Quantum Computing, an Introduction

Simon Perdrix

Inria, Mocqua/Loria

QComical

3 Nov 2025

https://qcomical2025.github.io/

QCOMICAL School 2025

on Quantum and Classical Programming Languages and Semantics

NOVEMBER 3 TO 7, 2025 – NANCY, FRANCE

<table>
<tr><th>Time</th><th>Monday</th><th>Tuesday</th><th>Wednesday</th><th>Thursday</th><th>Friday</th></tr>
<tr><td>9:30 – 11:30</td><td rowspan="4"></td><td>Quantum Programming Languages</td><td>Concurrency</td><td>Quantum Linear Optics</td><td>Quantitative Types</td></tr>
<tr><td>11:30 – 12:00</td><td colspan="4">Coffee break</td></tr>
<tr><td>12:00 – 13:00</td><td>Realisability</td><td>Quantum Programming Languages</td><td>Quantitative Types</td><td>Industrial Session</td></tr>
<tr><td>13:00 – 13:30</td><td colspan="4" rowspan="2">Lunch break</td></tr>
<tr><td>13:30 – 14:30</td><td rowspan="2">Tutorial: Introduction to Quantum Computing</td></tr>
<tr><td>14:30 – 15:30</td><td>Realisability</td><td>Quantum Programming Languages</td><td>Quantitative Types</td><td rowspan="3">Quantum Linear Optics</td></tr>
<tr><td>15:30 – 16:00</td><td colspan="4">Coffee break</td></tr>
<tr><td>16:00 – 16:30</td><td rowspan="2">Tutorial: Introduction to ZX Calculus</td><td rowspan="2">Concurrency</td><td rowspan="2">Realisability</td><td rowspan="2">Quantum Programming Languages</td></tr>
<tr><td>16:30 – 18:00</td><td></td></tr>
</table>

cnrs

GDR Groupement de recherche IFM Informatique Fondamentale et ses Mathématiques

Diamond and Gold sponsors

QUANTINUUM

Loria Laboratoire lorrain de recherche en informatique et ses applications

Gilles Dowek (1966-2025)

Quantum Computing, an Introduction

Simon Perdrix

Inria, Mocqua/Loria

QComical

3 Nov 2025

https://qcomical2025.github.io/

Why a "quantum" processing of information?

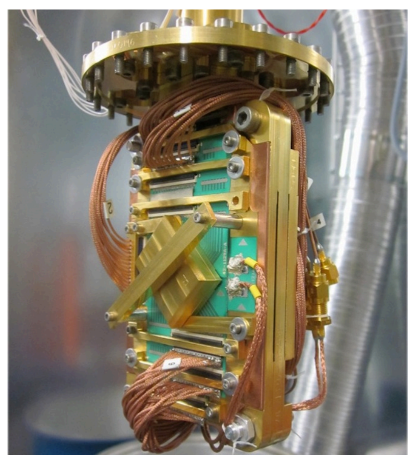
Some problems can be solved much more efficiently using quantum computers

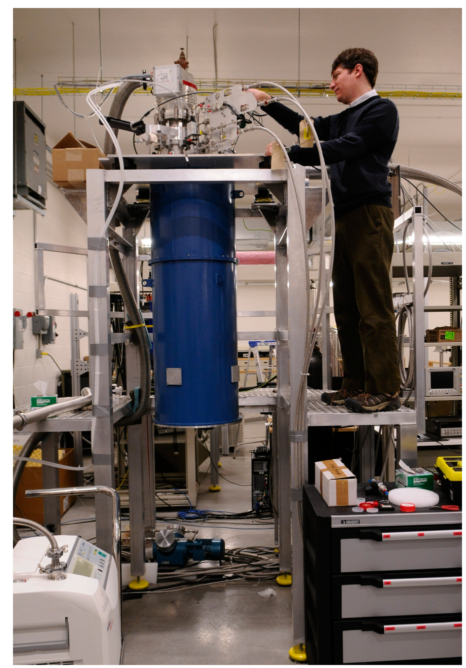
- Search [Grover’96]
- Solving Linear Systems [HHL’09]
- Factorisation [Shor’94]

Why a "quantum" processing of information?

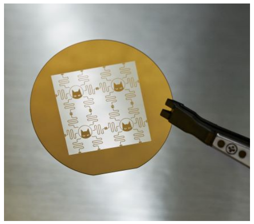
Some problems can be solved much more efficiently using quantum computers

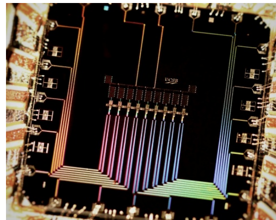
- Search [Grover'96]
- Solving Linear Systems [HHL'09]
- Factorisation [Shor'94]

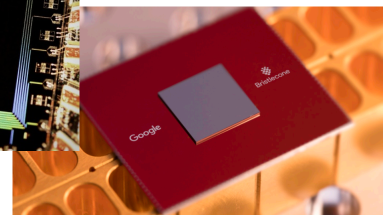


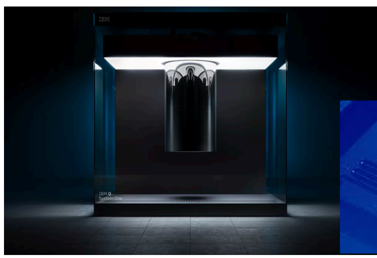


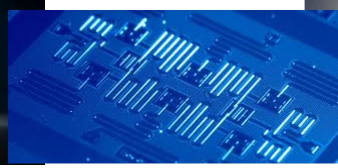
Various Quantum Technologies



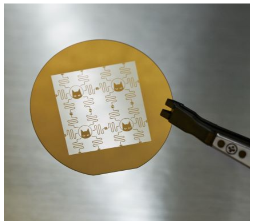


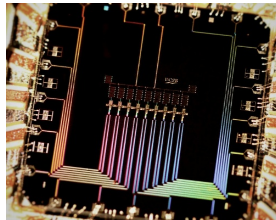


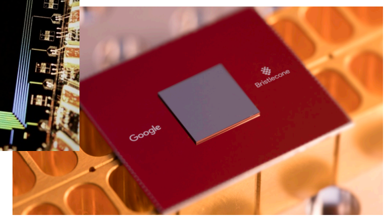


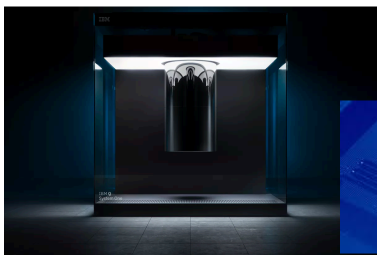


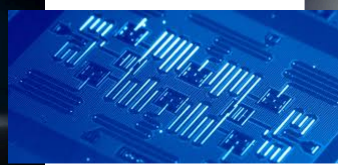
Various Quantum Technologies

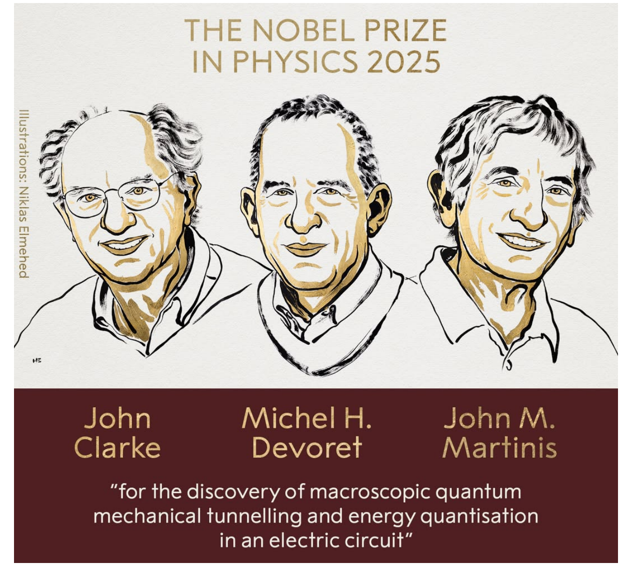




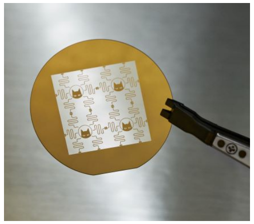


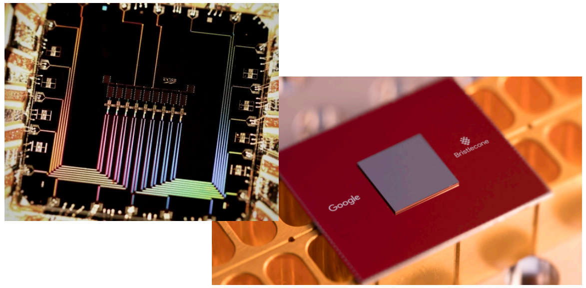


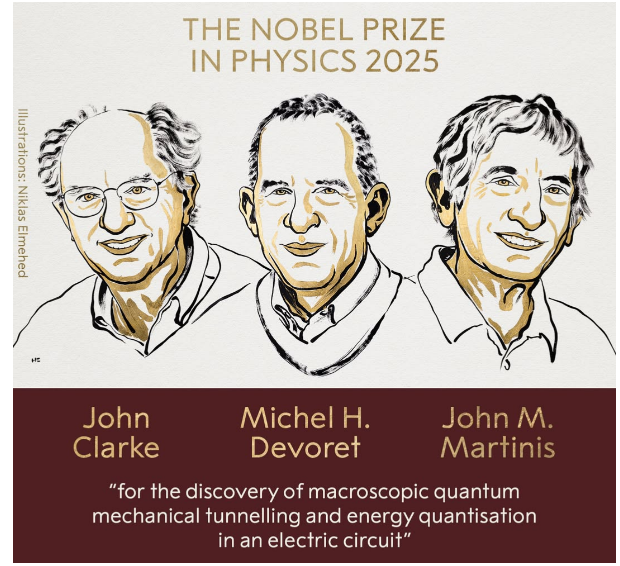


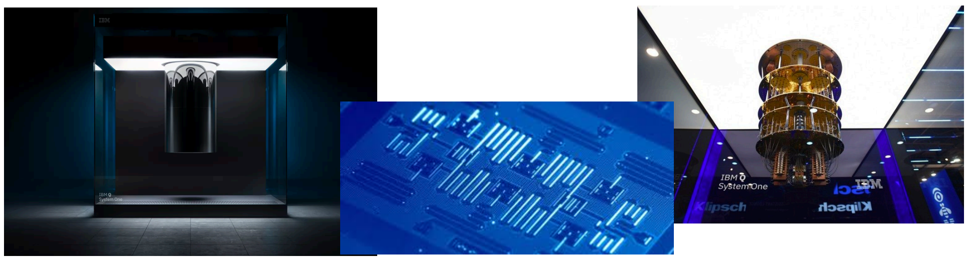


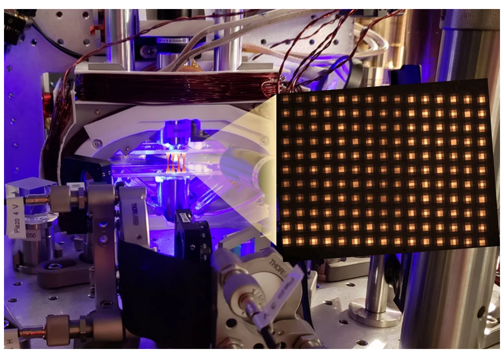
Various Quantum Technologies

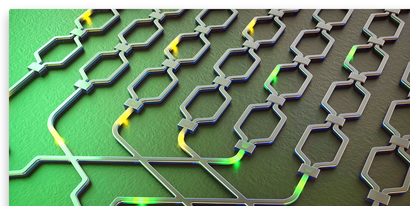


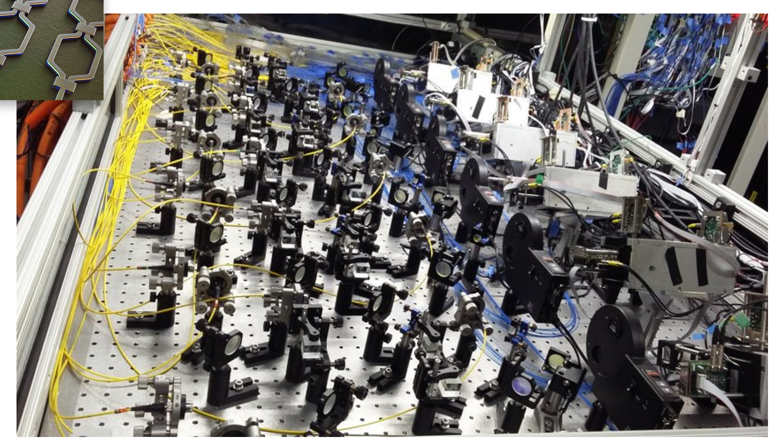


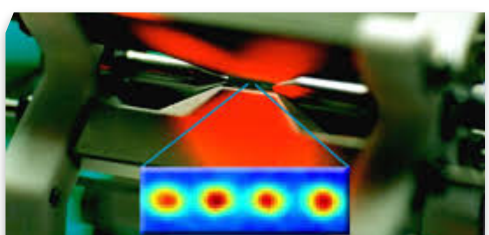




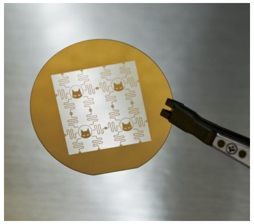


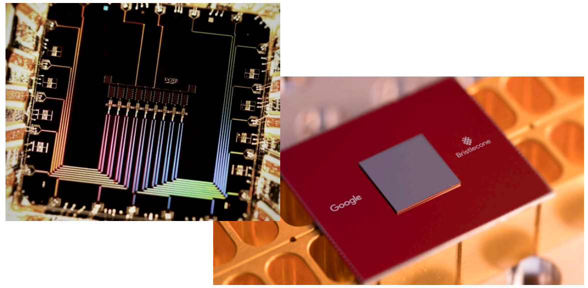


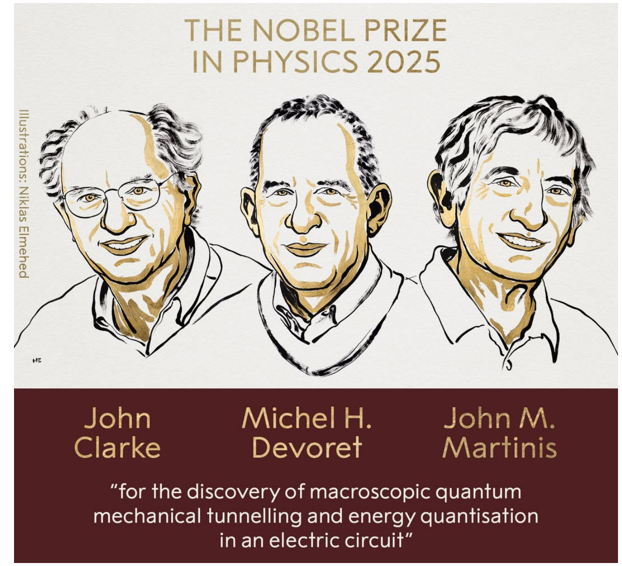


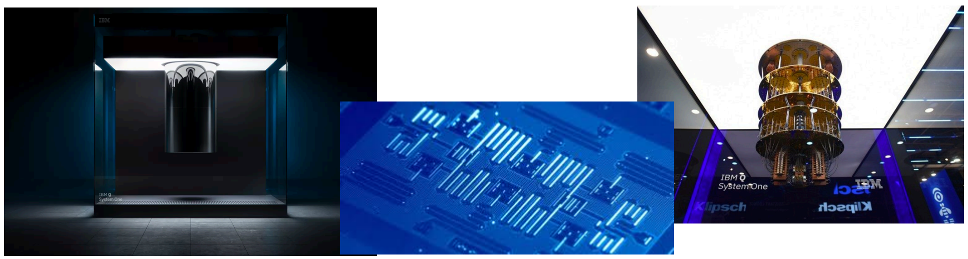


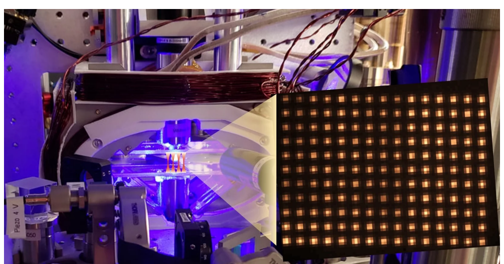
Various Quantum Technologies

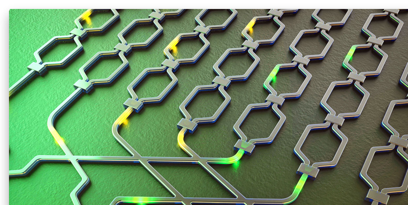


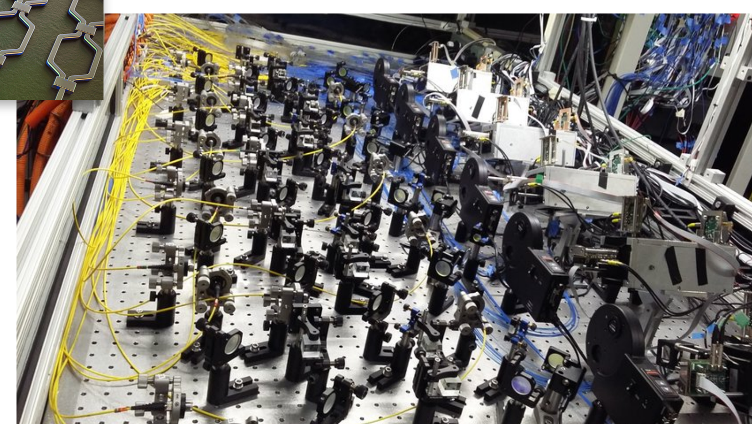


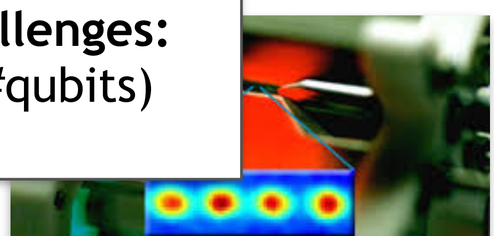






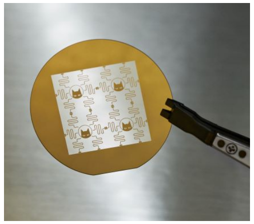


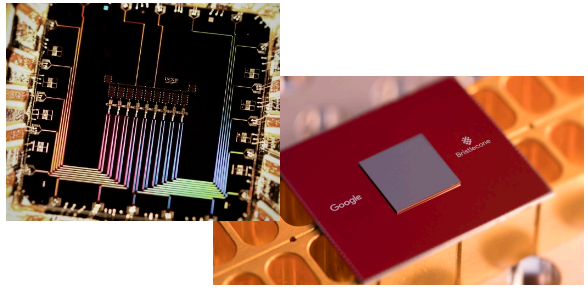


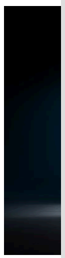


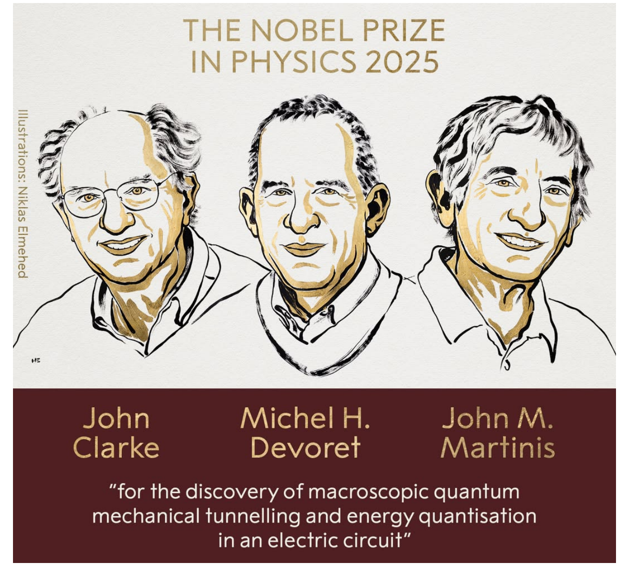
Main technological challenges:
- size of the memory (#qubits)
- quality of the qubits.

