POPL 2022

Sun, 16 Jan 2022 – Fri, 28 Jan 2022
Westin Philadelphia, US

A Quantum Interpretation of Separating Conjunction for Local Reasoning of Quantum Programs Based on Separation Logic

Xuan-Bach Le[1], Shang-Wei Lin[1], Jun Sun[2], David Sanan[1]

[1]Nanyang Technological University [2]Singapore Management University

In this talk

Local reasoning for quantum computation

In this talk

Local reasoning for quantum computation

(with a user-friendly and intuitive mindset)

Fill in the blank

It is hard to ……. quantum programs

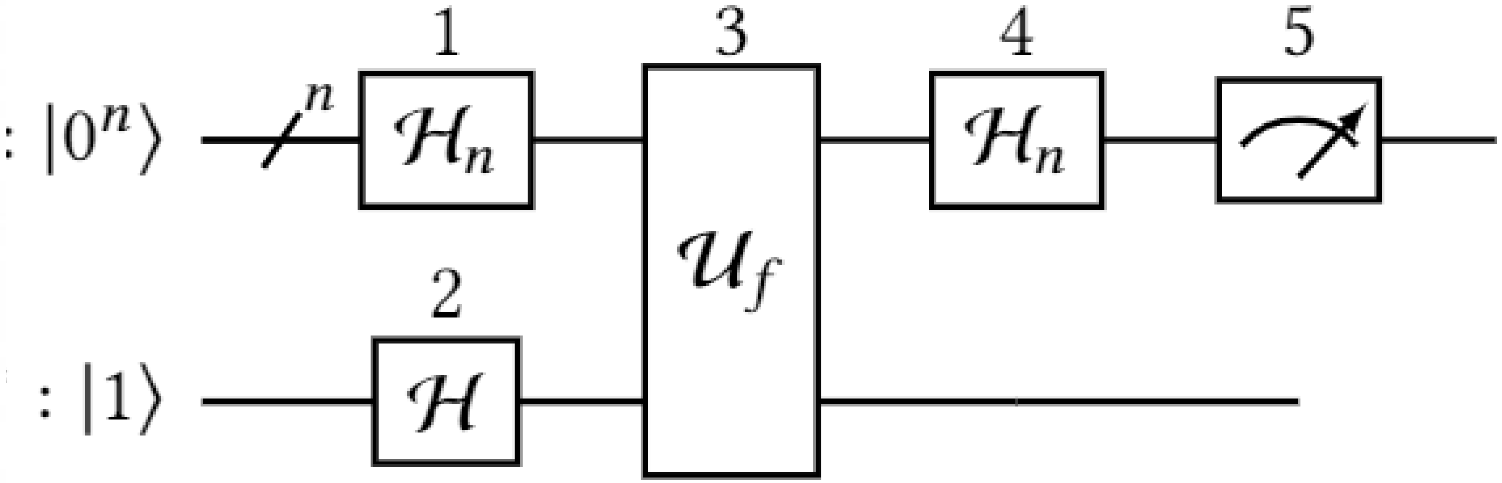
A. Write

B. Understand

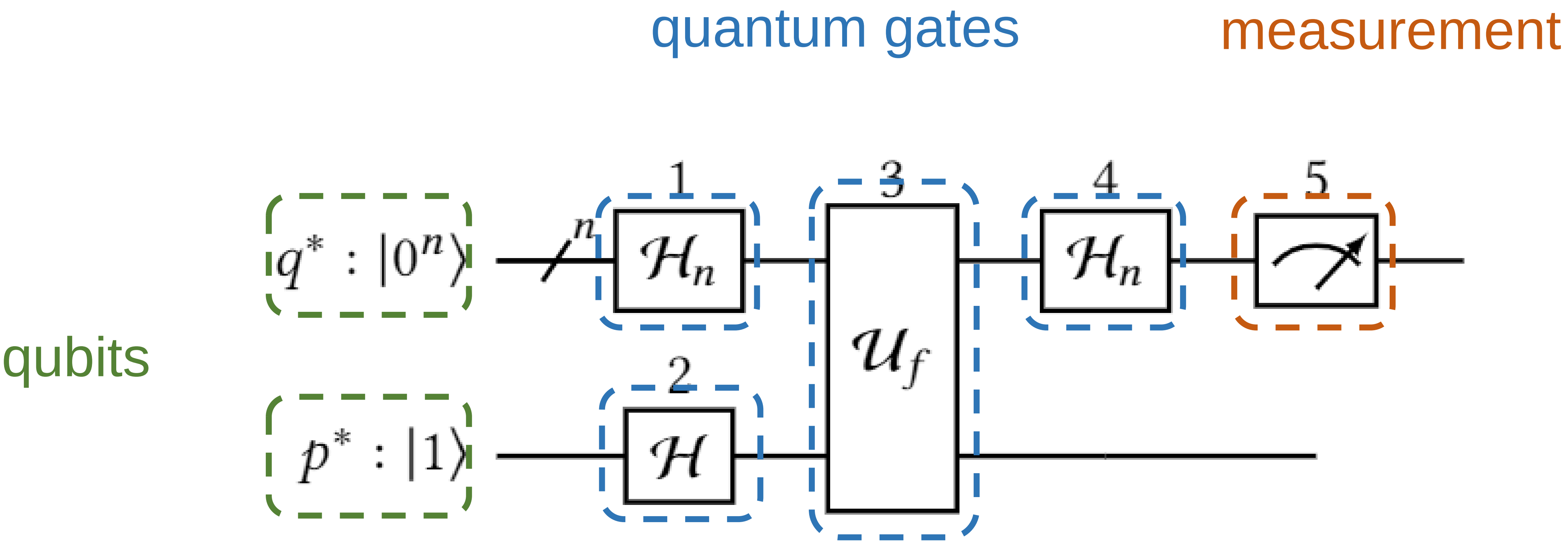
C. Verify

D. All of the above (‘superpositionally’)

Quantum programs: Example



Quantum programs: Example



Quantum programs: Example

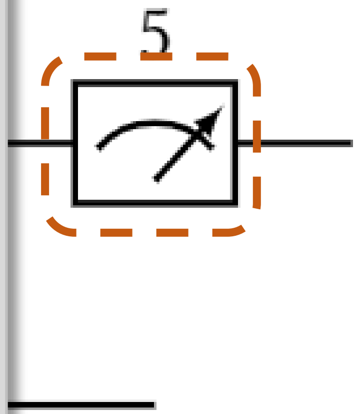
qubits

q^* :

p^*

1 $\mathcal{H}_n(q^*)$;
2 $\mathcal{H}(p^*)$;
3 $\mathcal{U}_f(q^* p^*)$;
4 $\mathcal{H}_n(q^*)$;
5 $v := \mathtt{measure}(q^*)$;

measurement



Quantum programs: Example

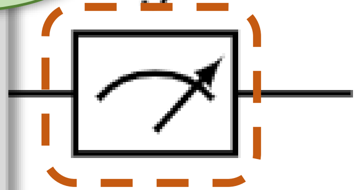
$\mathcal{H}_n(q^*);$
$\mathcal{H}(p^*);$
$\mathcal{U}_f(q^*p^*);$
