\in\{0,1\}}(-1)^{x\cdot y}|y\rangle$

For n qubits: $H^{\otimes n}|x_1 x_2 \cdots x_n\rangle$

$$= (H|x_1\rangle) \otimes (H|x_2\rangle) \cdots \otimes (H|x_n\rangle)$$

$$= \left(\frac{1}{\sqrt{2}}\sum_{y_1\in\{0,1\}}(-1)^{x_1\cdot y_1}|y_1\rangle\right) \otimes \left(\frac{1}{\sqrt{2}}\sum_{y_2\in\{0,1\}}(-1)^{x_2\cdot y_2}|y_2\rangle\right) \otimes \cdots$$

$$\cdots \otimes \left(\frac{1}{\sqrt{2}}\sum_{y_n\in\{0,1\}}(-1)^{x_n\cdot y_n}|y_n\rangle\right)$$

$$= \frac{1}{\sqrt{2^n}}\sum_{y_1 y_2 \cdots y_n \in\{0,1\}^n}(-1)^{x_1y_1+x_2y_2+\cdots+x_ny_n}|y_1 y_2 \cdots y_n\rangle$$

The Fourier transform:

For 1 qubit: $H|x\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + (-1)^x |1\rangle\right) = \frac{1}{\sqrt{2}} \sum_{y\in\{0,1\}} (-1)^{x\cdot y} |y\rangle$

For n qubits: $H^{\otimes n} |x_1 x_2 \cdots x_n\rangle$

$$= (H|x_1\rangle) \otimes (H|x_2\rangle) \cdots \otimes (H|x_n\rangle)$$

$$= \left(\frac{1}{\sqrt{2}} \sum_{y_1\in\{0,1\}} (-1)^{x_1\cdot y_1} |y_1\rangle\right) \otimes \left(\frac{1}{\sqrt{2}} \sum_{y_2\in\{0,1\}} (-1)^{x_2\cdot y_2} |y_2\rangle\right) \otimes \cdots$$

$$\cdots \otimes \left(\frac{1}{\sqrt{2}} \sum_{y_n\in\{0,1\}} (-1)^{x_n\cdot y_n} |y_n\rangle\right)$$

$$= \frac{1}{\sqrt{2^n}} \sum_{y_1 y_2 \cdots y_n \in \{0,1\}^n} \underbrace{(-1)^{x_1 y_1 + x_2 y_2 + \cdots + x_n y_n}}_{(-1)^{x\cdot y}} |y_1 y_2 \cdots y_n\rangle$$

where $x = x_1 x_2 \cdots x_n$, $y = y_1 y_2 \cdots y_n$.

3. Apply Hadamard to "first register" (first n qubits):

$$\frac{1}{\sqrt{2^n}} \sum_x |x\rangle (-1)^{f(x)} \xrightarrow{H^{\otimes n}} \frac{1}{2^n} \sum_x \sum_y |y\rangle (-1)^{x \cdot y} (-1)^{f(x)}$$

$$= \frac{1}{2^n} \sum_y \sum_x (-1)^{x \cdot y} (-1)^{f(x)} |y\rangle$$

3. Apply Hadamard to "first register" (first n qubits):

$$\frac{1}{\sqrt{2^n}} \sum_x |x\rangle (-1)^{f(x)} \xrightarrow{\quad H^{\otimes n} \quad} \frac{1}{2^n} \sum_x \sum_y |y\rangle (-1)^{x \cdot y} (-1)^{f(x)}$$

$$= \frac{1}{2^n} \sum_y \sum_x (-1)^{x \cdot y} (-1)^{f(x)} |y\rangle$$

If f constant, $(-1)^{f(x)} = c$,

$$\sum_x (-1)^{x \cdot y} (-1)^{f(x)} = c \sum_x (-1)^{x \cdot y} = c \begin{cases} 2^n & \text{if } y = 00\ldots0 \\ 0 & \text{otherwise} \end{cases}$$

3. Apply Hadamard to "first register" (first n qubits):

$$\frac{1}{\sqrt{2^n}}\sum_x |x\rangle (-1)^{f(x)} \xrightarrow{H^{\otimes n}} \frac{1}{2^n}\sum_x \sum_y |y\rangle (-1)^{x\cdot y} (-1)^{f(x)}$$

$$= \frac{1}{2^n}\sum_y \sum_x (-1)^{x\cdot y} (-1)^{f(x)} |y\rangle$$

If f constant, $(-1)^{f(x)} = c$,

$$\sum_x (-1)^{x\cdot y} (-1)^{f(x)} = c \sum_x (-1)^{x\cdot y} = c \begin{cases} 2^n & \text{if } y = 00...0 \\ 0 & \text{otherwise} \end{cases}$$

If f balanced, y = 00...0,

$$\sum_x (-1)^{x\cdot y} (-1)^{f(x)} = \sum_x (-1)^{f(x)} = 0.$$

3. Apply Hadamard to "first register" (first n qubits):

$$\frac{1}{\sqrt{2^n}}\sum_x |x\rangle (-1)^{f(x)} \xrightarrow{H^{\otimes n}} \frac{1}{2^n}\sum_x \sum_y |y\rangle (-1)^{x\cdot y} (-1)^{f(x)}$$

$$= \frac{1}{2^n}\sum_y \sum_x (-1)^{x\cdot y} (-1)^{f(x)} |y\rangle$$

If f constant, $(-1)^{f(x)} = c$,

$$\sum_x (-1)^{x\cdot y} (-1)^{f(x)} = c \sum_x (-1)^{x\cdot y} = c \begin{cases} 2^n & \text{if } y = 00...0 \\ 0 & \text{otherwise} \end{cases}$$

If f balanced, y = 00...0,

$$\sum_x (-1)^{x\cdot y} (-1)^{f(x)} = \sum_x (-1)^{f(x)} = 0.$$

$$\therefore \sum_x (-1)^{x\cdot y} (-1)^{f(x)} \begin{cases} \text{nonzero only for y=0 if f constant} \\ \text{zero for y=0 if f balanced} \end{cases}$$

4. Measure first register in computational basis:

$$\frac{1}{2^n}\sum_y \sum_x (-1)^{x\cdot y}(-1)^{f(x)}|y\rangle \longrightarrow \begin{array}{l}y=0 \text{ if f constant}\\ y\neq 0 \text{ if f balanced}\end{array}$$

nonzero only for y=0 if f constant, outcome y=0 always
zero for y=0 if f balanced, outcome never being y=0.

So, circuit works and 1 query suffices !

You saw the first exponential separation between quantum and classical computation, in the blackbox model if the answer must be correct.

If we allow a small error, classically, a constant # of queries suffices.

We will see more algorithms revolving about the Fourier transform, and the advantage will be over BPP, and eventually outside of the blackbox model.

The Deutsch problem with solution was first proposed by Deutsch in 1985. In 1992, it was extended to the Deutsch-Jozsa problem and algorithm.

The algorithm you saw today is an improved version from Cleve, Ekert, Macchiavello, and Mosca 1998, and independently, by Tapp.