Various Quantum Technologies

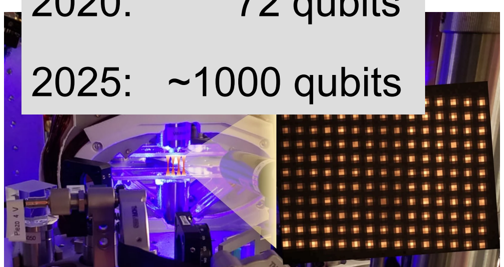


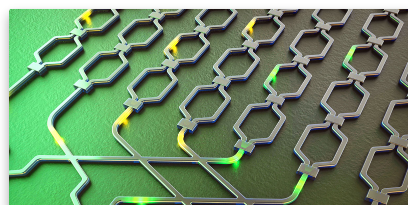


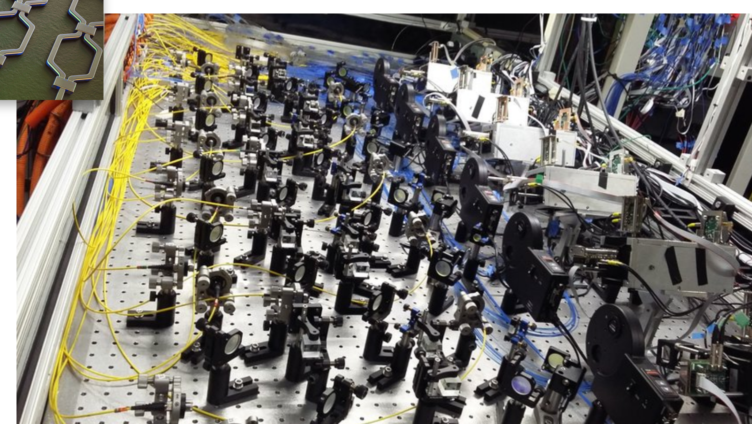


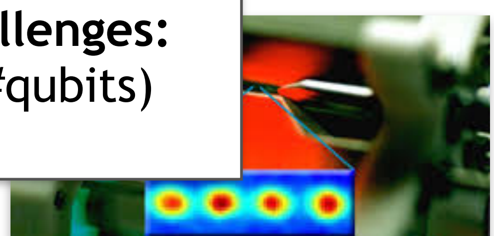


2002:	5 qubits
2008:	10 qubits
2015:	16 qubits
2018:	49 qubits
2020:	72 qubits
2025:	~1000 qubits





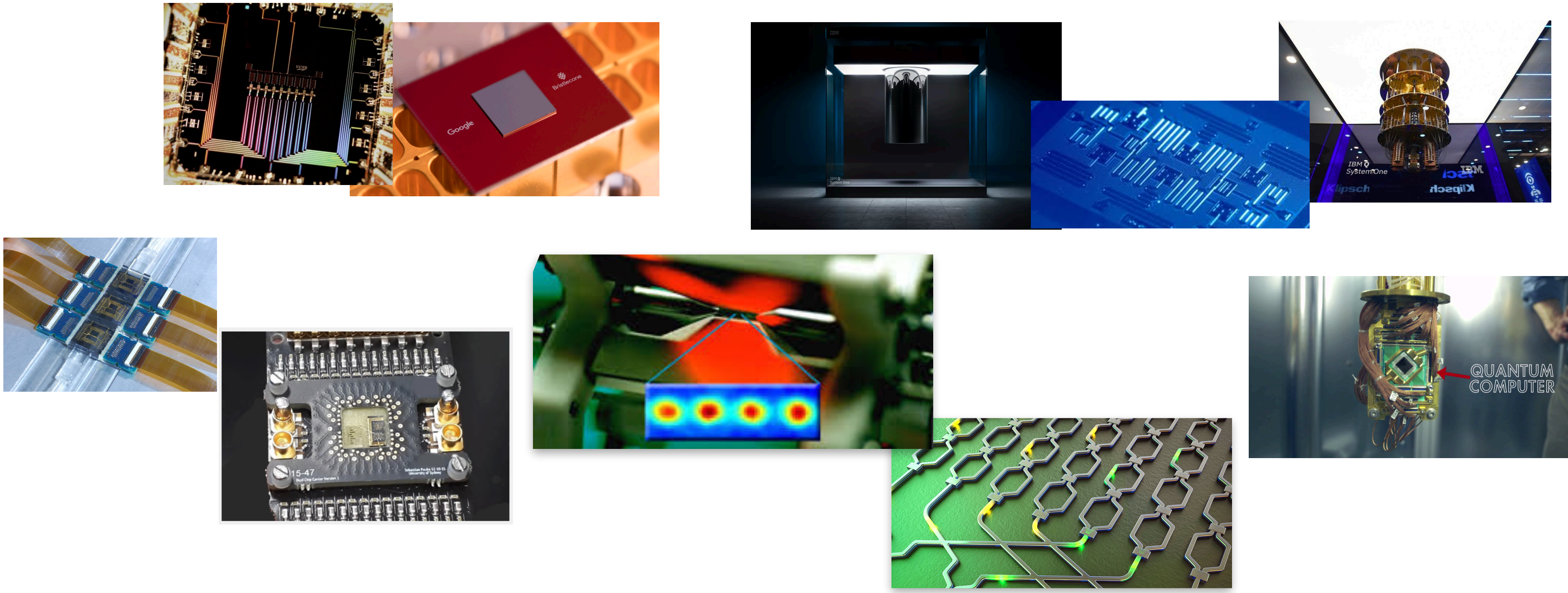




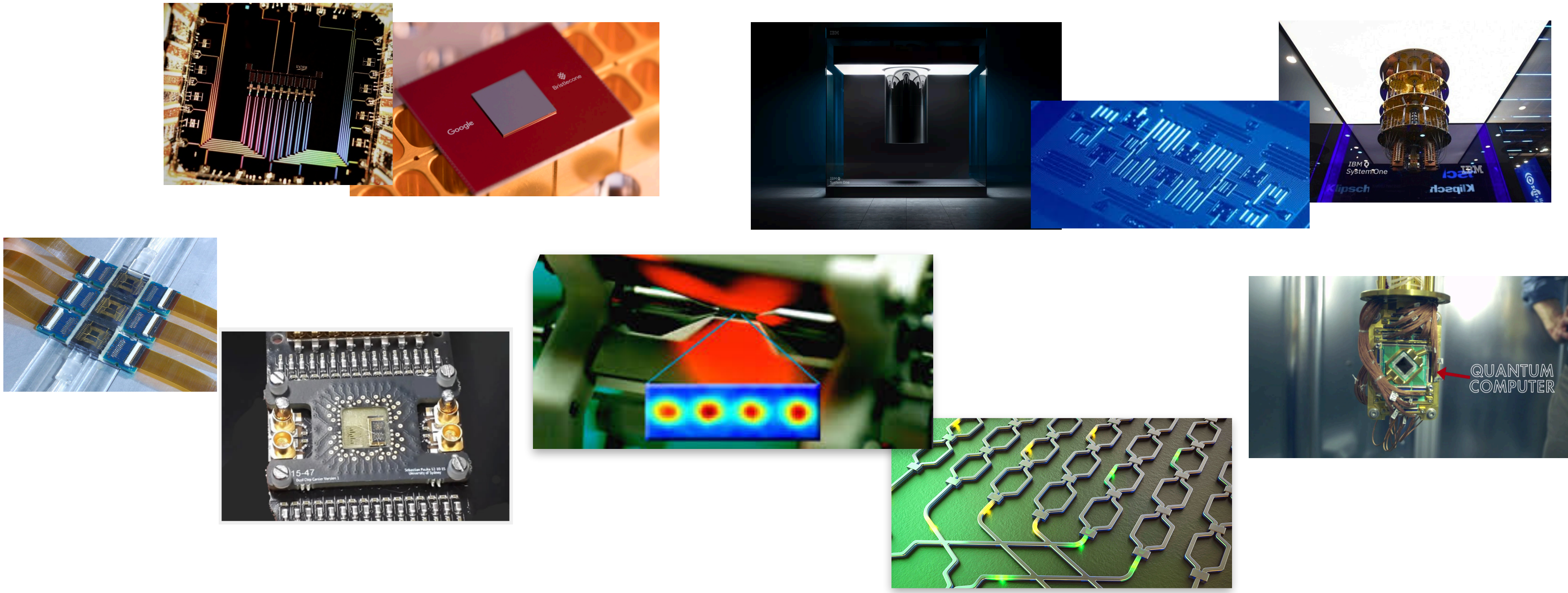
Main technological challenges:

- size of the memory (#qubits)
- quality of the qubits.

Noisy Intermediate-Scale Quantum (NISQ) devices

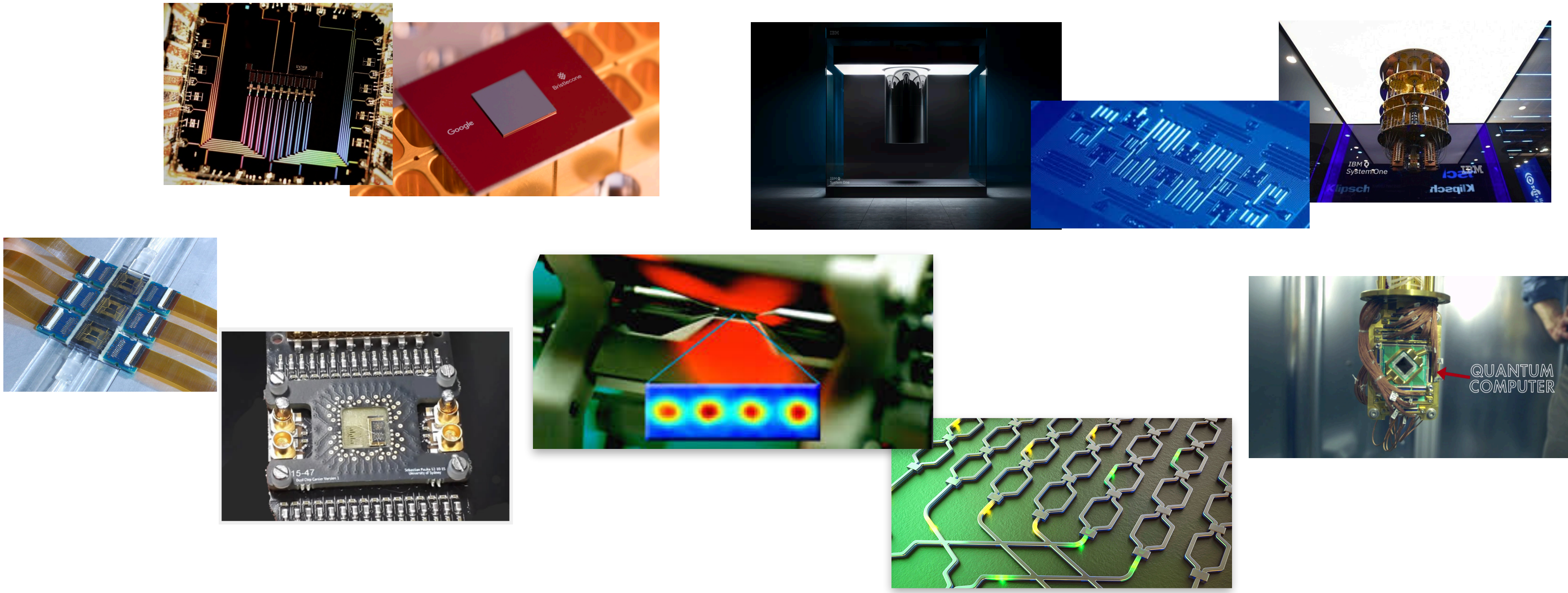


Noisy Intermediate-Scale Quantum (NISQ) devices



- Try to prove a theoretical separation classical / quantum computing
- Develop heuristics to try to outperform classical computers in practice

Noisy Intermediate-Scale Quantum (NISQ) devices



evidence of a

- Try to prove ~~a theoretical~~ separation classical / quantum computing
- Develop heuristics to try to outperform classical computers in practice

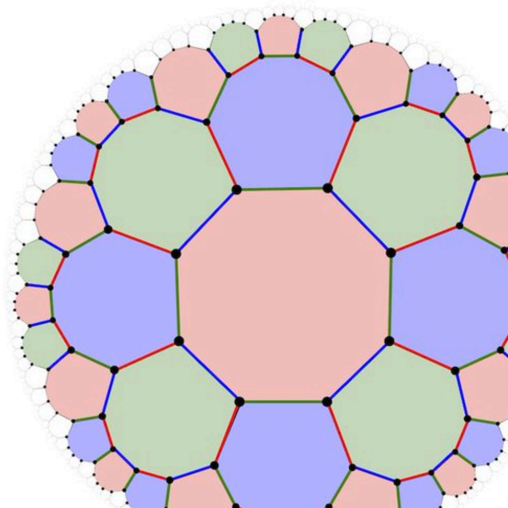
Towards Fault-Tolerant QC

- Quantum error correcting codes
- Threshold Theorem: correcting errors faster than they are created.

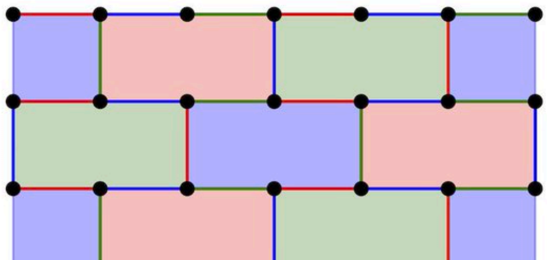
Physics: improve quality of the quantum memory

CS: develop codes with smaller threshold

Hyperbolic Floquet code



Toric honeycomb code



Factorisation of 2048-bit RSA integers

RSA-250 [edit]

RSA-250 has 250 decimal digits (829 bits), and was factored in February 2020 by Fabrice Boudot, Pierrick Gaudry, Aurore Guillevic, Nadia Heninger, Emmanuel Thomé, and Paul Zimmermann. The announcement of the factorization occurred on February 28, 2020.

```
RSA-250 = 2140324650240744961264423072839333563008614715144755017797754920881418023447
          1401366433455190958046796109928518724709145876873962619215573630474547705208
          0511905649310668769159001975940569345745223058932597669747168173806936489469
          9871578494975937497937
```

```
RSA-250 = 6413528947707158027879019017057738908482501474294344720811685963202453234463
          0238623598752668347708737661925585694639798853367
        × 3337202759497815655622601060535511422794076034476755466678452098702384172921
          0037080257448673296881877565718986258036932062711
```

The factorisation of RSA-250 utilised approximately 2700 CPU core-years, using a 2.1 GHz Intel Xeon Gold 6130 CPU as a reference. The computation was performed with the Number Field Sieve algorithm, using the open source CADO-NFS software.

(wikipedia, RSA factorisation challenges)

Factorisation of 2048-bit RSA integers

RSA-250 [edit]

RSA-250 has 250 decimal digits (829 bits), and was factored in February 2020 by Fabrice Boudot, Pierrick Gaudry, Aurore Guillevic, Nadia Heninger, Emmanuel Thomé, and Paul Zimmermann. The announcement of the factorization occurred on February 28, 2020.

[Submitted on 23 May 2019 (v1), last revised 13 Apr 2021 (this version, v3)]

How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits

Craig Gidney, Martin Ekerå

We significantly reduce the cost of factoring integers and computing discrete logarithms in finite fields on a quantum computer by combining techniques from Shor 1994, Griffiths-Niu 1996, Zalka 2006, Fowler 2012, Ekerå-Håstad 2017, Ekerå 2017, Ekerå 2018, Gidney-Fowler 2019, Gidney 2019. We estimate the approximate cost of our construction using plausible physical assumptions for large-scale superconducting qubit platforms: a planar grid of qubits with nearest-neighbor connectivity, a characteristic physical gate error rate of 10^{-3}, a surface code cycle time of 1 microsecond, and a reaction time of 10 microseconds. We account for factors

(wikipedia, RSA factorisation challenges)

Factorisation of 2048-bit RSA integers

RSA-250 [edit]

RSA-250 has 250 decimal digits (829 bits), and was factored in February 2020 by Fabrice Boudot, Pierrick Gaudry, Aurore Guillevic, Nadia Heninger, Emmanuel Thomé, and Paul Zimmermann. The announcement of the factorization occurred on February 28, 2020.

[Submitted on 23 May 2019 (v1), last revised 13 Apr 2021 (this version, v3)]

How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits

Craig Gidney, Martin Ekerå

We significantly reduce the cost of factoring integers and computing discrete logarithms in finite fields on a quantum computer by combining techniques from Shor 1994, Griffiths-Niu 1996, Zalka 2006, Fowler 2012, Ekerå-Håstad 2017, Ekerå 2017, Ekerå 2018, Gidney-Fowler 2019, Gidney 2019. We estimate the approximate cost of our construction using plausible physical assumptions for large-scale superconducting qubit platforms: a planar grid of qubits with nearest-neighbor connectivity, a characteristic physical gate error rate of 10^{-3}, a surface code cycle time of 1 microsecond, and a reaction time of 10 microseconds. We account for factors

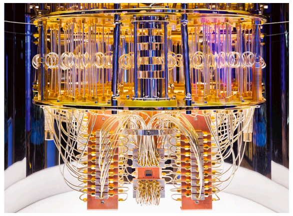
[Submitted on 21 May 2025]

How to factor 2048 bit RSA integers with less than a million noisy qubits

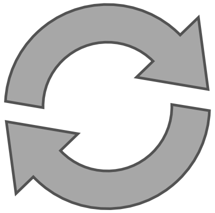
Craig Gidney

Planning the transition to quantum-safe cryptosystems requires understanding the cost of quantum attacks on vulnerable cryptosystems. In Gidney+Ekerå 2019, I co-published an estimate stating that 2048 bit RSA integers could be factored in eight hours by a quantum computer with 20 million noisy qubits. In this paper, I substantially reduce the number of qubits required. I estimate that a 2048 bit RSA integer could be factored in less than a week by a quantum computer with less than a million noisy qubits. I make the same assumptions as in 2019: a square grid of qubits with nearest neighbor connections, a uniform gate error rate of 0.1%, a surface code cycle time of 1 microsecond, and a control system reaction time of 10 microseconds. The qubit count reduction comes mainly from using approximate residue arithmetic (Chevignard+Fouque+Schrottenloher 2024), from storing idle logical qubits with yoked surface codes (Gidney+Newman+Brooks+Jones 2023), and from allocating

Current challenges in Quantum Computing



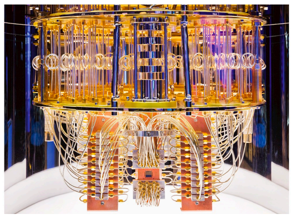
Quantum
Technologies



Quantum
Software

Current challenges in Quantum Computing

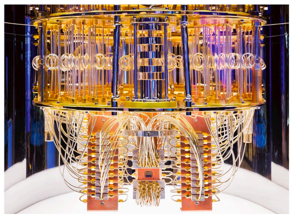
Applications /
Quantum Algorithms



Quantum
Technologies

Quantum
Software

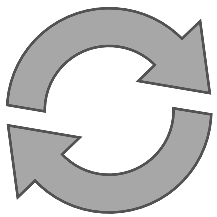
Current challenges in Quantum Computing



Applications /
Quantum Algorithms

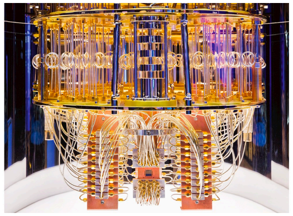
Environment / Languages

Quantum
Technologies

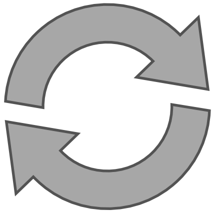


Quantum
Software

Current challenges in Quantum Computing



Quantum Technologies



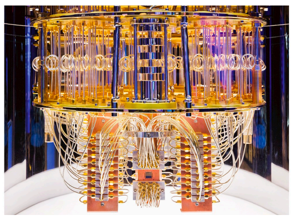
Quantum Software

Applications / Quantum Algorithms

Environment / Languages

Models of Computation

Current challenges in Quantum Computing



Quantum Technologies

Quantum Software

Applications / Quantum Algorithms

Environment / Languages

Models of Computation

Error correcting codes

Outline

Challenges in Quantum computing

Postulates i.e. standard quantum computational model.