$\mathcal{H}_n(q^*);$
$v := \texttt{measure}(q^*);$

Is this better?

qubits

q^* :

p^*



ent

Example: Deutsch's algorithm

$$f: \{0,1\} \mapsto \{0,1\}$$

f is constant

$$f(0) = f(1)$$

f is balanced

$$f(0) \neq f(1)$$

Example: Deutsch's algorithm

$$f:\{0,1\} \mapsto \{0,1\}$$

f is constant

$$f(0) = f(1)$$

f is balanced

$$f(0) \neq f(1)$$

Example: Deutsch's algorithm

$$f: \{0,1\} \mapsto \{0,1\}$$

f is constant

$$f(0) = f(1)$$

f is balanced

$$f(0) \neq f(1)$$

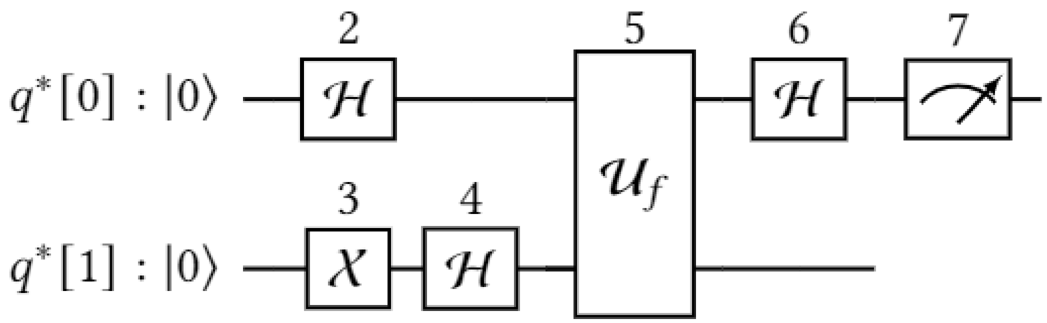
Classical algorithm: evaluate f twice

Quantum algorithm: evaluate f once

Example: Deutsch’s algorithm

```
q* := qbit(2);
H(q*[0]);
X(q*[1]);
H(q*[1]);
U_f(q*);
H(q*[0]);
v := measure(q*[0]);
dispose(q*);
```

(a) The algorithm’s code



(b) The algorithm’s circuit design

Superposition

Classical computation: **Bit**

0 or 1

Superposition

Quantum computation: **Quit**

0 ‘and’ 1

Superposition

Parallelism

Quantum computation: **Quit**

0 ‘and’ 1

“A good quantum computer algorithm ensures that computational paths leading to a wrong answer cancel out and that paths leading to a correct answer reinforce.”