1st Quantum Algorithm

Reasoning on Quantum Circuits

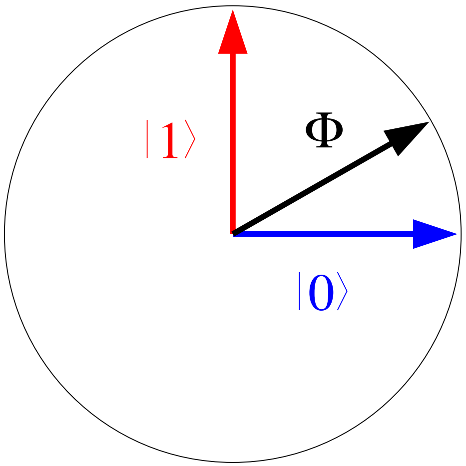
Grover

Quantum states

- Classical bit: $b \in \{0, 1\}$
- Quantum bit (**qubit**): $\mathbf{\Phi} \in \mathbb{C}^2$,

$$\mathbf{\Phi} = \alpha |0\rangle + \beta |1\rangle$$

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



Examples:

$$|0\rangle$$

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

Register of qubits

Definition. The state of a n-qubit register is a unit vector of $\mathbb{C}^{2^n}$.

$$\mathbf{\Phi} = \sum_{x \in \{0,1\}^n} \alpha_x |x\rangle \text{ with } ||\mathbf{\Phi}||^2 = \sum_{x \in \{0,1\}^n} |\alpha_x|^2 = 1$$

Examples:

$$\frac{1}{\sqrt{2}}(|00\rangle - |01\rangle)$$

$$\frac{1}{\sqrt{3}}(|00\rangle + i\,|01\rangle + |11\rangle)$$

$$\frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$$

Postulate 2: composed system

Definition. Let $\mathbf{\Phi_1}$ be a n-qubit state and $\mathbf{\Phi_2}$ be a m-qubit state, the $(n+m)$-qubit state of the composed system is

$$\mathbf{\Phi} = \mathbf{\Phi_1} \otimes \mathbf{\Phi_2}$$

where $\cdot \otimes \cdot$ is bilinear and $\forall x \in \{0,1\}^n$, $\forall y \in \{0,1\}^m$, $|x\rangle \otimes |y\rangle = |xy\rangle$.

Examples:

1. $|0\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}} = \frac{|0\rangle \otimes |0\rangle - |0\rangle \otimes |1\rangle}{\sqrt{2}} = \frac{|00\rangle - |01\rangle}{\sqrt{2}}$
2. $\frac{|01\rangle + |11\rangle}{\sqrt{2}} \quad = \quad ? \otimes ?$
3. $\frac{|00\rangle + |11\rangle}{\sqrt{2}} \quad = \quad ? \otimes ?$

Postulate 2: composed system

Definition. Let $\mathbf{\Phi_1}$ be a n-qubit state and $\mathbf{\Phi_2}$ be a m-qubit state, the $(n+m)$-qubit state of the composed system is

$$\mathbf{\Phi} = \mathbf{\Phi_1} \otimes \mathbf{\Phi_2}$$

where $\cdot \otimes \cdot$ is bilinear and $\forall x \in \{0,1\}^n$, $\forall y \in \{0,1\}^m$, $|x\rangle \otimes |y\rangle = |xy\rangle$.

Examples:

1. $|0\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}} = \frac{|0\rangle \otimes |0\rangle - |0\rangle \otimes |1\rangle}{\sqrt{2}} = \frac{|00\rangle - |01\rangle}{\sqrt{2}}$
2. $\frac{|01\rangle + |11\rangle}{\sqrt{2}} = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes |1\rangle$
3. $\frac{|00\rangle + |11\rangle}{\sqrt{2}} = ? \otimes ?$

Postulate 2: composed system

Definition. Let $\mathbf{\Phi_1}$ be a n-qubit state and $\mathbf{\Phi_2}$ be a m-qubit state, the $(n+m)$-qubit state of the composed system is

$$\mathbf{\Phi} = \mathbf{\Phi_1} \otimes \mathbf{\Phi_2}$$

where $\cdot \otimes \cdot$ is bilinear and $\forall x \in \{0,1\}^n$, $\forall y \in \{0,1\}^m$, $|x\rangle \otimes |y\rangle = |xy\rangle$.

Examples:

1. $|0\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}} = \frac{|0\rangle\otimes|0\rangle - |0\rangle\otimes|1\rangle}{\sqrt{2}} = \frac{|00\rangle - |01\rangle}{\sqrt{2}}$
2. $\frac{|01\rangle + |11\rangle}{\sqrt{2}} = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes |1\rangle$
3. $$\begin{aligned} \frac{|00\rangle + |11\rangle}{\sqrt{2}} &= (a\,|0\rangle + b\,|1\rangle) \otimes (c\,|0\rangle + d\,|1\rangle) \\ &= ac\,|00\rangle + ad\,|01\rangle + bc\,|10\rangle + bd\,|11\rangle \end{aligned}$$

Postulate 2: composed system

Definition. Let $\mathbf{\Phi_1}$ be a n-qubit state and $\mathbf{\Phi_2}$ be a m-qubit state, the $(n+m)$-qubit state of the composed system is

$$\mathbf{\Phi} = \mathbf{\Phi_1} \otimes \mathbf{\Phi_2}$$

where $\cdot \otimes \cdot$ is bilinear and $\forall x \in \{0,1\}^n$, $\forall y \in \{0,1\}^m$, $|x\rangle \otimes |y\rangle = |xy\rangle$.

Examples:

1. $|0\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}} = \frac{|0\rangle \otimes |0\rangle - |0\rangle \otimes |1\rangle}{\sqrt{2}} = \frac{|00\rangle - |01\rangle}{\sqrt{2}}$
2. $\frac{|01\rangle + |11\rangle}{\sqrt{2}} = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes |1\rangle$
3. $$\begin{aligned} \frac{|00\rangle + |11\rangle}{\sqrt{2}} &\neq (a\,|0\rangle + b\,|1\rangle) \otimes (c\,|0\rangle + d\,|1\rangle) \\ &= ac\,|00\rangle + ad\,|01\rangle + bc\,|10\rangle + bd\,|11\rangle \\ &\implies ad = 0 \implies ac = 0 \text{ or } bd = 0 \quad \text{impossible} \end{aligned}$$

Postulate 2: composed system

Definition. Let $\mathbf{\Phi_1}$ be a n-qubit state and $\mathbf{\Phi_2}$ be a m-qubit state, the $(n+m)$-qubit state of the composed system is

$$\mathbf{\Phi} = \mathbf{\Phi_1} \otimes \mathbf{\Phi_2}$$

where $\cdot \otimes \cdot$ is bilinear and $\forall x \in \{0,1\}^n$, $\forall y \in \{0,1\}^m$, $|x\rangle \otimes |y\rangle = |xy\rangle$.

Examples:

1. $|0\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}} = \frac{|0\rangle \otimes |0\rangle - |0\rangle \otimes |1\rangle}{\sqrt{2}} = \frac{|00\rangle - |01\rangle}{\sqrt{2}}$
2. $\frac{|01\rangle + |11\rangle}{\sqrt{2}} = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes |1\rangle$
3. $$\begin{aligned} \frac{|00\rangle + |11\rangle}{\sqrt{2}} &\neq (a\,|0\rangle + b\,|1\rangle) \otimes (c\,|0\rangle + d\,|1\rangle) \\ &= ac\,|00\rangle + ad\,|01\rangle + bc\,|10\rangle + bd\,|11\rangle \\ &\implies ad = 0 \implies ac = 0 \text{ or } bd = 0 \quad \text{impossible} \end{aligned}$$

$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ **is an entangled state.**

Postulate 2: composed system

Definition. Let $\mathbf{\Phi_1}$ be a n-qubit state and $\mathbf{\Phi_2}$ be a m-qubit state, the $(n+m)$-qubit state of the composed system is

$$\mathbf{\Phi} = \mathbf{\Phi_1} \otimes \mathbf{\Phi_2}$$

where $\cdot \otimes \cdot$ is bilinear and $\forall x \in \{0,1\}^n$, $\forall y \in \{0,1\}^m$, $|x\rangle \otimes |y\rangle = |xy\rangle$.

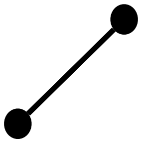
Examples:

1. $|0\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}} = \frac{|0\rangle \otimes |0\rangle - |0\rangle \otimes |1\rangle}{\sqrt{2}} = \frac{|00\rangle - |01\rangle}{\sqrt{2}}$
2. $\frac{|01\rangle + |11\rangle}{\sqrt{2}} = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes |1\rangle$
3. $$\begin{aligned} \frac{|00\rangle + |11\rangle}{\sqrt{2}} &\neq (a\,|0\rangle + b\,|1\rangle) \otimes (c\,|0\rangle + d\,|1\rangle) \\ &= ac\,|00\rangle + ad\,|01\rangle + bc\,|10\rangle + bd\,|11\rangle \\ &\implies ad = 0 \implies ac = 0 \text{ or } bd = 0 \quad \text{impossible} \end{aligned}$$

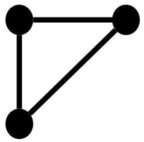
$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ **is an entangled state.**

Representing Entanglement

$$\frac{|00\rangle + |11\rangle}{\sqrt{2}}$$



$$\frac{|000\rangle + |111\rangle}{\sqrt{2}}$$



Representing Entanglement with Graph states

Representing Entanglement with Graph states

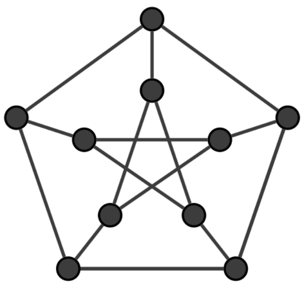
Def. Graph states:

$$G \quad \mapsto \quad |G\rangle = \frac{1}{\sqrt{2}^{|V|}} \sum_{x \in 2^V} (-1)^{|G[x]|} |x\rangle$$

Representing Entanglement with Graph states

Def. Graph states: $G \quad \mapsto \quad |G\rangle = \frac{1}{\sqrt{2}^{|V|}} \sum_{x \in 2^V} (-1)^{|G[x]|} |x\rangle$

- compact representation



$\mapsto \quad \cdots$

Representing Entanglement with Graph states

Def. Graph states:

$$G \quad \mapsto \quad |G\rangle = \frac{1}{\sqrt{2}^{|V|}} \sum_{x \in 2^V} (-1)^{|G[x]|} |x\rangle$$

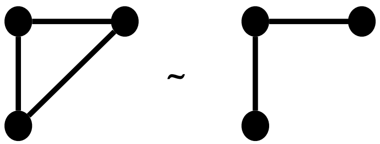
- compact representation

Representing Entanglement with Graph states

Def. Graph states:

$$G \quad \mapsto \quad |G\rangle = \frac{1}{\sqrt{2}^{|V|}} \sum_{x \in 2^V} (-1)^{|G[x]|} |x\rangle$$

- compact representation
- representation of entanglement is not unique

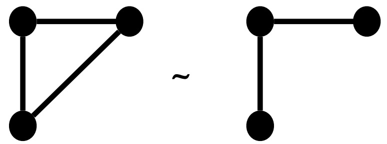


same entanglement

Representing Entanglement with Graph states

Def. Graph states: $$G \quad \mapsto \quad |G\rangle = \frac{1}{\sqrt{2}^{|V|}} \sum_{x \in 2^V} (-1)^{|G[x]|} |x\rangle$$

- compact representation
- representation of entanglement is not unique
- Local complementation preserves entanglement:

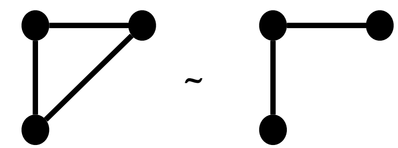


same entanglement

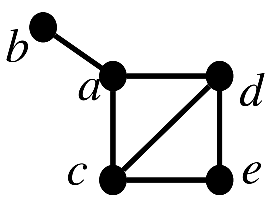
Representing Entanglement with Graph states

Def. Graph states: $G \quad \mapsto \quad |G\rangle = \frac{1}{\sqrt{2}^{|V|}} \sum_{x \in 2^V} (-1)^{|G[x]|} |x\rangle$

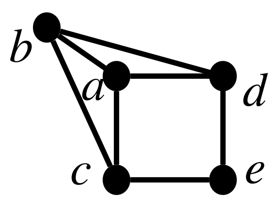
- compact representation
- representation of entanglement is not unique
- Local complementation preserves entanglement:



same entanglement



G

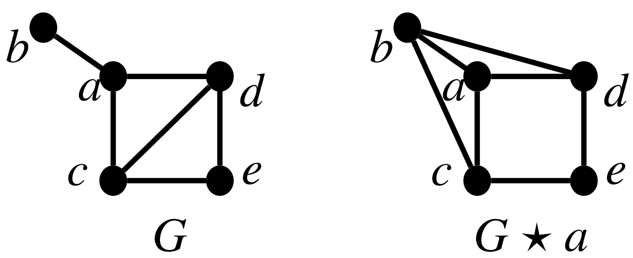


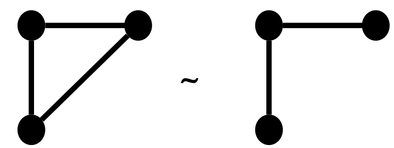
$G \star a$

Representing Entanglement with Graph states

Def. Graph states: $G \quad \mapsto \quad |G\rangle = \frac{1}{\sqrt{2}^{|V|}} \sum_{x \in 2^V} (-1)^{|G[x]|} |x\rangle$

- compact representation
- representation of entanglement is not unique
- Local complementation preserves entanglement:





same entanglement

THM[1]. Two graphs represent the same entanglement iff the can be transformed into each other by means of generalised local complementation

[1] N. Claudet, S. Perdrix, QIP25, ICALP'25

Measurement

$\alpha |0\rangle + \beta |1\rangle \xrightarrow{|\alpha|^2} |0\rangle$ *with classical outcome* 0

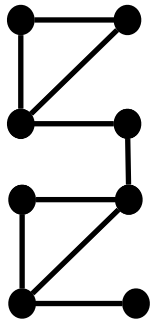
$\alpha |0\rangle + \beta |1\rangle \xrightarrow{|\beta|^2} |1\rangle$ *with classical outcome* 1

Measurement is **probabilistic** and **irreversible**.

Measure $\Longrightarrow$ Interaction $\Longrightarrow$ Transformation

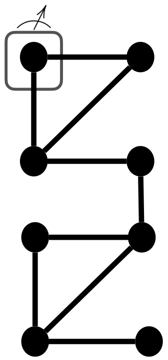
Measurement based quantum computation

MBQC [Briegel, Raussendorf 2001] Universal model of Quantum computing.



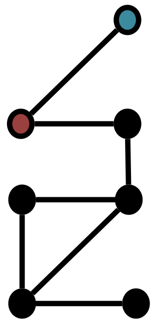
Measurement based quantum computation

MBQC [Briegel, Raussendorf 2001] Universal model of Quantum computing.



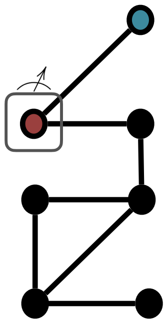
Measurement based quantum computation

MBQC [Briegel, Raussendorf 2001] Universal model of Quantum computing.



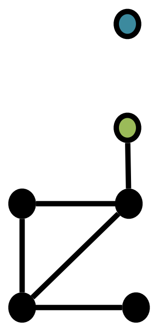
Measurement based quantum computation

MBQC [Briegel, Raussendorf 2001] Universal model of Quantum computing.



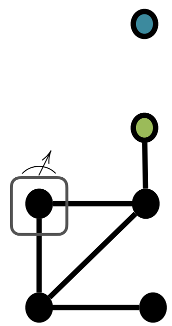
Measurement based quantum computation

MBQC [Briegel, Raussendorf 2001] Universal model of Quantum computing.



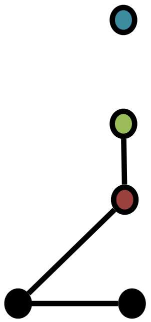
Measurement based quantum computation

MBQC [Briegel, Raussendorf 2001] Universal model of Quantum computing.



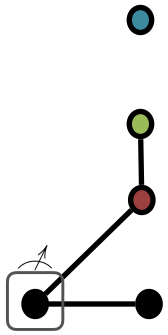
Measurement based quantum computation

MBQC [Briegel, Raussendorf 2001] Universal model of Quantum computing.



Measurement based quantum computation

MBQC [Briegel, Raussendorf 2001] Universal model of Quantum computing.



Measurement based quantum computation

MBQC [Briegel, Raussendorf 2001] Universal model of Quantum computing.

Measurement

$$\alpha|0\rangle + \beta|1\rangle \xrightarrow{|\alpha|^2} |0\rangle \text{ with classical outcome } 0$$

$$\alpha|0\rangle + \beta|1\rangle \xrightarrow{|\beta|^2} |1\rangle \text{ with classical outcome } 1$$

Measurement is **probabilistic** and **irreversible**.

Measure $\Longrightarrow$ Interaction $\Longrightarrow$ Transformation

Closed Systems: a Unitary Evolution

Definition. An isolated system evolves

- linearly i.e., $U(\alpha\Phi + \beta\Psi) = \alpha U(\Phi) + \beta U(\Psi)$
- preserving the normalisation condition i.e., $||U(\Phi)|| = ||\Phi||$

Example:

$$\begin{array}{cccccc} H & : & |0\rangle & \mapsto & \frac{|0\rangle+|1\rangle}{\sqrt{2}} \\ & & |1\rangle & \mapsto & \frac{|0\rangle-|1\rangle}{\sqrt{2}} \end{array}$$

$H(H(|0\rangle)) =$

Closed Systems: a Unitary Evolution

Definition. An isolated system evolves

- linearly i.e., $U(\alpha\Phi + \beta\Psi) = \alpha U(\Phi) + \beta U(\Psi)$
- preserving the normalisation condition i.e., $||U(\Phi)|| = ||\Phi||$

Example:

$$\begin{array}{rcccc} H & : & |0\rangle & \mapsto & \frac{|0\rangle+|1\rangle}{\sqrt{2}} \\ & & |1\rangle & \mapsto & \frac{|0\rangle-|1\rangle}{\sqrt{2}} \end{array}$$

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

Closed Systems: a Unitary Evolution

Definition. An isolated system evolves

- linearly i.e., $U(\alpha\Phi + \beta\Psi) = \alpha U(\Phi) + \beta U(\Psi)$
- preserving the normalisation condition i.e., $||U(\Phi)|| = ||\Phi||$

Example:

$$\begin{array}{cccc} H & : & |0\rangle & \mapsto & \frac{|0\rangle+|1\rangle}{\sqrt{2}} \\ & & |1\rangle & \mapsto & \frac{|0\rangle-|1\rangle}{\sqrt{2}} \end{array}$$

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

Closed Systems: a Unitary Evolution

Definition. An isolated system evolves

- linearly i.e., $U(\alpha\Phi + \beta\Psi) = \alpha U(\Phi) + \beta U(\Psi)$
- preserving the normalisation condition i.e., $||U(\Phi)|| = ||\Phi||$

Example:

$$\begin{array}{llll} H & : & |0\rangle & \mapsto \frac{|0\rangle+|1\rangle}{\sqrt{2}} \\ & & |1\rangle & \mapsto \frac{|0\rangle-|1\rangle}{\sqrt{2}} \end{array}$$

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

Closed Systems: a Unitary Evolution

Definition. An isolated system evolves

- linearly i.e., $U(\alpha\Phi + \beta\Psi) = \alpha U(\Phi) + \beta U(\Psi)$
- preserving the normalisation condition i.e., $||U(\Phi)|| = ||\Phi||$

Example:

$$\begin{array}{rcrcl} H & : & |0\rangle & \mapsto & \frac{|0\rangle + |1\rangle}{\sqrt{2}} \\ & & |1\rangle & \mapsto & \frac{|0\rangle - |1\rangle}{\sqrt{2}} \end{array}$$

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

Closed Systems: a Unitary Evolution

Definition. An isolated system evolves

- linearly i.e., $U(\alpha\Phi + \beta\Psi) = \alpha U(\Phi) + \beta U(\Psi)$
- preserving the normalisation condition i.e., $||U(\Phi)|| = ||\Phi||$

Example:

$$
\begin{array}{cccc}
H & : & |0\rangle & \mapsto & \frac{|0\rangle+|1\rangle}{\sqrt{2}} \\
 & & |1\rangle & \mapsto & \frac{|0\rangle-|1\rangle}{\sqrt{2}}
\end{array}
$$

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

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

Closed Systems: a Unitary Evolution

Definition. An isolated system evolves

- linearly i.e., $U(\alpha\Phi + \beta\Psi) = \alpha U(\Phi) + \beta U(\Psi)$
- preserving the normalisation condition i.e., $||U(\Phi)|| = ||\Phi||$

Example:

$$\forall x \in \{0,1\}, \quad H|x\rangle = \frac{|0\rangle + (-1)^x|1\rangle}{\sqrt{2}}$$

Closed Systems: a Unitary Evolution

Definition. An isolated system evolves

- linearly i.e., $U(\alpha\Phi + \beta\Psi) = \alpha U(\Phi) + \beta U(\Psi)$
- preserving the normalisation condition i.e., $||U(\Phi)|| = ||\Phi||$

Example:

$$\forall x \in \{0,1\}, \quad H|x\rangle = \frac{|0\rangle + (-1)^x|1\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}} \sum_{y \in \{0,1\}} (-1)^{xy}|y\rangle$$

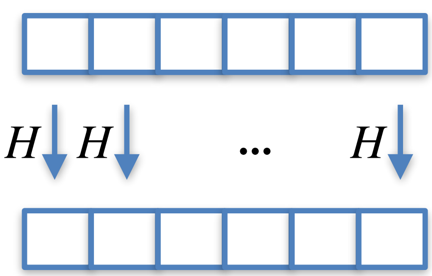
Closed Systems: a Unitary Evolution

Definition. An isolated system evolves

- linearly i.e., $U(\alpha\Phi + \beta\Psi) = \alpha U(\Phi) + \beta U(\Psi)$
- preserving the normalisation condition i.e., $||U(\Phi)|| = ||\Phi||$

Example:

$$\forall x \in \{0,1\}, \quad H|x\rangle = \frac{|0\rangle + (-1)^x|1\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}} \sum_{y\in\{0,1\}} (-1)^{xy}|y\rangle$$



$$\forall x \in \{0,1\}^n, \quad H_n|x\rangle = \frac{1}{\sqrt{2^n}} \sum_{y\in\{0,1\}^n} (-1)^{x\bullet y}|y\rangle$$

with $$x \bullet y = \sum_{i=1}^{n} x_i y_i \bmod 2$$

Outline

Challenges in Quantum computing

Postulates

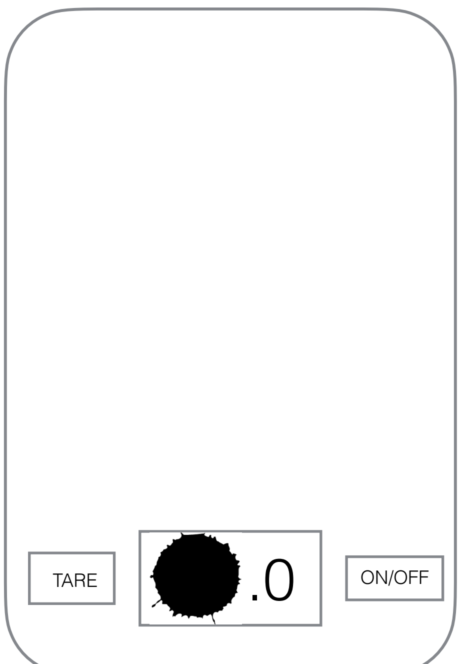
1st Quantum Algorithm: Detecting fake coins with a quantum scale

Reasoning on Quantum Circuits

Grover

Detecting fake coins

A true coin weighs 8g,
a fake 7.5g.



Detecting fake coins

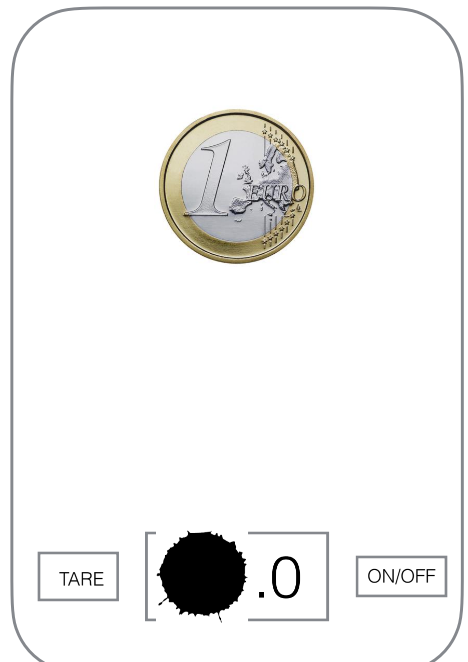
A true coin weighs 8g,
a fake 7.5g.

Detecting fake coins

A true coin weighs 8g,
a fake 7.5g.

Detecting fake coins

A true coin weighs 8g,
a fake 7.5g.



Detecting fake coins

A true coin weighs 8g,
a fake 7.5g.

Detecting fake coins

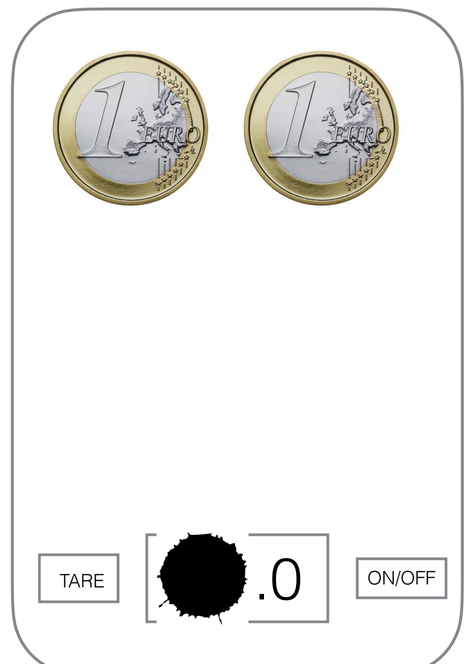
A true coin weighs 8g,
a fake 7.5g.

Detecting fake coins

A true coin weighs 8g,
a fake 7.5g.

Detecting fake coins

A true coin weighs 8g,
a fake 7.5g.



Detecting fake coins

A true coin weighs 8g,
a fake 7.5g.

Detecting fake coins

A true coin weighs 8g,
a fake 7.5g.

Detecting fake coins

A true coin weighs 8g,
a fake 7.5g.

Detecting fake coins

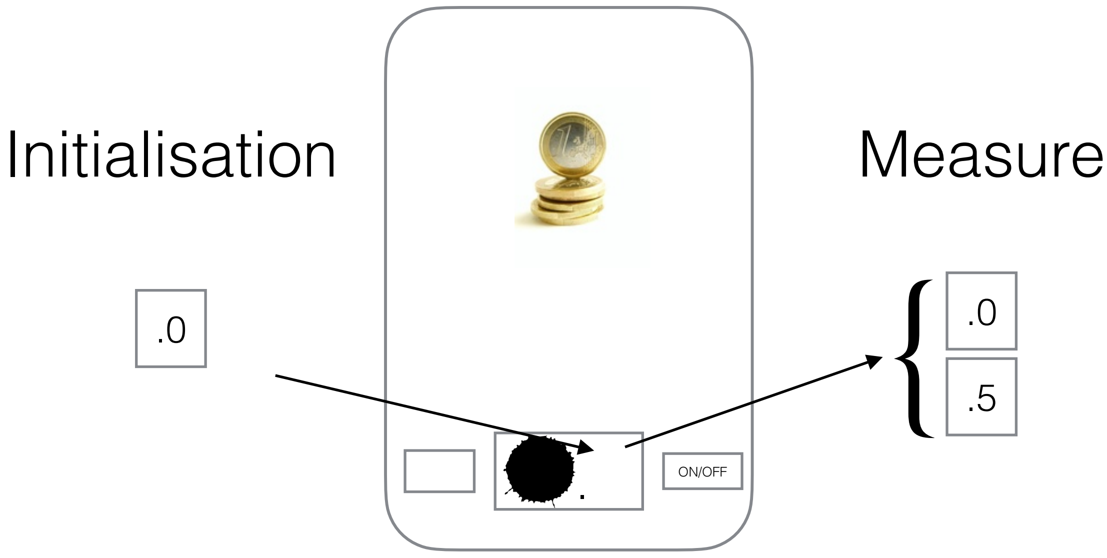
A true coin weighs 8g,
a fake 7.5g.

odd number of fake coins

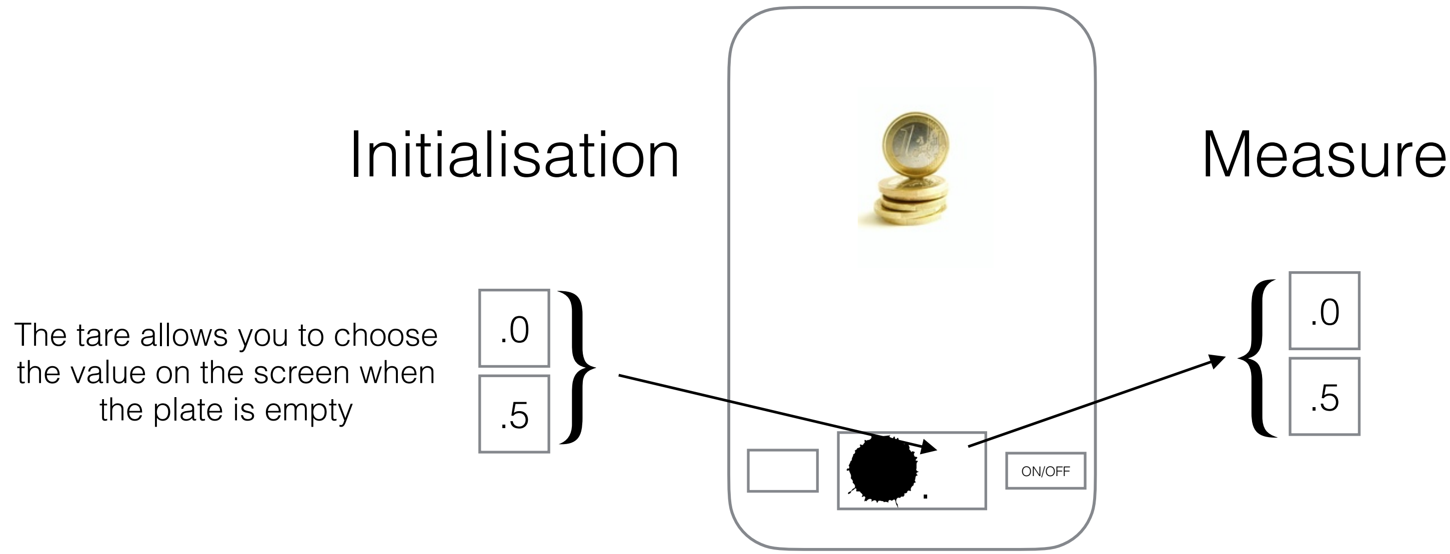
Detecting fake coins

A true coin weighs 8g,
a fake 7.5g.

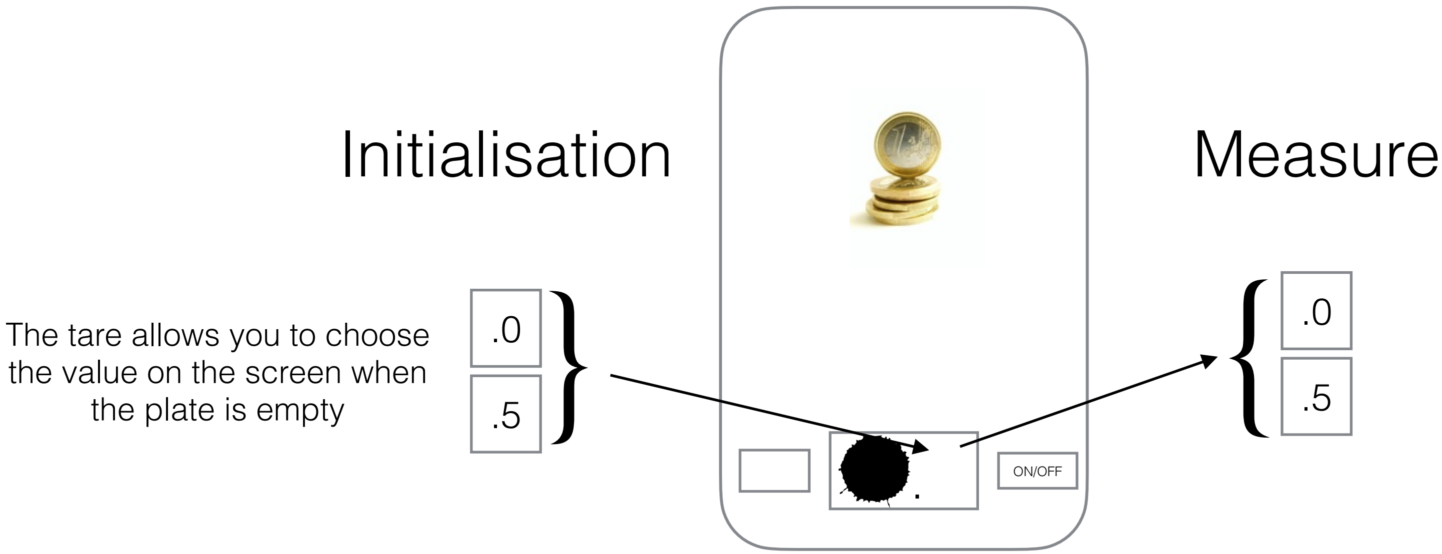
Digression : Tare weight



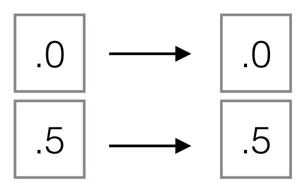
Digression : Tare weight



Digression : Tare weight

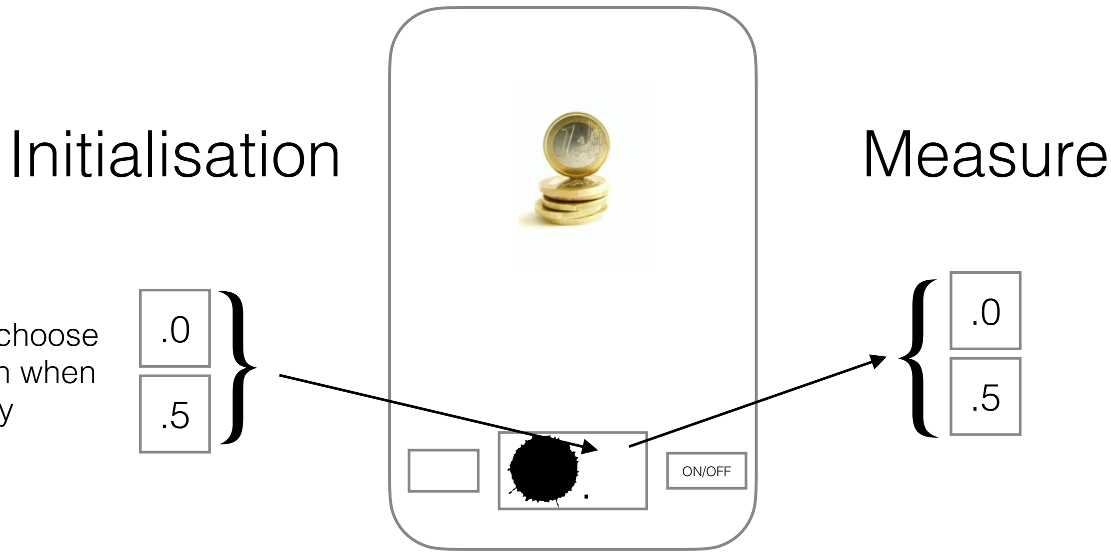


- **even** number of fake coins



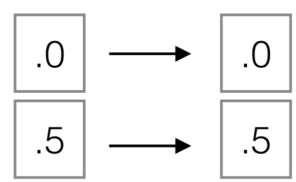
Digression : Tare weight

The tare allows you to choose the value on the screen when the plate is empty



- **even** number of fake coins

Screen does **not** change



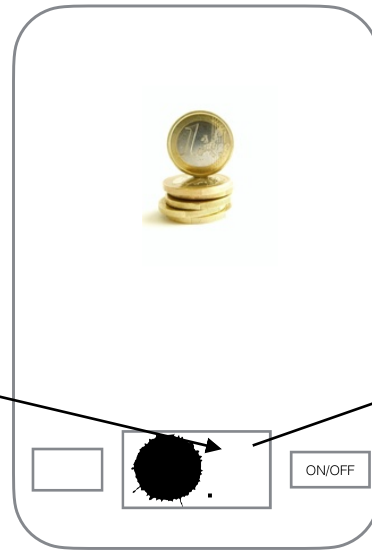
Digression : Tare weight

Initialisation

The tare allows you to choose the value on the screen when the plate is empty

.0

.5



Measure

.0

.5

- **even** number of fake coins

Screen does not change

.0	.0
.5	.5

- **odd** number of fake coins

.0	.5
.5	.0

Digression : Tare weight

Initialisation

The tare allows you to choose the value on the screen when the plate is empty

.0

.5

Measure

.0

.5

- **even** number of fake coins

Screen does **not** change

.0 → .0

.5 → .5

- **odd** number of fake coins

Screen does change

.0 → .5

.5 → .0

Mathematical modelling

$\leftrightarrow$ 0 1 0 0 1 0

A subset of n coins $\leftrightarrow$ a binary word of size n

Let $a \in \{0,1\}^n$ be the set of **fake** coins

Mathematical modelling

$\leftrightarrow$ 0 1 0 0 1 0

A subset of n coins $\leftrightarrow$ a binary word of size n

Let $a \in \{0,1\}^n$ be the set of **fake** coins

A weighing is described by a function $f_a : \{0,1\}^n \rightarrow \{0,1\}$ which associates with every subset x of coins, the parity $f_a(x)$ of fake coins in x.

$$f_a(x) = \sum_{i=1}^{n} x_i a_i \bmod 2 = x \bullet a$$

How to (classically) identify the fake coins among n?