Scott Aaronson

Example: Deutsch's algorithm

1 $q^* := \texttt{qbit}(2);$

2 $\mathcal{H}(q^*[0]);$

3 $\mathcal{X}(q^*[1]);$

4 $\mathcal{H}(q^*[1]);$

5 $\mathcal{U}_f(q^*);$

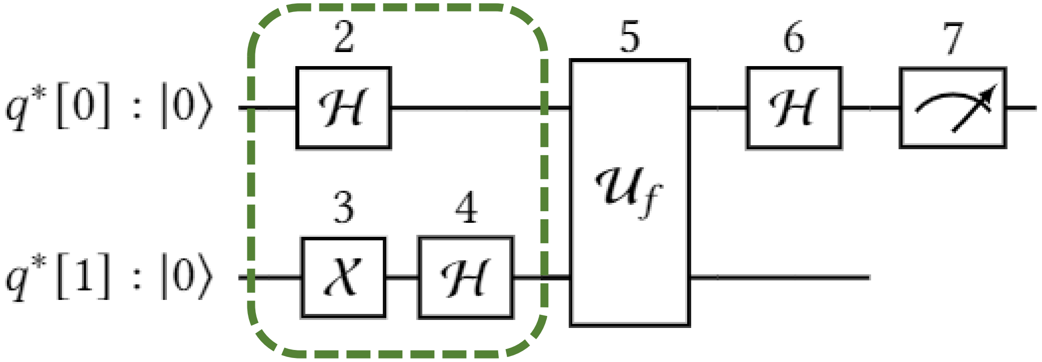
6 $\mathcal{H}(q^*[0]);$

7 $v := \texttt{measure}(q^*[0]);$

8 $\texttt{dispose}(q^*);$

(a) The algorithm's code

encode the superposition

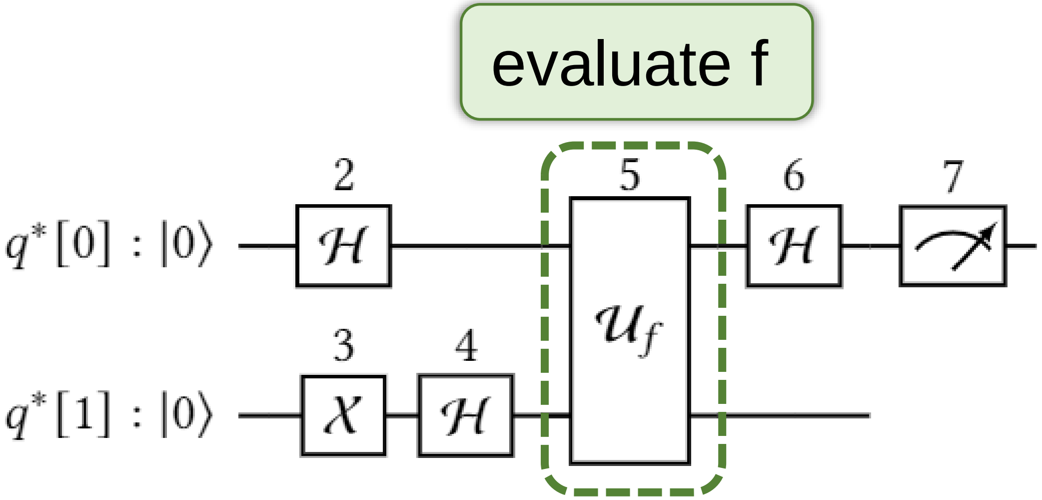


(b) The algorithm's circuit design

Example: Deutsch's algorithm

```
q* := qbit(2);
H(q*[0]);
X(q*[1]);
H(q*[1]);
U_f(q*);
H(q*[0]);
v := measure(q*[0]);
dispose(q*);
```

(a) The algorithm's code

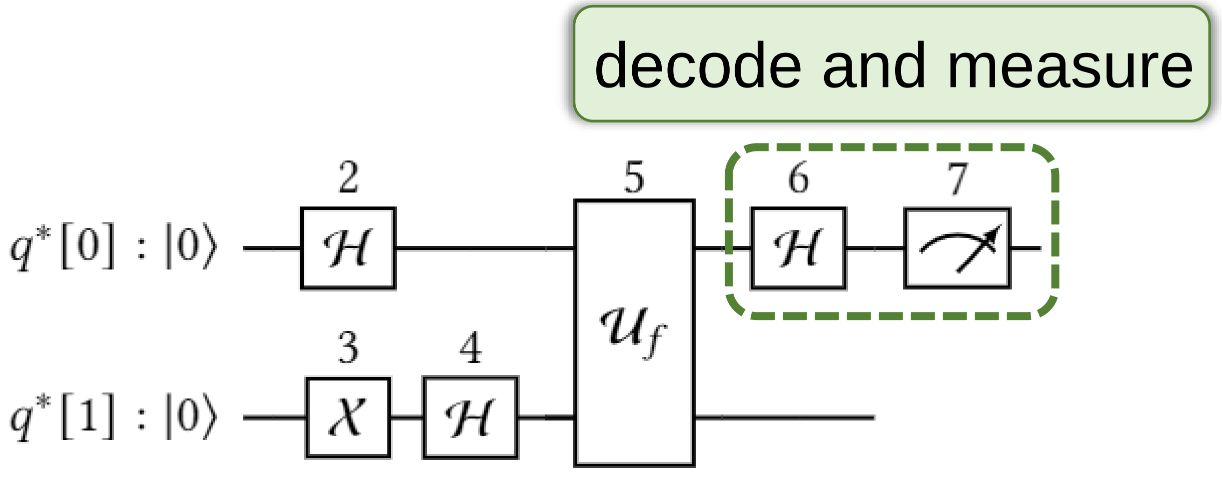


(b) The algorithm's circuit design

Example: Deutsch's algorithm

```
1  q* := qbit(2);
2  H(q*[0]);
3  X(q*[1]);
4  H(q*[1]);
5  U_f(q*);
6  H(q*[0]);
7  v := measure(q*[0]);
8  dispose(q*);
```

(a) The algorithm's code



(b) The algorithm's circuit design

Programming language

$$
\begin{aligned}
c \quad ::= \quad & \mathbf{skip} \mid x := e \mid \mathbf{if}\ b\ \mathbf{do}\ c\ \mathbf{else}\ c \mid \mathbf{while}\ b\ \mathbf{do}\ c \mid c\ ;c \mid \\
& q^* := \mathbf{qbit}(e) \mid \mathcal{G}(e^*) \mid x := \mathbf{measure}(e^*) \mid \mathbf{dispose}(q^*)
\end{aligned}
$$