- Greedy algorithm:
  -> Weighing coins one by one: **n Weighings**

- Better algorithm?

How to (classically) identify the fake coins among n?

- Greedy algorithm:
  -> Weighing coins one by one: **n Weighings**
- Better algorithm?

No, the greedy algorithm is optimal

How to (classically) identify the fake coins among n?

- Greedy algorithm:
  -> Weighing coins one by one: **n Weighings**

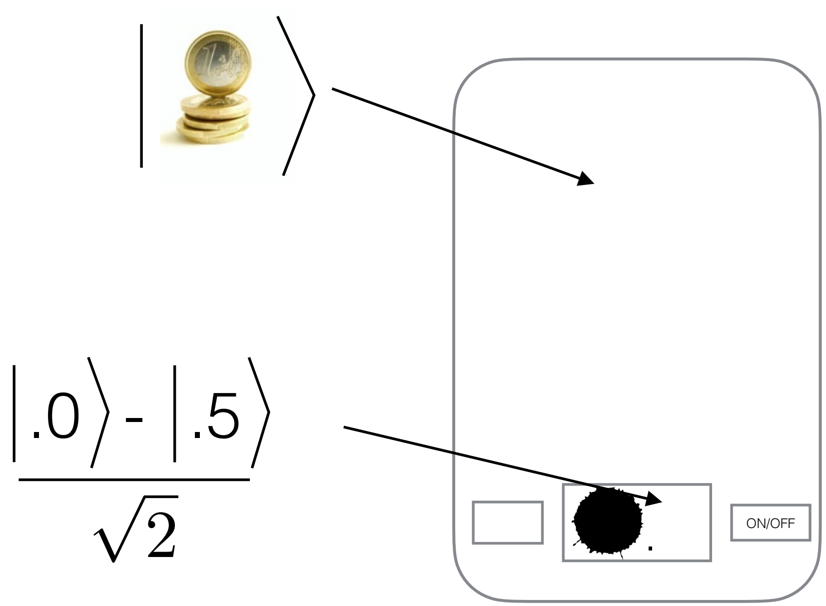
- Better algorithm?

No, the greedy algorithm is optimal

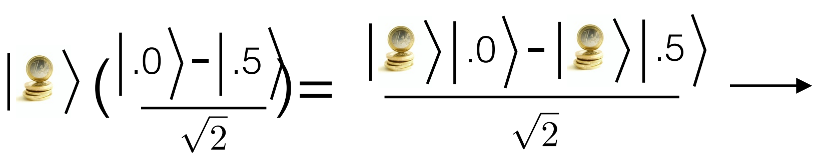
Intuition:

- Need (at least) n bits to describe the solution (because 2^n possible answers).
- Each weighing gives a single bit of information (".0" or ".5")
- So at least n weighings are necessary

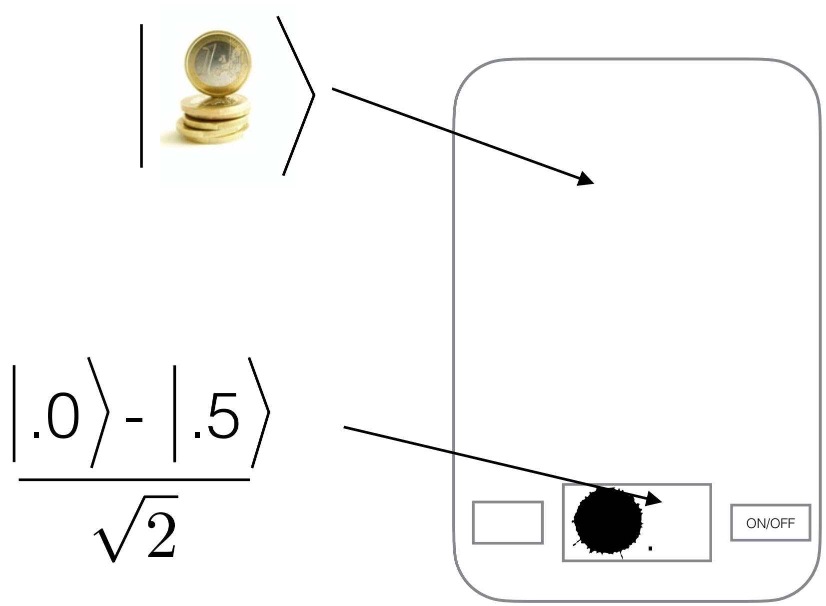
Quantum scale (disclaimer: this is a thought experiment)



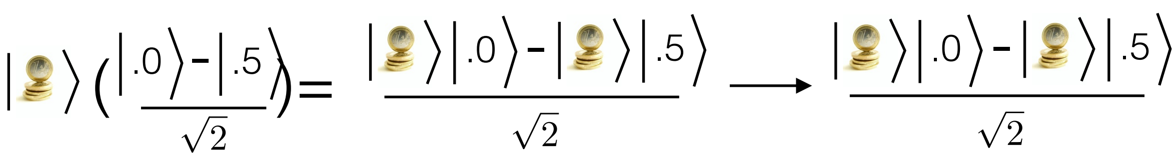
- if **even** number of fake coins:



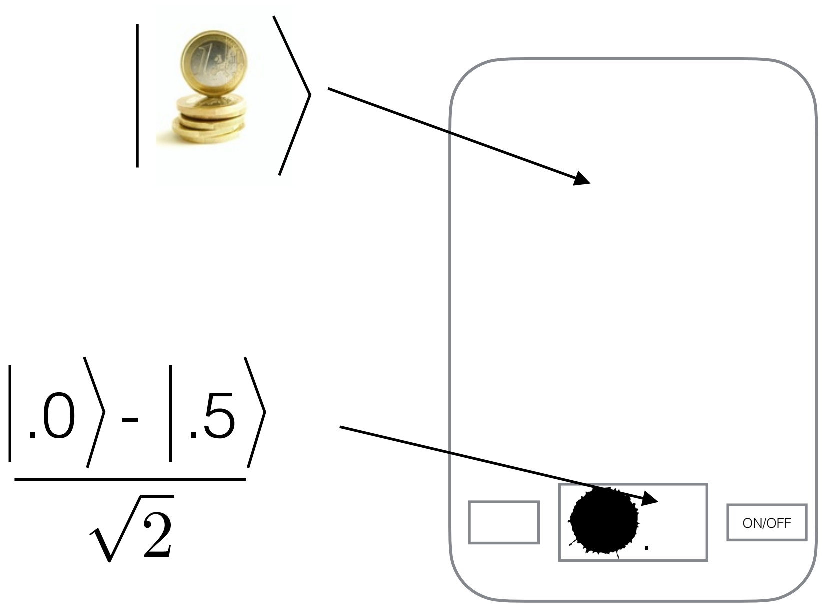
Quantum scale (disclaimer: this is a thought experiment)



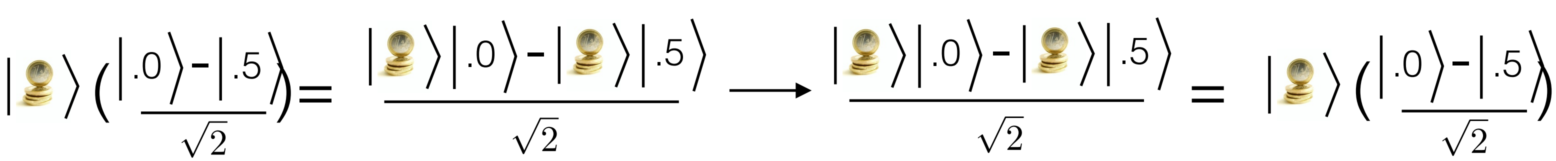
- if **even** number of fake coins:



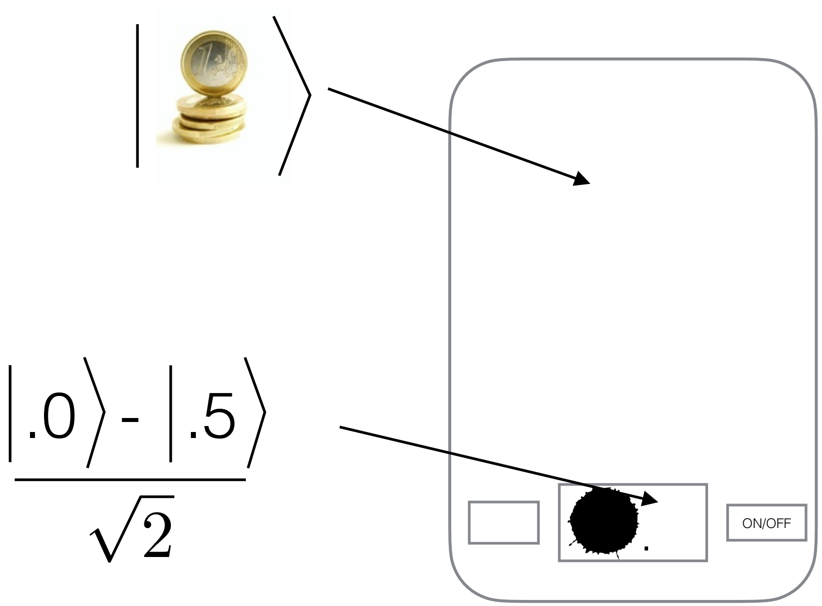
Quantum scale (disclaimer: this is a thought experiment)



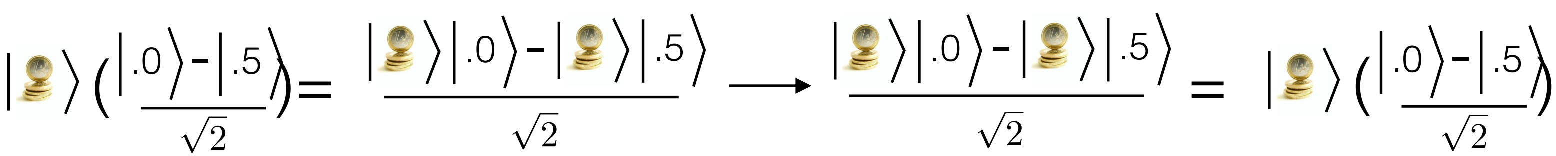
- if **even** number of fake coins:



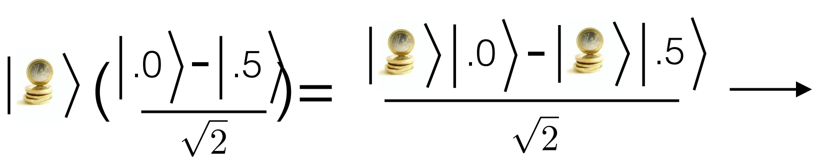
Quantum scale (disclaimer: this is a thought experiment)



- if **even** number of fake coins:

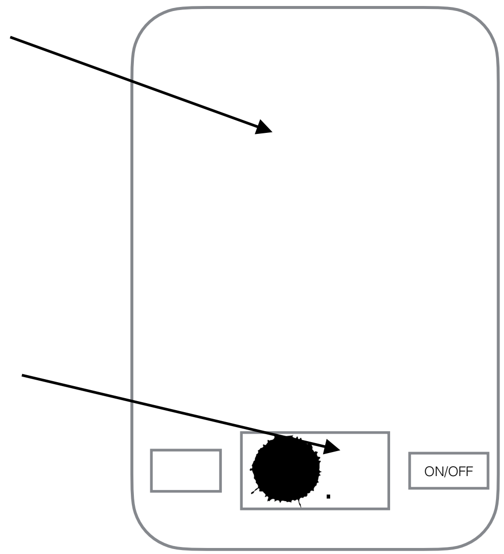


- if **odd** number of fake coins:

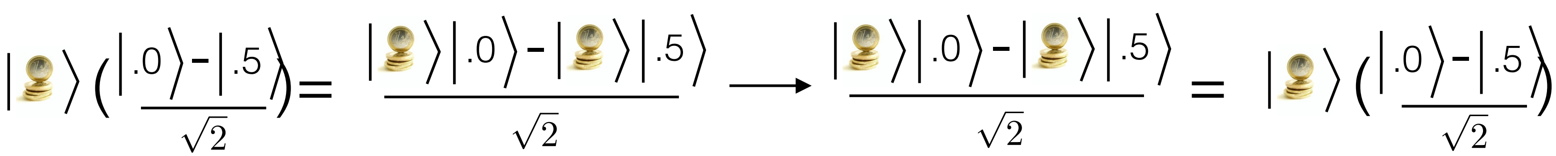


Quantum scale (disclaimer: this is a thought experiment)

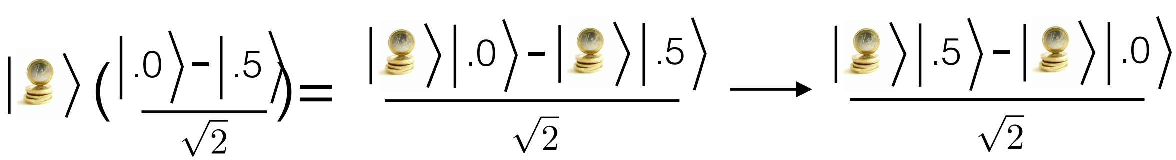
$$\frac{|.0\rangle - |.5\rangle}{\sqrt{2}}$$



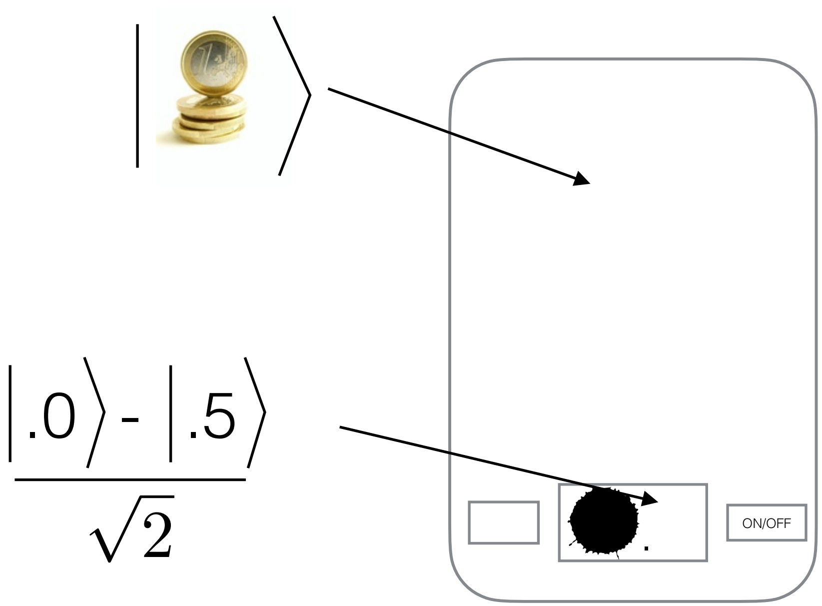
- if **even** number of fake coins:



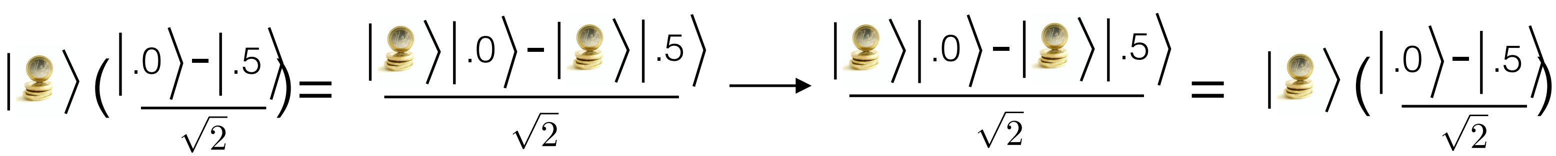
- if **odd** number of fake coins:



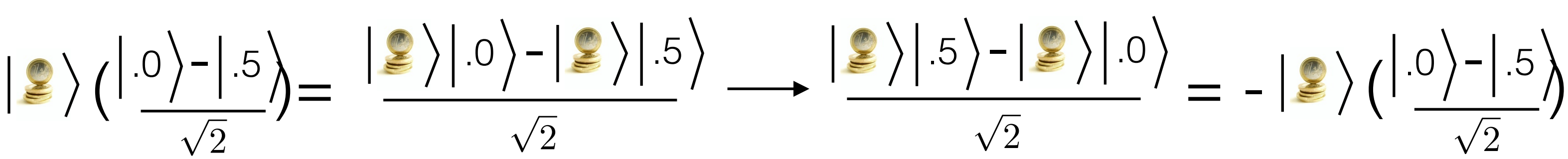
Quantum scale (disclaimer: this is a thought experiment)



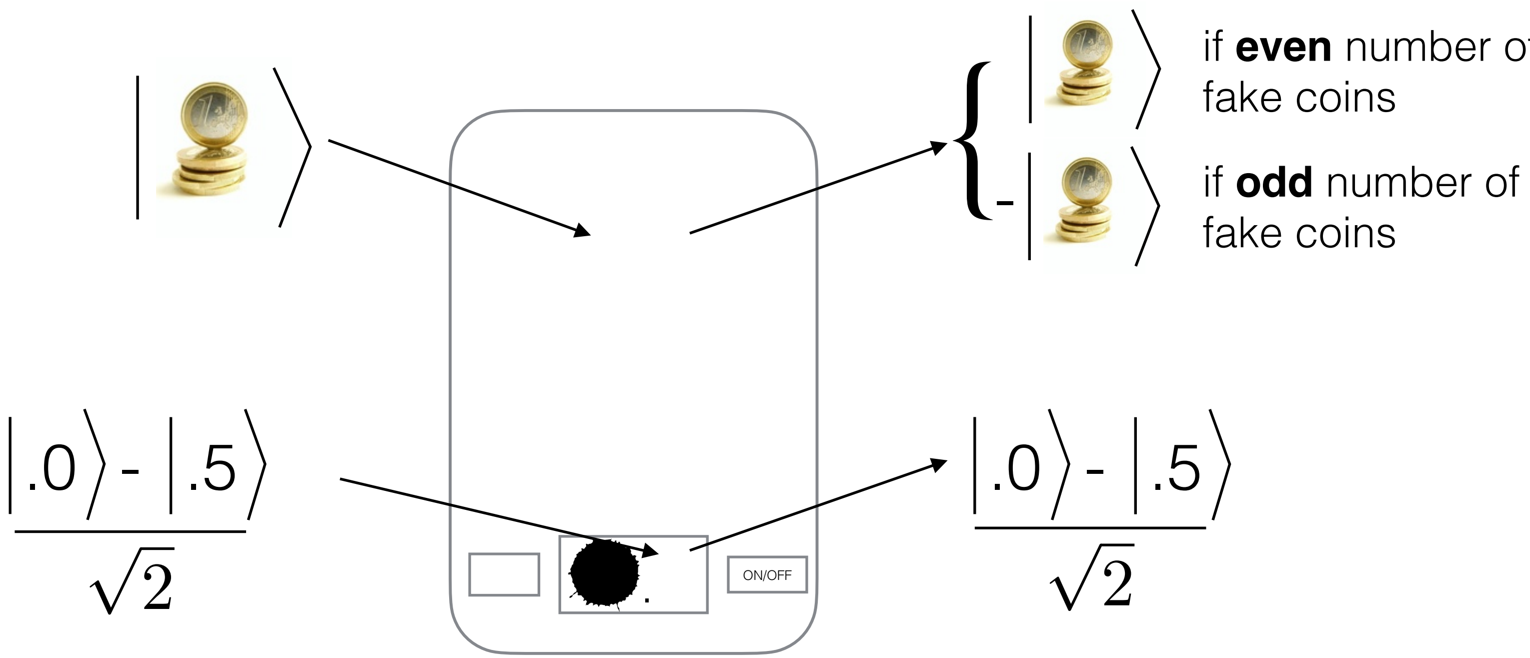
- if **even** number of fake coins:



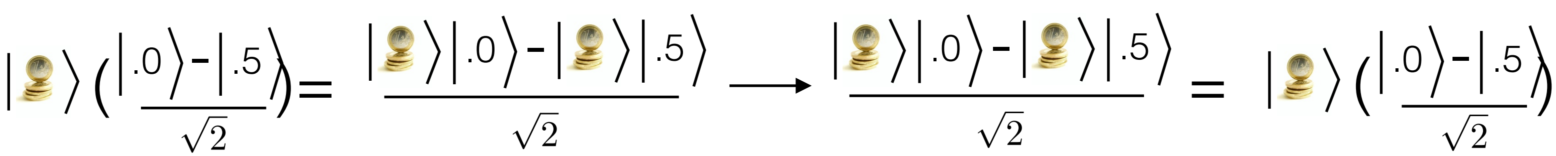
- if **odd** number of fake coins:



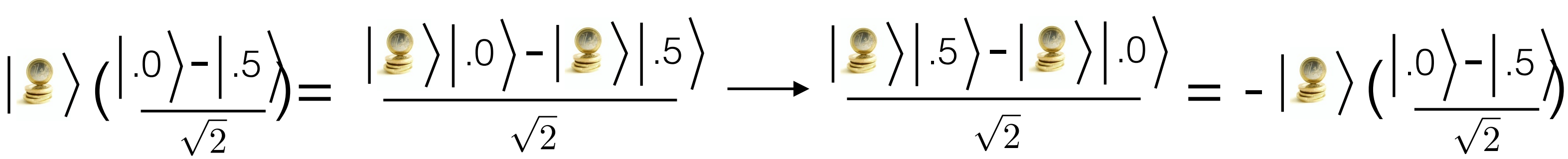
Quantum scale (disclaimer: this is a thought experiment)



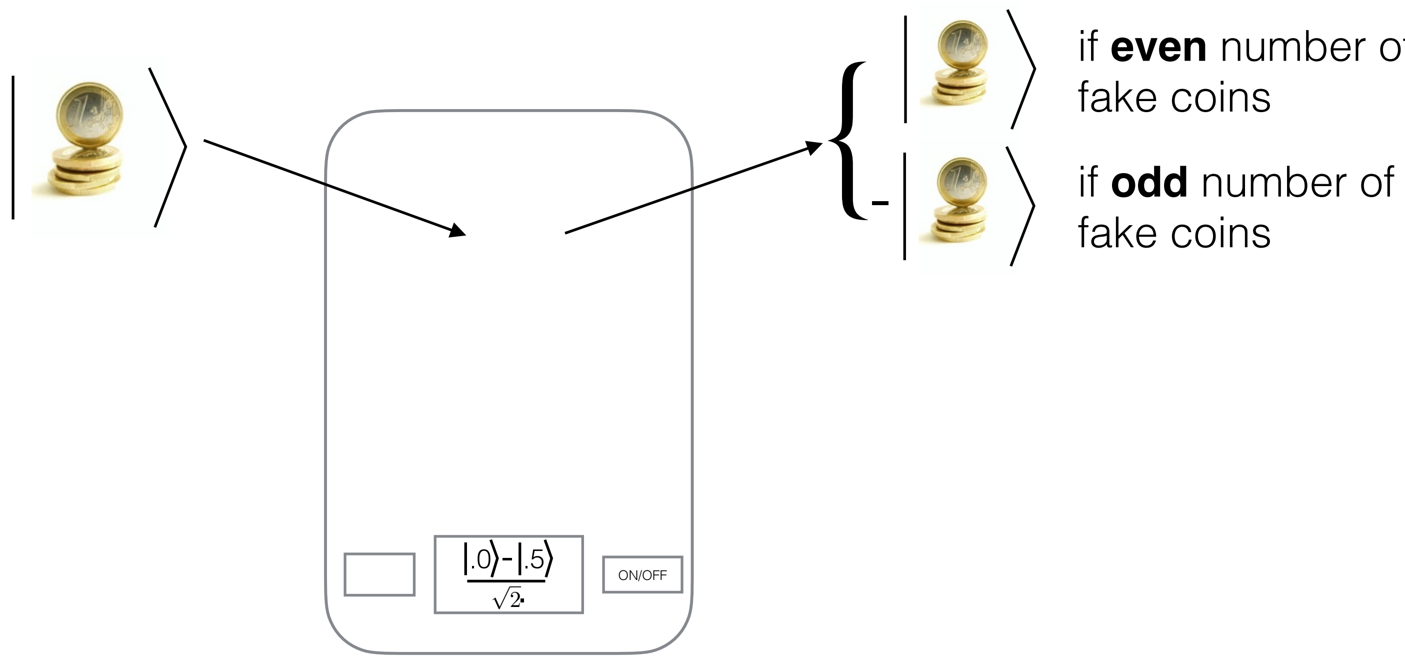
- if **even** number of fake coins:



- if **odd** number of fake coins:

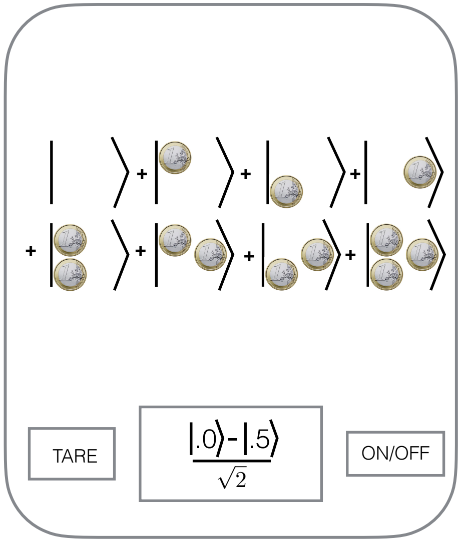


Quantum scale (disclaimer: this is a thought experiment)



$$|x\rangle \mapsto (-1)^{f_a(x)}|x\rangle = (-1)^{x\bullet a}|x\rangle$$

Bernstein-Vazirani Algorithm



$$H_n|0\ldots0\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle$$

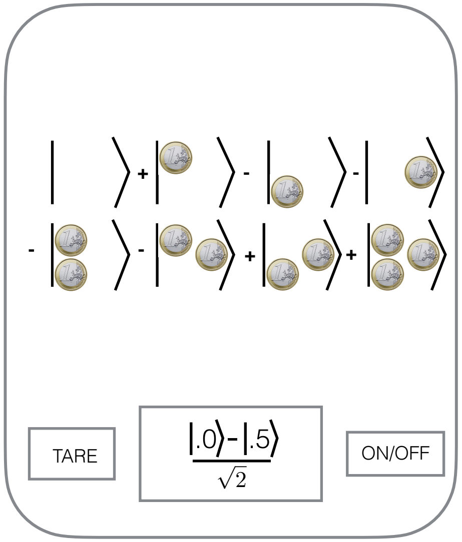
weigh. $U_{f_a} : |x\rangle \mapsto (-1)^{x\bullet a}|x\rangle$

Hadamard $H_n : |y\rangle \mapsto \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{x\bullet y}|x\rangle$

$$H_n|0\ldots0\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle$$

$$H_n \circ H_n = I$$

Bernstein-Vazirani Algorithm



weighing

$$H_n|0\ldots0\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle \quad \mapsto \quad \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{x\bullet a}|x\rangle$$

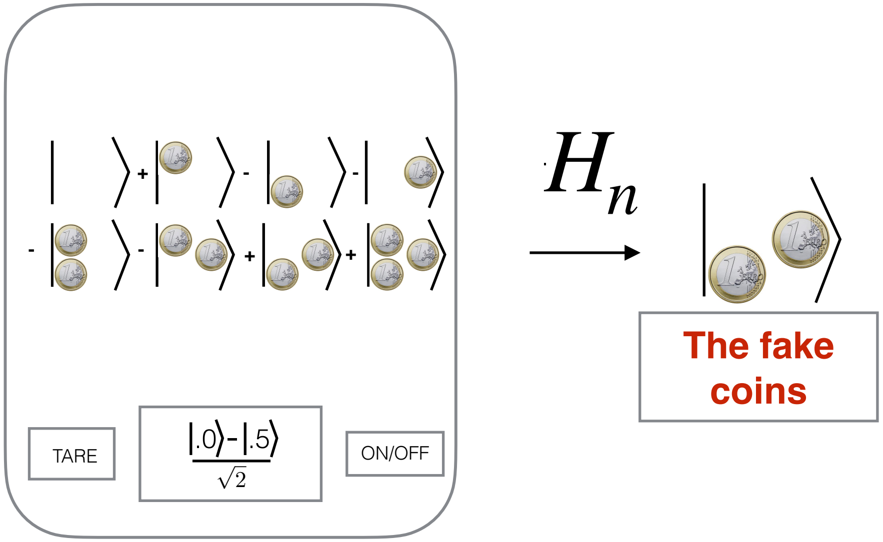
weigh. $U_{f_a} : |x\rangle \mapsto (-1)^{x\bullet a}|x\rangle$

Hadamard $H_n : |y\rangle \mapsto \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{x\bullet y}|x\rangle$

$$H_n|0\ldots0\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle$$

$$H_n \circ H_n = I$$

Bernstein-Vazirani Algorithm



weighing

$$H_n|0\ldots0\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle \quad \mapsto \quad \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{x\bullet a}|x\rangle$$

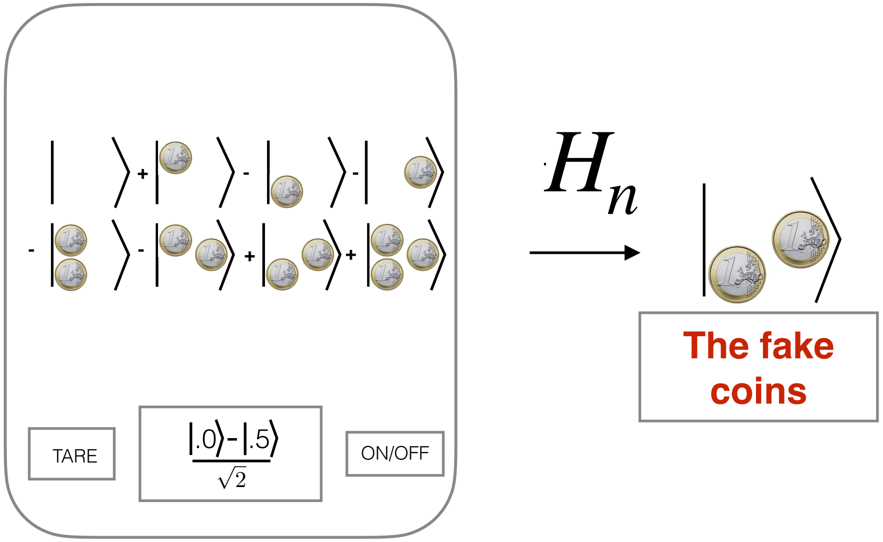
weigh. $U_{f_a} : |x\rangle \mapsto (-1)^{x\bullet a}|x\rangle$

Hadamard $H_n : |y\rangle \mapsto \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{x\bullet y}|x\rangle$

$$H_n|0\ldots0\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle$$

$$H_n \circ H_n = I$$

Bernstein-Vazirani Algorithm



weighing

$$H_n|0\ldots0\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle \mapsto \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{x\bullet a}|x\rangle = H_n|a\rangle$$

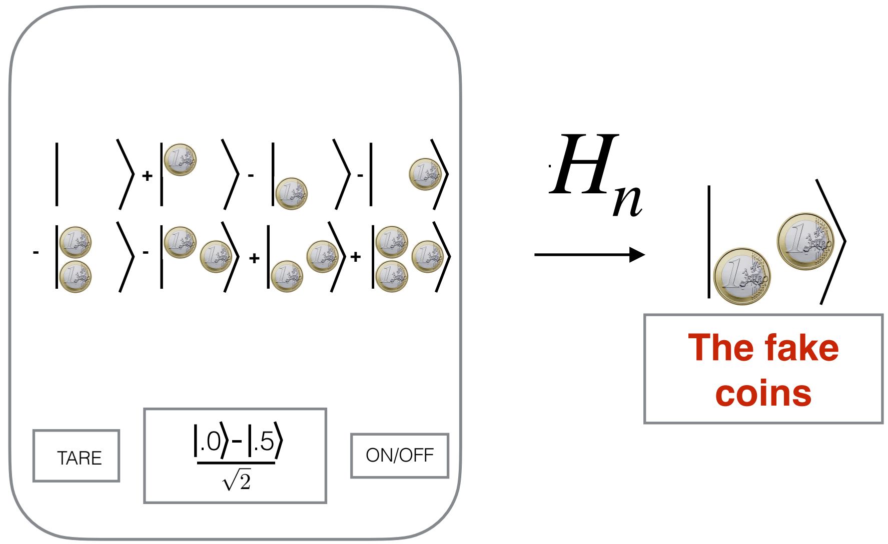
weigh. $U_{f_a} : |x\rangle \mapsto (-1)^{x\bullet a}|x\rangle$

Hadamard $H_n : |y\rangle \mapsto \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{x\bullet y}|x\rangle$

$$H_n|0\ldots0\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle$$

$$H_n \circ H_n = I$$

Bernstein-Vazirani Algorithm



weighing

$$H_n|0\ldots0\rangle = \frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}|x\rangle \;\mapsto\; \frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}(-1)^{x\bullet a}|x\rangle \;=\; H_n|a\rangle \;\overset{H_n}{\mapsto}\; = H_nH_n|a\rangle = |a\rangle$$

weigh. $U_{f_a} : |x\rangle \mapsto (-1)^{x\bullet a}|x\rangle$

Hadamard $H_n : |y\rangle \mapsto \frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}(-1)^{x\bullet y}|x\rangle$

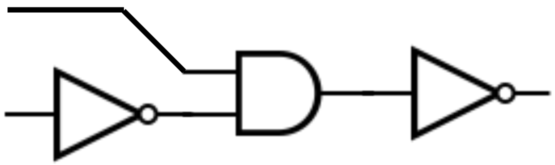
$$H_n|0\ldots0\rangle = \frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}|x\rangle$$

$$H_n \circ H_n = I$$

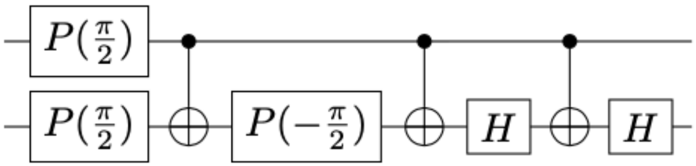
Is it fair to compare classical and quantum scales?

Is it fair to compare classical and quantum ~~scales~~? circuits

Classical circuit

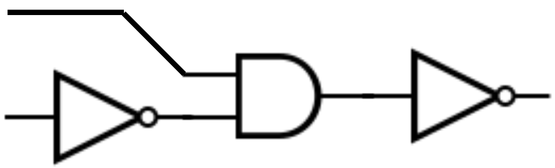


Quantum circuit

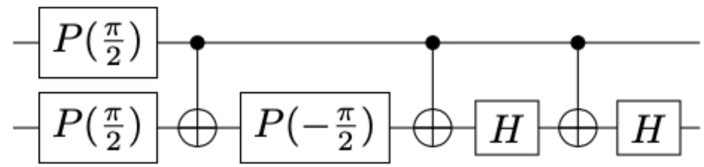


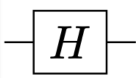
Is it fair to compare classical and quantum ~~scales~~? circuits

Classical circuit

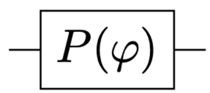


Quantum circuit

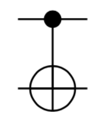




$$|0\rangle \mapsto \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$
$$|1\rangle \mapsto \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$



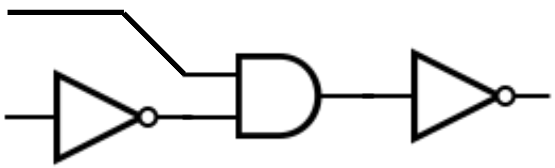
$$|0\rangle \mapsto |0\rangle$$
$$|1\rangle \mapsto e^{i\varphi}|1\rangle$$



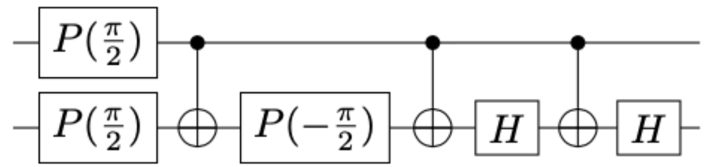
$$|00\rangle \mapsto |00\rangle$$
$$|01\rangle \mapsto |01\rangle$$
$$|10\rangle \mapsto |11\rangle$$
$$|11\rangle \mapsto |10\rangle$$

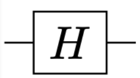
Is it fair to compare classical and quantum ~~scales~~? circuits

Classical circuit

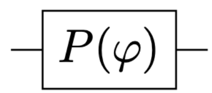


Quantum circuit

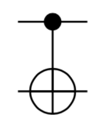




$$|0\rangle \mapsto \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$
$$|1\rangle \mapsto \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

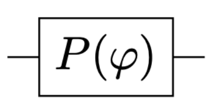


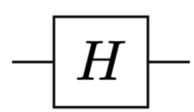
$$|0\rangle \mapsto |0\rangle$$
$$|1\rangle \mapsto e^{i\varphi}|1\rangle$$

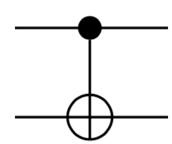


$$|00\rangle \mapsto |00\rangle$$
$$|01\rangle \mapsto |01\rangle$$
$$|10\rangle \mapsto |11\rangle$$
$$|11\rangle \mapsto |10\rangle$$

Universality: Any unitary transformation acting on a finite number of qubits can be represented by a quantum circuit which gates are:

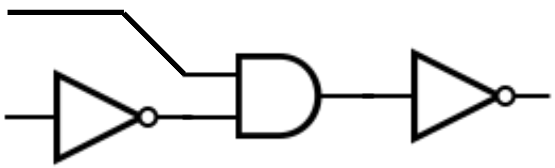




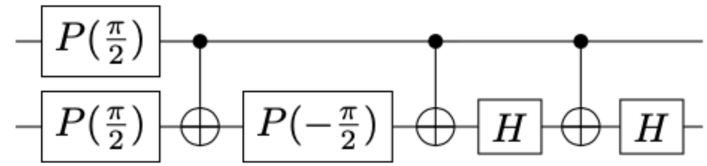


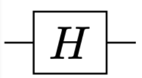
Is it fair to compare classical and quantum ~~scales~~ circuits?