Programming language

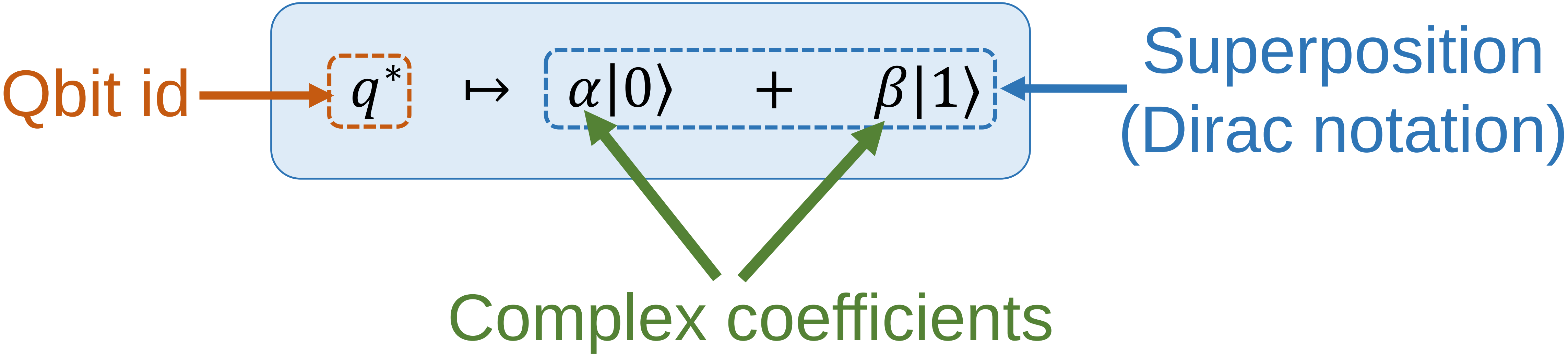
$$c \quad ::= \quad \mathbf{skip} \mid x := e \mid \mathbf{if}\ b\ \mathbf{do}\ c\ \mathbf{else}\ c \mid \mathbf{while}\ b\ \mathbf{do}\ c \mid c\ ;c \mid$$
$$q^* := \mathbf{qbit}(e) \mid \mathcal{G}(e^*) \mid x := \mathbf{measure}(e^*) \mid \mathbf{dispose}(q^*)$$

Qubit allocation and deallocation

Superposition

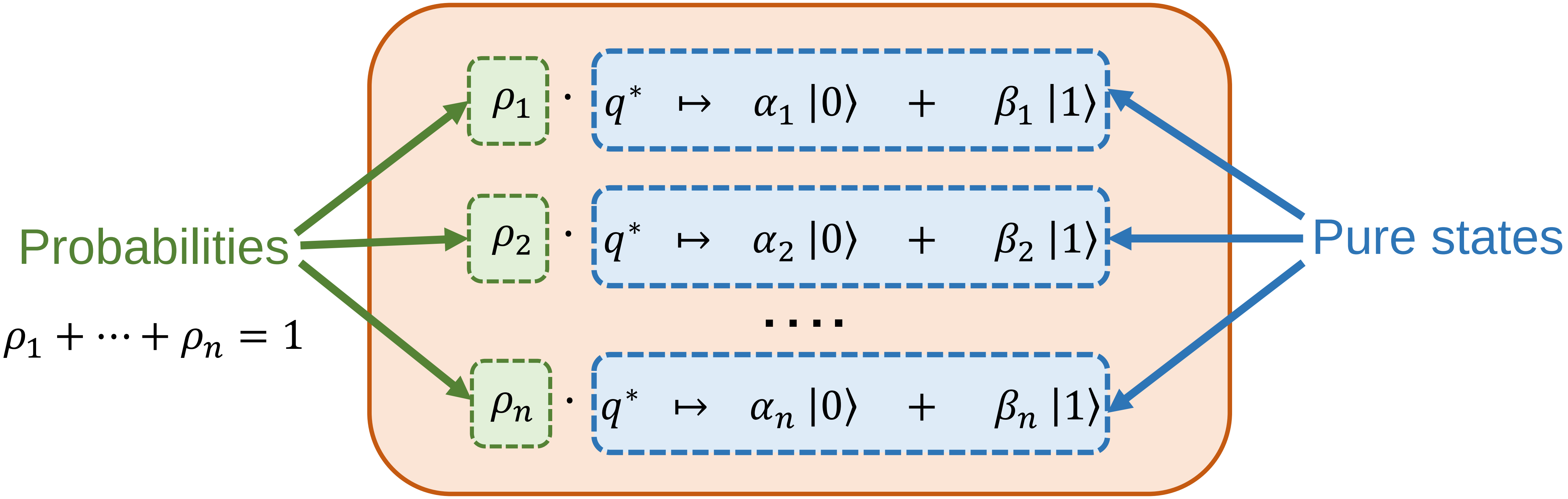
$$q^* \mapsto \alpha|0\rangle + \beta|1\rangle$$

Superposition



$$|\alpha|^2 + |\beta|^2 = 1$$

Mixed state



Probabilities
$\rho_1 + \cdots + \rho_n = 1$
ρ_1 · $q^* \mapsto \alpha_1 |0\rangle + \beta_1 |1\rangle$
ρ_2 · $q^* \mapsto \alpha_2 |0\rangle + \beta_2 |1\rangle$
....
ρ_n · $q^* \mapsto \alpha_n |0\rangle + \beta_n |1\rangle$
Pure states

A simpler representation

Pure state tagged with probability

$$\rho \cdot q^* \mapsto \alpha|0\rangle + \beta|1\rangle$$

A simpler representation

Pure state tagged with probability

$$\rho \cdot q^* \mapsto \alpha|0\rangle + \beta|1\rangle$$

Pros: simple, intuitive, local reasoning

Cons: expressiveness, completeness

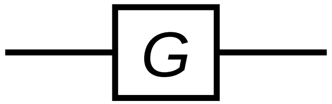
Programming language

$$
\begin{aligned}
c \quad ::= \quad & \mathbf{skip} \mid x := e \mid \mathbf{if}\ b\ \mathbf{do}\ c\ \mathbf{else}\ c \mid \mathbf{while}\ b\ \mathbf{do}\ c \mid c\ ; c \mid \\
& q^* := \mathbf{qbit}(e) \mid \mathcal{G}(e^*) \mid x := \mathbf{measure}(e^*) \mid \mathbf{dispose}(q^*)
\end{aligned}
$$

Transformation

Transformation

$\rho \quad \cdot \quad q^* \mapsto |s\rangle$

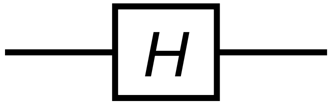


$\rho \quad \cdot \quad q \mapsto G|s\rangle$

Transformation: Example

Hadamard gate: $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$

$q^* \mapsto |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$



$q^* \mapsto H|0\rangle = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$

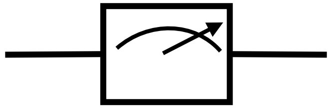
Programming language

$$
\begin{aligned}
c \quad ::= \quad & \mathbf{skip} \mid x := e \mid \mathbf{if}\ b\ \mathbf{do}\ c\ \mathbf{else}\ c \mid \mathbf{while}\ b\ \mathbf{do}\ c \mid c\ ; c \mid \\
& q^* := \mathbf{qbit}(e) \mid \mathcal{G}(e^*) \mid \boxed{x := \mathbf{measure}(e^*)} \mid \mathbf{dispose}(q^*)
\end{aligned}
$$

Measurement

Measurement

$$\rho \quad \cdot \quad q^* \mapsto \alpha_0|0\rangle + \cdots \alpha_n|n\rangle$$



$$\rho|\alpha_0|^2 \quad \cdot \quad q^* \mapsto |0\rangle$$

V

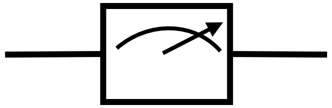
....

V

$$\rho|\alpha_n|^2 \cdot \quad q^* \mapsto |n\rangle$$

Measurement: Example

$$\frac{1}{2} \cdot q^* \mapsto \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$$



$$\frac{1}{4} \cdot q^* \mapsto |0\rangle$$

$$\vee$$

$$\frac{1}{4} \cdot q^* \mapsto |1\rangle$$

Programming language

$$
\begin{aligned}
c \quad ::= \quad & \textbf{skip} \mid x := e \mid \textbf{if}\ b\ \textbf{do}\ c\ \textbf{else}\ c \mid \textbf{while}\ b\ \textbf{do}\ c \mid c\ ;c\ \mid \\
& q^* := \textbf{qbit}(e) \mid \mathcal{G}(e^*) \mid x := \textbf{measure}(e^*) \mid \textbf{dispose}(q^*)
\end{aligned}
$$

Quantum rules

Allocation

$$\frac{}{\{|\mathbf{emp}\rangle \wedge e = n > 0\}q^* := \mathbf{qbit}(e)\{q^*[0, n-1] \mapsto |0^n\rangle \wedge |q^*| = n\}}\ \text{Qubit}$$

Deallocation

$$\frac{}{\{q^*[0, n-1] \mapsto |v\rangle \wedge |q^*| = n\}\mathbf{dispose}(q^*)\{|\mathbf{emp}\rangle\}}\ \text{Dis}$$

Transformation

$$\frac{\mathcal{G} : \mathbb{V}_{\mathcal{B}}^{|e^*|} \mapsto \mathbb{V}_{\mathcal{B}}^{|e^*|} \qquad |e_i\rangle \in \mathbb{B}^{|e^*|}}{\{e^*e'^* \mapsto \sum_{i,j} a_{i,j}|e_i\rangle|e'_j\rangle\}\mathcal{G}(e^*)\{e^*e'^* \mapsto \sum_{i,j} a_{i,j}\mathcal{G}(|e_i\rangle)|e'_j\rangle\}}\ \text{Trans}$$

Measurement

$$\frac{\begin{array}{c}|e_i\rangle \in \mathbb{B}^{|e^*|}, |e'_j\rangle \in \mathbb{B}^{|e'^*|} \qquad v \notin \mathrm{free}(\Psi) \qquad \rho_i \triangleq \sum_j |a_{i,j}|^2 \\ \Psi \triangleq e^*e'^* \mapsto \sum_{i,j} a_{i,j}|e_i\rangle|e'_j\rangle \qquad \Psi_i \triangleq e^* \mapsto |e_i\rangle \circledast e'^* \mapsto \sum_j \frac{a_{i,j}}{\sqrt{\rho_i}}|e'_j\rangle\end{array}}{\{\Psi \wedge (\bigwedge_i \overline{\Phi}_i[v/e_i])\}v := \mathbf{measure}(e^*)\{\bigvee_i(\rho_i \cdot \Psi_i \wedge \overline{\Phi}_i)\}}\ \text{Ms}$$