Classical circuit

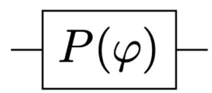


Quantum circuit

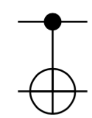




$$|0\rangle \mapsto \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$
$$|1\rangle \mapsto \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

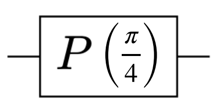


$$|0\rangle \mapsto |0\rangle$$
$$|1\rangle \mapsto e^{i\varphi}|1\rangle$$

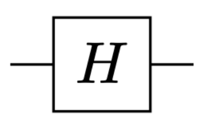


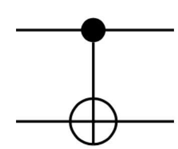
$$|00\rangle \mapsto |00\rangle$$
$$|01\rangle \mapsto |01\rangle$$
$$|10\rangle \mapsto |11\rangle$$
$$|11\rangle \mapsto |10\rangle$$

Universality: Any unitary transformation acting on a finite number of qubits can be ***approximated*** *with arbitrary precision* by a quantum circuit which gates are:



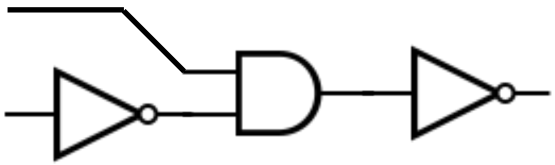
T gate



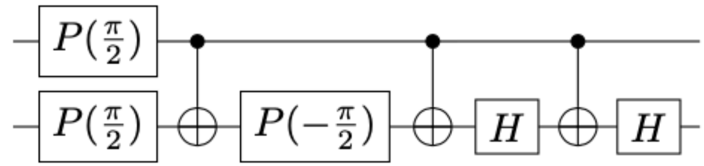


Is it fair to compare classical and quantum ~~scales~~? circuits

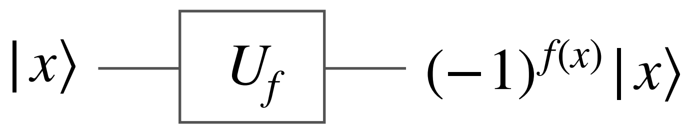
Classical circuit



Quantum circuit

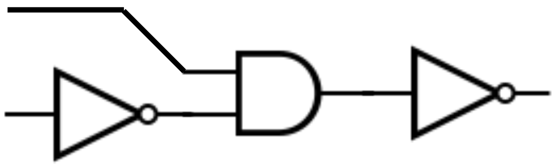


Quantum extensions of a boolean function $f : \{0,1\}^n \rightarrow \{0,1\}$:

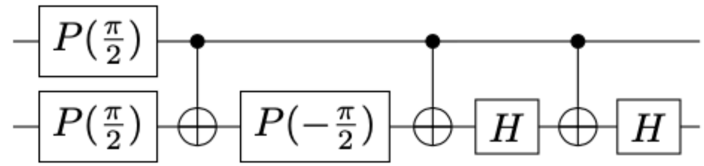


Is it fair to compare classical and quantum ~~scales~~? circuits

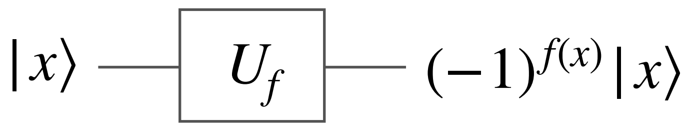
Classical circuit



Quantum circuit



Quantum extensions of a boolean function $f:\{0,1\}^n \to \{0,1\}$:

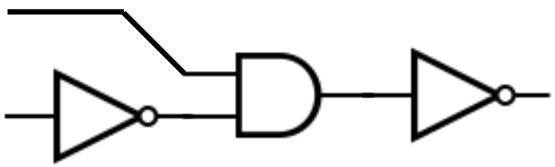


THM: if a boolean function $f:\{0,1\}^n \to \{0,1\}$ can be implemented by a boolean circuit of size s then U_f can be implemented by a quantum circuit of size $O(s)$.

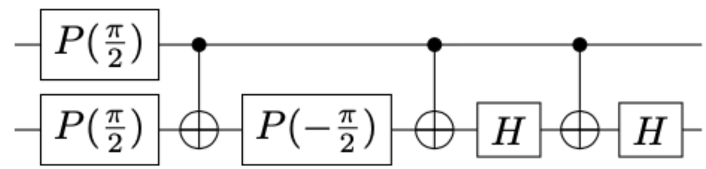
Is it fair to compare classical and quantum ~~scales~~ circuits?

YES!

Classical circuit



Quantum circuit



Quantum extensions of a boolean function $f : \{0,1\}^n \rightarrow \{0,1\}$:

$$|x\rangle \;—\; \boxed{U_f} \;—\; (-1)^{f(x)}|x\rangle$$

THM: if a boolean function $f : \{0,1\}^n \rightarrow \{0,1\}$ can be implemented by a boolean circuit of size s then U_f can be implemented by a quantum circuit of size $O(s)$.

Outline

Challenges in Quantum computing

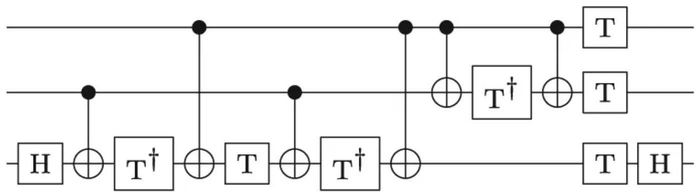
Postulates

1st Quantum Algorithm

Reasoning on Quantum Circuits

Grover

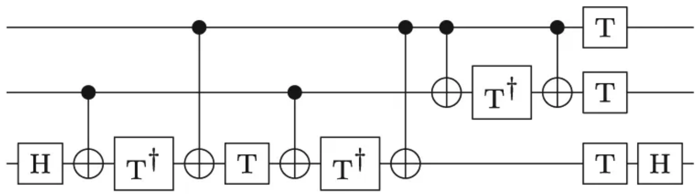
Quantum Circuits



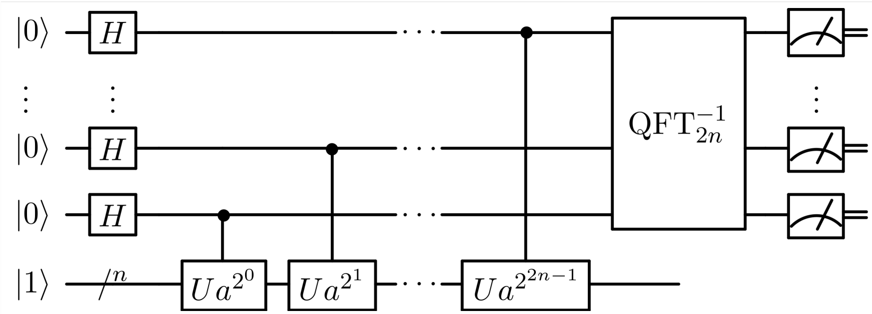
Quantum Circuits

D. Deutsch. Quantum computational networks. Proceedings of the Royal Society of London, A425:73–90, 1989.

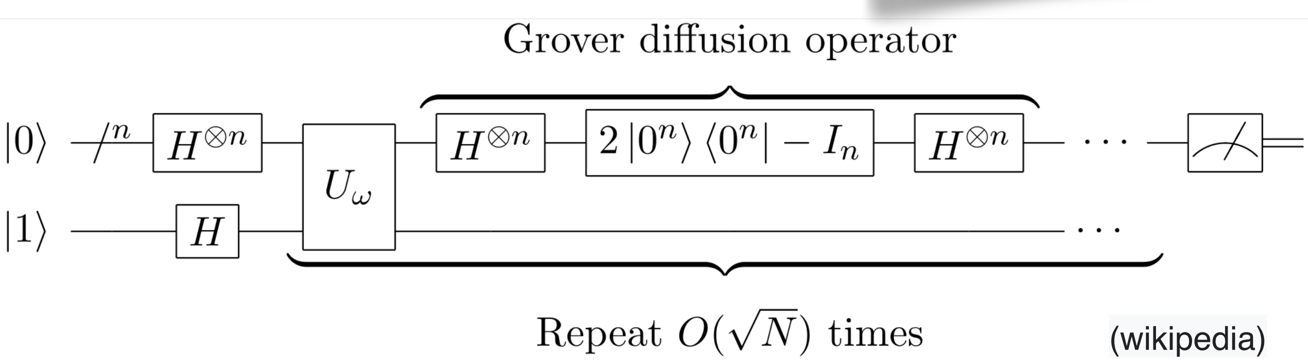
Quantum Circuits



Quantum Circuits



Quantum subroutine in Shor's algorithm (wikipedia)



(wikipedia)

D. Deutsch. Quantum computational networks. Proceedings of the Royal Society of London, A425:73-90, 1989.

Modern Quantum Programming Languages

Quipper, Qiskit, …

cf Benoit's talks

Quipper :

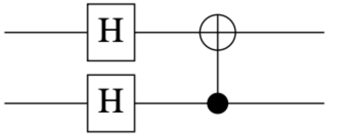
```
mycirc :: Qubit -> Qubit -> Circ (Qubit, Qubit)
mycirc a b = do
  a <- hadamard a
  b <- hadamard b
  (a,b) <- controlled_not a b
  return (a,b)
```

Modern Quantum Programming Languages

Quipper, Qiskit, …

Quipper :

```
mycirc :: Qubit -> Qubit -> Circ (Qubit, Qubit)
mycirc a b = do
  a <- hadamard a
  b <- hadamard b
  (a,b) <- controlled_not a b
  return (a,b)
```

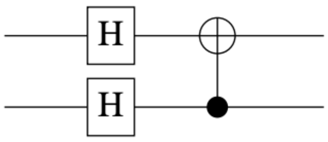


Modern Quantum Programming Languages

Quipper, Qiskit, … Langages for circuit description.

Quipper :

```
mycirc :: Qubit -> Qubit -> Circ (Qubit, Qubit)
mycirc a b = do
  a <- hadamard a
  b <- hadamard b
  (a,b) <- controlled_not a b
  return (a,b)
```

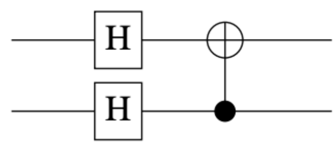


Modern Quantum Programming Languages

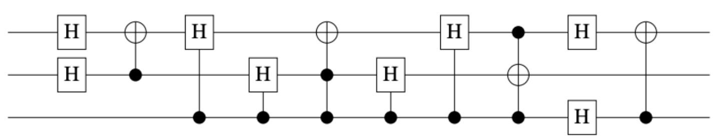
Quipper, Qiskit, … Langages for circuit description.

Quipper :

```
mycirc :: Qubit -> Qubit -> Circ (Qubit, Qubit)
mycirc a b = do
  a <- hadamard a
  b <- hadamard b
  (a,b) <- controlled_not a b
  return (a,b)
```




```
mycirc2 :: Qubit -> Qubit -> Qubit
  -> Circ (Qubit, Qubit, Qubit)
mycirc2 a b c = do
  mycirc a b
  with_controls c $ do
    mycirc a b
    mycirc b a
  mycirc a c
  return (a,b,c)
```



Modern Quantum Programming Languages

Quipper, Qiskit, … Langages for circuit description.

Quipper :

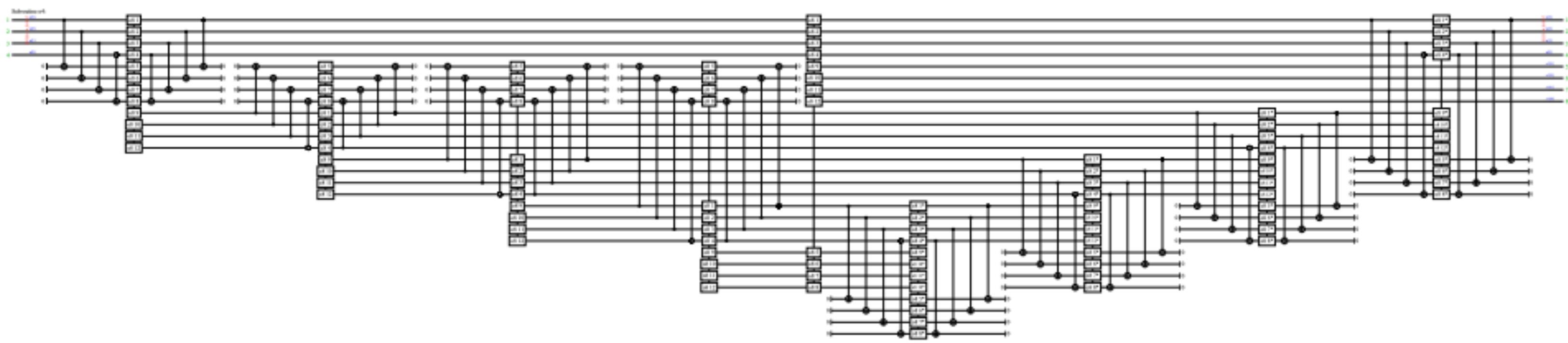
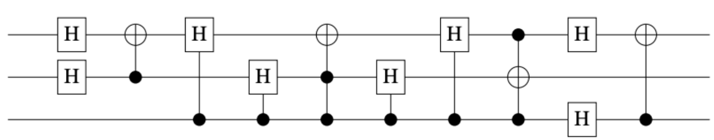


Figure 2. The circuit for o4_POW17

```
mycirc a b
with_controls c $ do
  mycirc a b
  mycirc b a
mycirc a c
return (a,b,c)
```



Modern Quantum Programming Languages

Quipper, Qiskit, … Langages for circuit description.

Quipper :

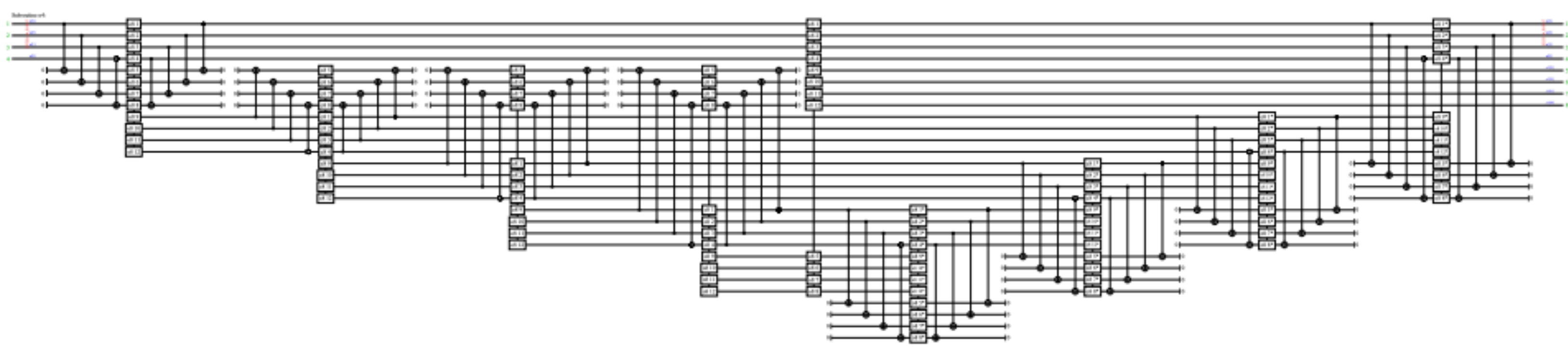


Figure 2. The circuit for o4_POW17

Figure 3. The circuit for o8_MUL

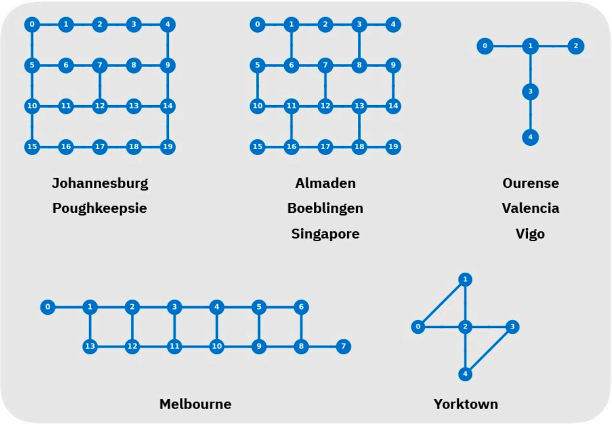
Quantum Circuits

Ubiquitous intermediate language for:

- Resource optimisation (#gates, #T, #CNot…)
- Hardware-constraint satisfaction (primitives, topological constraints, …)
- Fault-tolerant Quantum Computing
- Verification, circuit equivalence testing.

=> Circuit Transformation

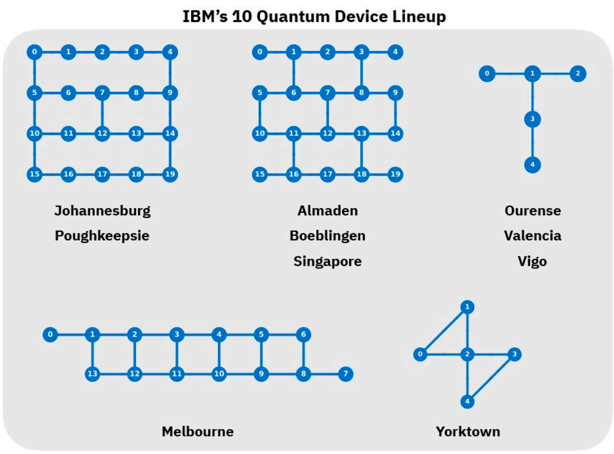
IBM's 10 Quantum Device Lineup



Quantum Circuits

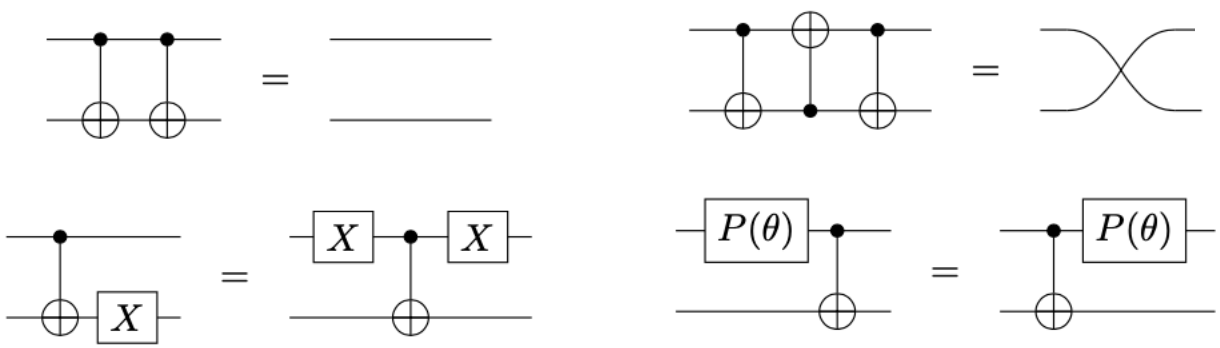
Ubiquitous intermediate language for:

- Resource optimisation (#gates, #T, #CNot…)
- Hardware-constraint satisfaction (primitives, topological constraints, …)
- Fault-tolerant Quantum Computing
- Verification, circuit equivalence testing.