Frame rule

$$\frac{\{P\}c\{Q\} \qquad FV(F) \cap MV(c) = \emptyset}{\{P \star F\}c\{Q \star F\}}$$

Frame rule

$$\frac{\{P\}c\{Q\} \qquad FV(F) \cap MV(c) = \emptyset}{\{P \star F\}c\{Q \star F\}}$$

Key idea: qubits as resources

$$F_1 \star F_2$$

The qubits in F_1 and F_2 are disjoint and their states can be expressed independently

Multiple qubits

$$\rho \cdot q_1^* q_2^* \mapsto \alpha|00\rangle + \beta|01\rangle + \delta|10\rangle + \omega|11\rangle$$

$$|01\rangle = |0\rangle \otimes |1\rangle$$

tensor product

$$|\alpha|^2 + |\beta|^2 + |\delta|^2 + |\omega|^2 = 1$$

Factorization for local reasoning

$$\rho_1\rho_2 \;\cdot\; q_1^* q_2^* \mapsto |s_1\rangle \otimes |s_2\rangle$$

$$\Updownarrow$$

$$\rho_1 \cdot q_1^* \mapsto |s_1\rangle \quad \star \quad \rho_2 \cdot q_2^* \mapsto |s_2\rangle$$

- $\|s_1\| = \|s_2\| = 1$
- $|s_1\rangle, |s_2\rangle, \rho_1, \rho_2$ are not unique (differ by a constant)

Factorization for local reasoning

$$q_1^* q_2^* \mapsto |01\rangle$$

Factorization for local reasoning

$$q_1^* q_2^* \mapsto |01\rangle$$

$$q_1^* \mapsto |0\rangle \;\star\; q_2^* \mapsto |1\rangle$$

Factorization for local reasoning

$$q_1^* q_2^* \mapsto |01\rangle$$

$$q_1^* \mapsto |0\rangle \;\star\; q_2^* \mapsto |1\rangle \quad \equiv \quad q_1^* \mapsto i|0\rangle \;\star\; q_2^* \mapsto -i|1\rangle$$

$(i^2 = -1)$

Factorization for local reasoning

$$q_1^* q_2^* \quad \mapsto \quad \frac{1}{2}|00\rangle + \frac{1}{2}|01\rangle - \frac{1}{2}|10\rangle - \frac{1}{2}|11\rangle$$

Factorization for local reasoning

$$q_1^* q_2^* \quad \mapsto \quad \frac{1}{2}|00\rangle + \frac{1}{2}|01\rangle - \frac{1}{2}|10\rangle - \frac{1}{2}|11\rangle$$

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

Factorization for local reasoning

$$q_1^* q_2^* \quad \mapsto \quad \frac{1}{2}|00\rangle + \frac{1}{2}|01\rangle - \frac{1}{2}|10\rangle - \frac{1}{2}|11\rangle$$

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

$$\downarrow$$

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

Factorization for local reasoning

$$q_1^* q_2^* \quad \mapsto \quad \frac{1}{2}|00\rangle + \frac{1}{2}|01\rangle - \frac{1}{2}|10\rangle - \frac{1}{2}|11\rangle$$

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

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

$$q_1^* \mapsto \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle \quad \star \quad q_2^* \mapsto \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$$

Factorization for local reasoning

$$q_1^* q_2^* \mapsto \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$$

⬇

$$q_1^* \mapsto \ ??? \ \star \ q_2^* \mapsto \ ???$$ ✗

Quantum heaps

	h
probability	$\boldsymbol{\rho}$
quantum state	$\boldsymbol{Q}$
classical state	$\boldsymbol{\sigma}$

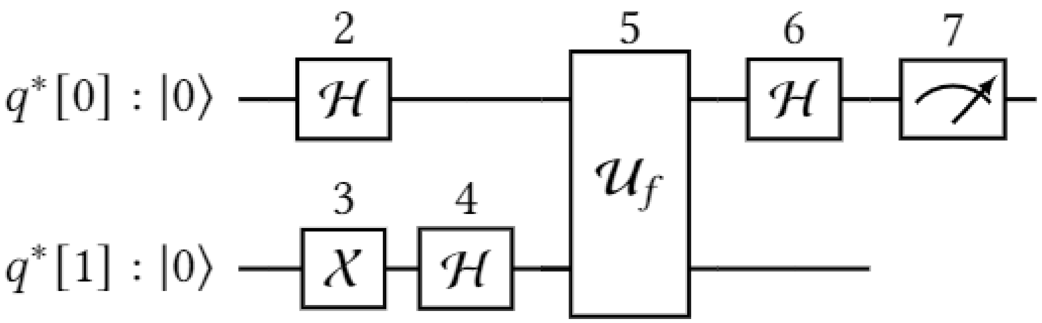
Quantum heaps

	h		h'		$h * h'$
probability	$\boldsymbol{\rho}$		$\boldsymbol{\rho}'$		$\boldsymbol{\rho}\boldsymbol{\rho}'$
quantum state	$\boldsymbol{Q}$	$*$	$\boldsymbol{Q}'$	$=$	$\boldsymbol{Q} \otimes \boldsymbol{Q}'$
classical state	$\boldsymbol{\sigma}$		$\boldsymbol{\sigma}$		$\boldsymbol{\sigma}$

Example: Deutsch's algorithm

$q^* := \mathbf{qbit}(2);$
$\mathcal{H}(q^*[0]);$
$\mathcal{X}(q^*[1]);$
$\mathcal{H}(q^*[1]);$
$\mathcal{U}_f(q^*);$
$\mathcal{H}(q^*[0]);$
$v := \mathbf{measure}(q^*[0]);$
$\mathbf{dispose}(q^*);$

(a) The algorithm's code



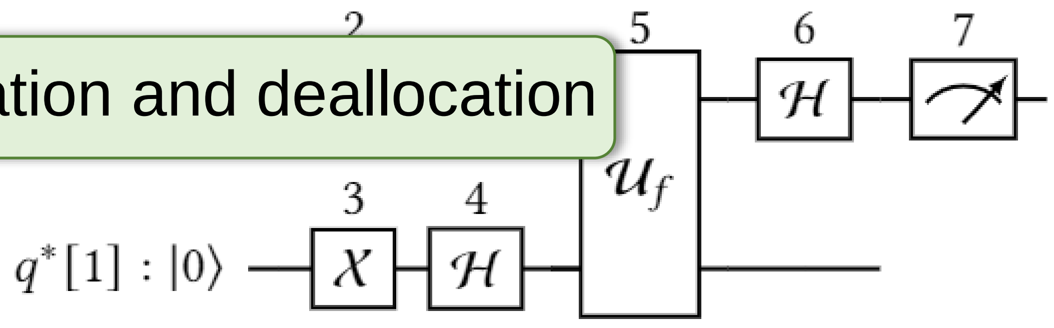
(b) The algorithm's circuit design

Example: Deutsch's algorithm

$q^* := \texttt{qbit}(2);$
$\mathcal{H}(q^*[0]);$
$\mathcal{X}(q^*[1]);$
$\mathcal{H}(q^*[1]);$
$\mathcal{U}_f(q^*);$
$\mathcal{H}(q^*[0]);$
$v := \texttt{measure}(q^*[0]);$
$\texttt{dispose}(q^*);$

(a) The algorithm's code

allocation and deallocation



(b) The algorithm's circuit design

Example: Deutsch's algorithm

$$\{|\boldsymbol{emp}\rangle\}$$

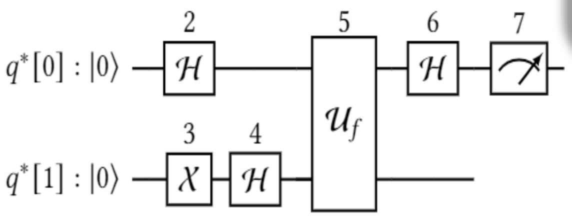
$$C$$

$$\{\underbrace{(f_{=} \leftrightarrow v = 0)}_{\text{constant}} \wedge \underbrace{(f_{\neq} \leftrightarrow v = 1)}_{\text{balanced}} \wedge |\boldsymbol{emp}\rangle\}$$

Example: Deutsch's algorithm

```
q* := qbit(2);
H(q*[0]);
X(q*[1]);
H(q*[1]);
U_f(q*);
H(q*[0]);
v := measure(q*[0]);
dispose(q*);
```

(a) The algorithm's code



(b) The algorithm's circuit design

$$\{|\boldsymbol{emp}\rangle\}$$

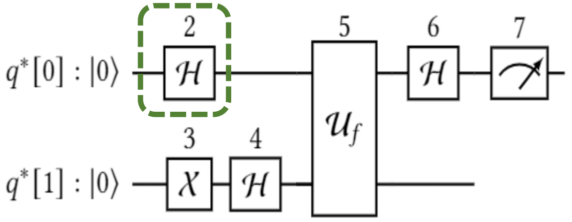
q* := **qubit**(2)

$$\{q^*[0] \mapsto |0\rangle \star q^*[1] \mapsto |0\rangle \wedge |q^*| = 2\}$$

Example: Deutsch’s algorithm

```
q* := qbit(2);
H(q*[0]);
X(q*[1]);
H(q*[1]);
U_f(q*);
H(q*[0]);
v := measure(q*[0]);
dispose(q*);
```

(a) The algorithm’s code



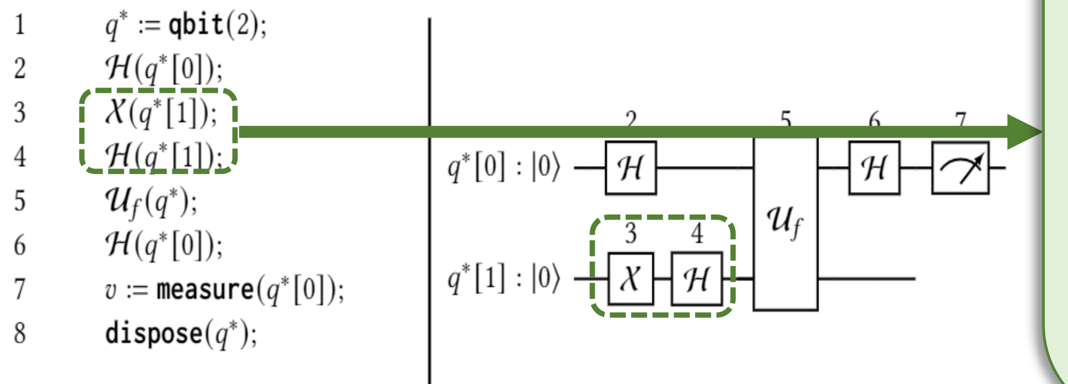
(b) The algorithm’s circuit design

$$\{q^*[0] \mapsto |0\rangle\}$$

H(q*[0])

$$\{q^*[0] \mapsto \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\}$$

Example: Deutsch's algorithm



(a) The algorithm's code

(b) The algorithm's circuit design

$$\{q^*[1] \mapsto |0\rangle\}$$

X(q*[1]);

$$\{q^*[1] \mapsto |1\rangle\}$$

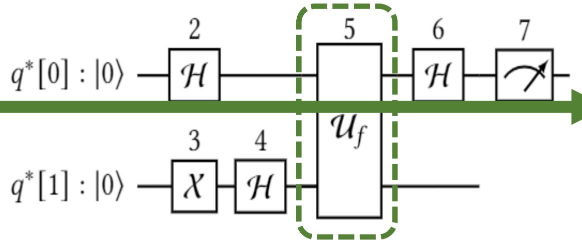
H(q*[1])

$$\{q^*[1] \mapsto \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\}$$

Example: Deutsch's algorithm

```
1    q* := qbit(2);
2    H(q*[0]);
3    X(q*[1]);
4    H(q*[1]);
5    U_f(q*);
6    H(q*[0]);
7    v := measure(q*[0]);
8    dispose(q*);
```

(a) The algorithm's code



(b) The algorithm's circuit design

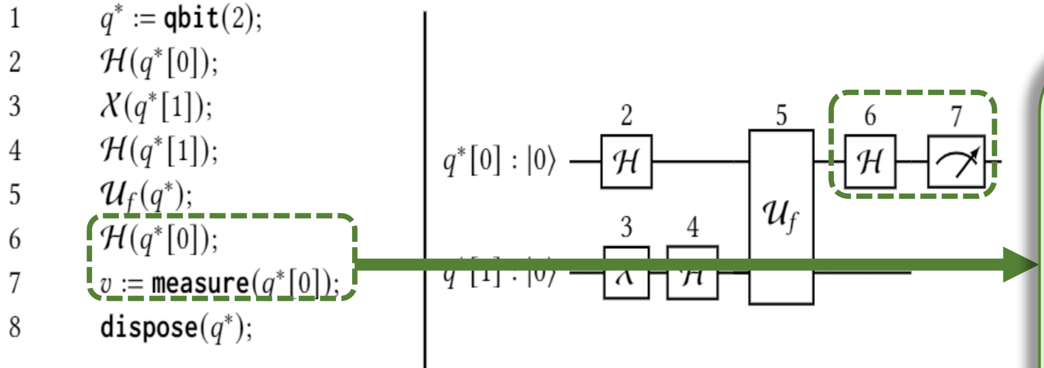
$\{q^* \mapsto \cdots\}$

Uf(q*)

$$\{q^* \mapsto (\frac{1}{\sqrt{2}}|0\rangle + \frac{(-1)^{f(0)\oplus f(1)}}{\sqrt{2}}|1\rangle) \otimes (-1)^{f(0)}|-\rangle\}$$

$$\{q^*[0] \mapsto \frac{1}{\sqrt{2}}|0\rangle + \frac{(-1)^{f(0)\oplus f(1)}}{\sqrt{2}}|1\rangle \star q^*[1] \mapsto (-1)^{f(0)}|-\rangle\}$$

Example: Deutsch's algorithm



(a) The algorithm's code

(b) The algorithm's circuit design

$\{q^*[0] \mapsto \cdots\}$

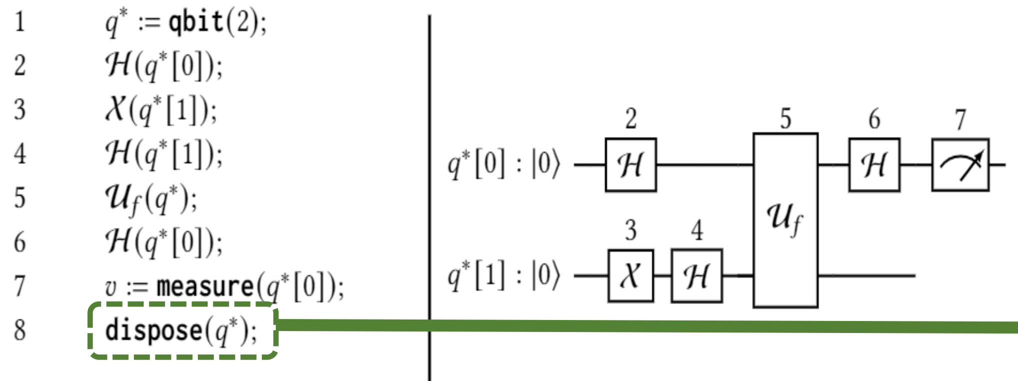
H(q*[0]);

$\{(f_= \wedge q^*[0] \mapsto |0\rangle) \;\vee\; (f_\neq \wedge q^*[0] \mapsto |1\rangle)\}$

v := **measure**(q*[0])

$\{(v = 0 \wedge f_= \wedge q^*[0] \mapsto |0\rangle) \vee (v = 1 \wedge f_\neq \wedge q^*[0] \mapsto |1\rangle)\}$

Example: Deutsch's algorithm



(a) The algorithm's code

(b) The algorithm's circuit design

$\{q^* \mapsto \cdots\}$

dispose(q*)

$\{|\boldsymbol{emp}\rangle\}$

Conclusion

Local reasoning for quantum computation

- Pure states + probabilities
- Separability via tensor-factorization
- Deutsch-Josza’s algorithm, Grover’s algorithm,…

Future works

- Completeness, entanglement
- Mechanization, decision procedures, automatic reasoning