=> Circuit Transformation

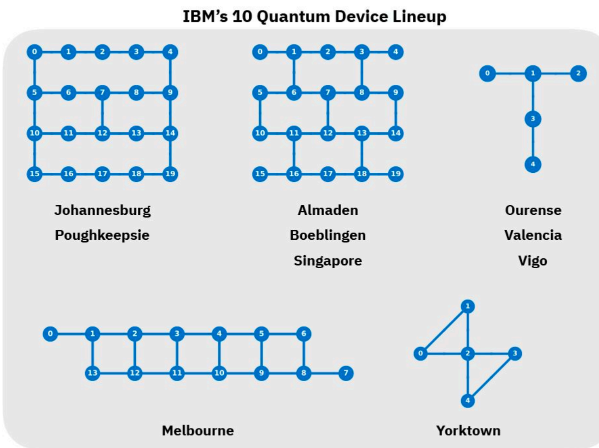
Equational theory, e.g.:



Quantum Circuits

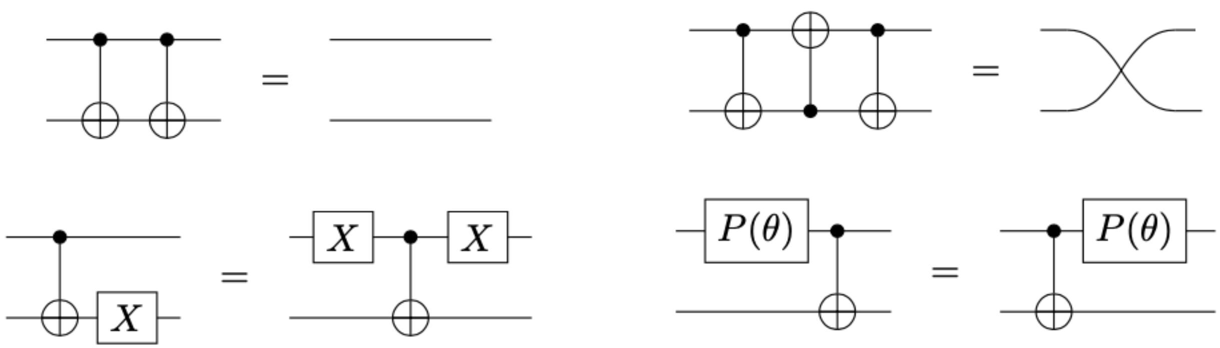
Ubiquitous intermediate language for:

- Resource optimisation (#gates, #T, #CNot…)
- Hardware-constraint satisfaction (primitives, topological constraints, …)
- Fault-tolerant Quantum Computing
- Verification, circuit equivalence testing.



=> Circuit Transformation

Equational theory, e.g.:



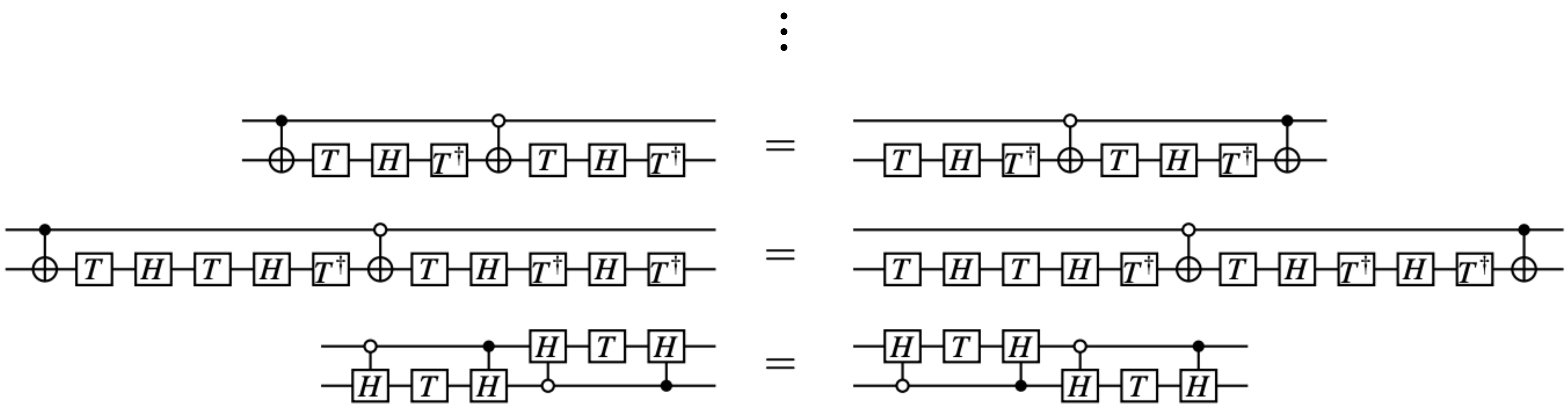
Is this equational theory complete[1]?

1. if two circuits represent the same unitary, one can be transformed into the other using the equational theory, i.e, all true equations can be derived.

Completeness

Complete equational theories for non-universal and classically simulatable **fragments**:

- 2-qubit circuits (Clifford+T) [Bian,Selinger'14]



(C18)

(C19)

(C20)

Completeness

Complete equational theories for non-universal and classically simulatable **fragments**:

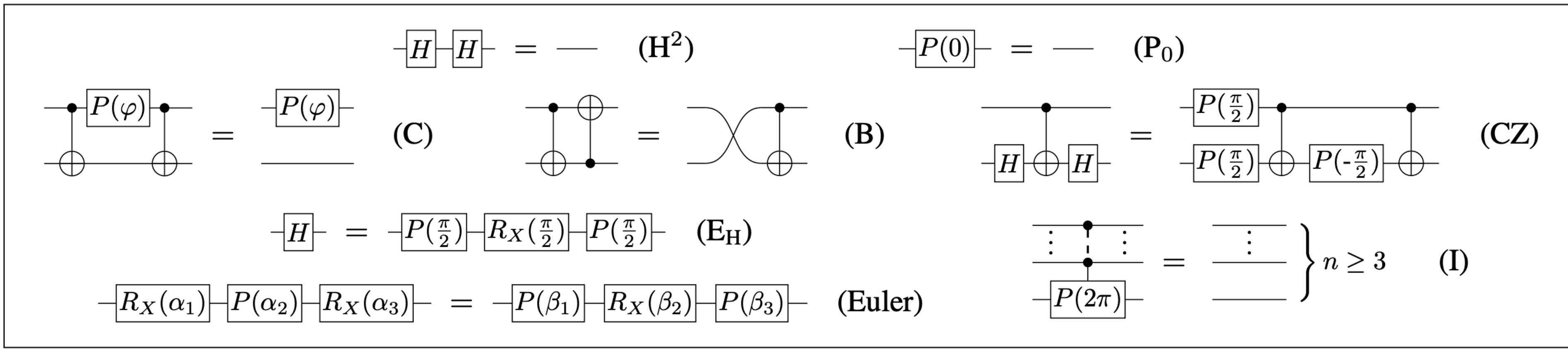
- 2-qubit circuits (Clifford+T) [Bian,Selinger'14]
- Stabilizer [Ranchin,Coecke'18], CNot-dihedral (CNot+X+T) [Amy,Chen,Ross'21].

Completeness

Complete equational theories for non-universal and classically simulatable **fragments:**

- 2-qubit circuits (Clifford+T) [Bian,Selinger'14]
- Stabilizer [Ranchin,Coecke'18], CNot-dihedral (CNot+X+T) [Amy,Chen,Ross'21].

Theorem [1,2,3]. First complete equational theory for quantum circuits.



...

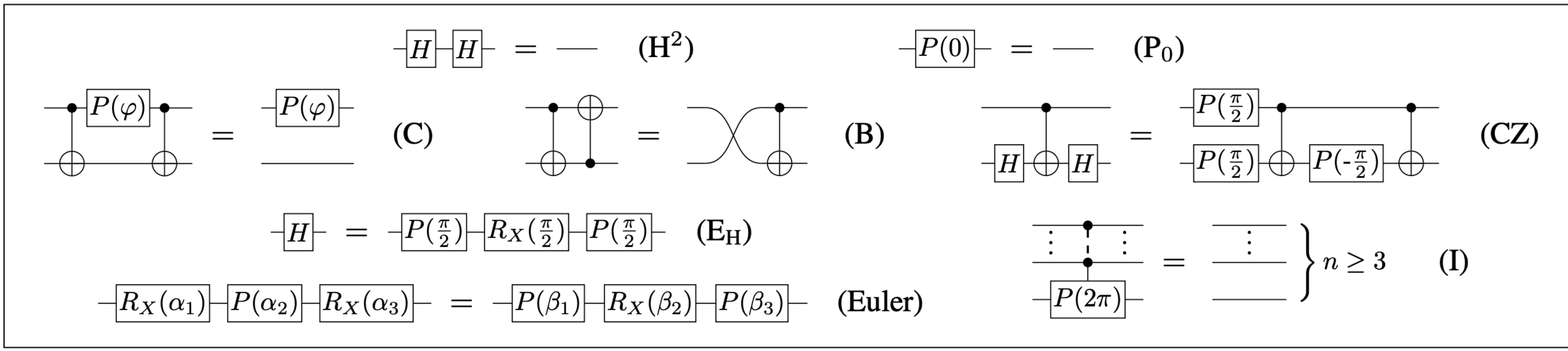
1. Clément, Heurtel, Mansfield, Perdrix, Valiron. LICS'23
2. Clément, Delorme, Perdrix, Vilmart. CSL'24
3. Clément, Delorme, Perdrix, LICS'24

Completeness

Complete equational theories for non-universal and classically simulatable **fragments:**

- 2-qubit circuits (Clifford+T) [Bian,Selinger'14]
- Stabilizer [Ranchin,Coecke'18], CNot-dihedral (CNot+X+T) [Amy,Chen,Ross'21].

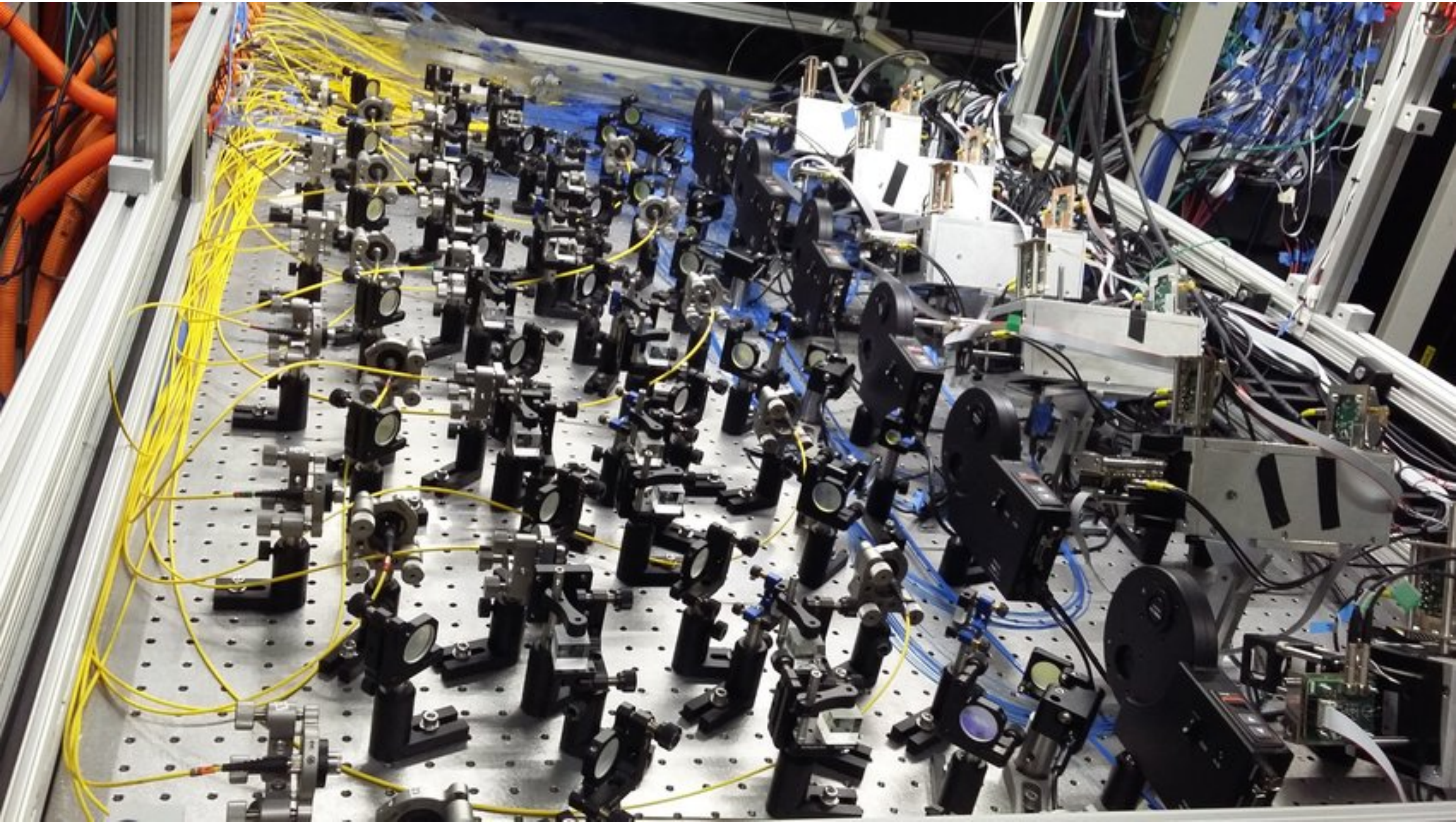
Theorem [1,2,3]. First complete equational theory for quantum circuits.



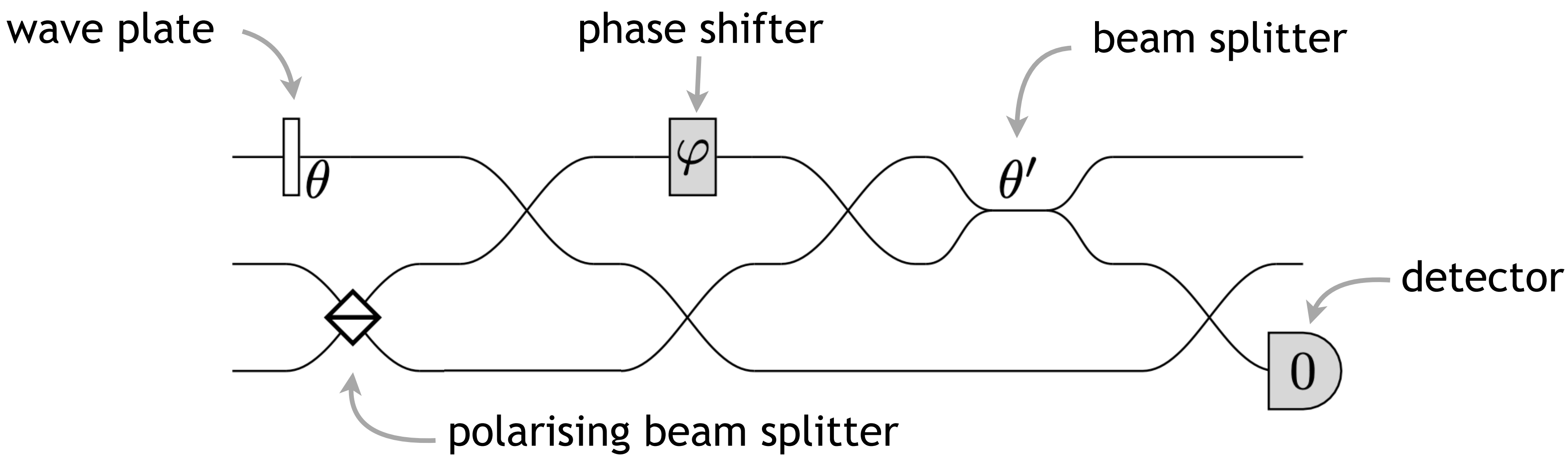
Proposition. This complete equational theory is minimal.

...

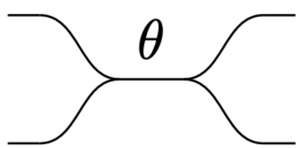
1. Clément, Heurtel, Mansfield, Perdrix, Valiron. LICS'23
2. Clément, Delorme, Perdrix, Vilmart. CSL'24
3. Clément, Delorme, Perdrix, LICS'24

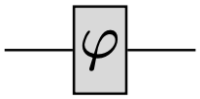


The LO$_v$-calculus



-> For this talk restriction to *beam splitters* and *phase shifters*:



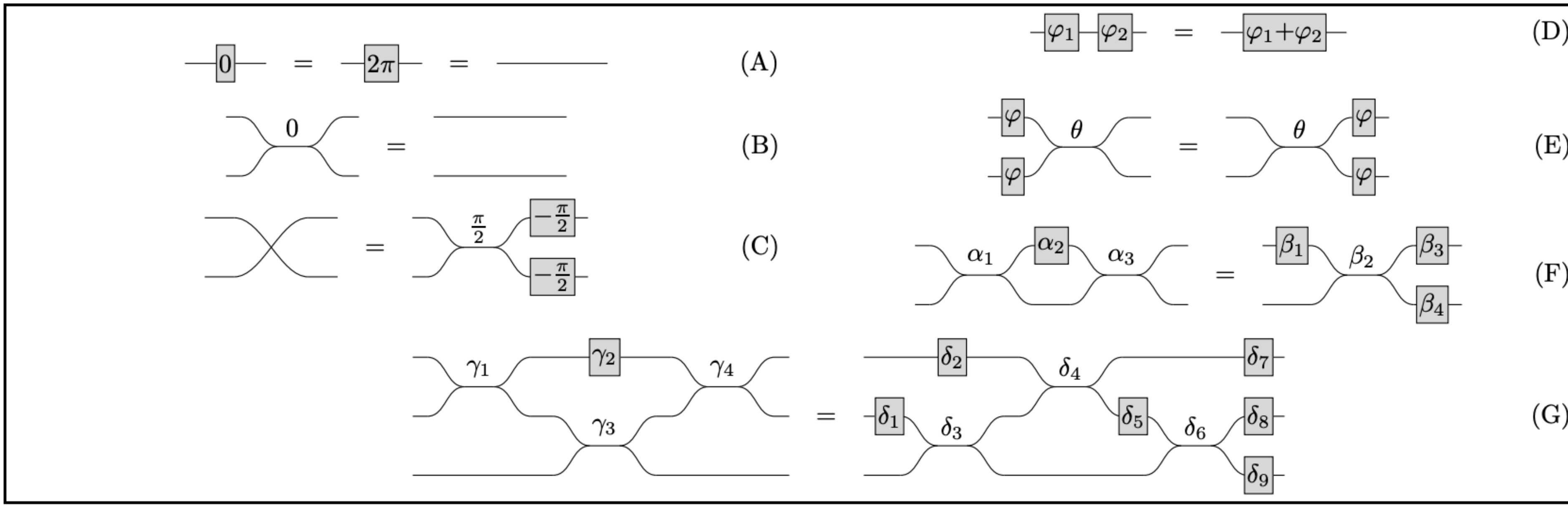


1. A. Clément, N. Heurtel, S. Mansfield, S. Perdrix, B. Valiron. LOv-Calculus: A Graphical Language for Linear Optical Quantum Circuits. MFCS'22.

Completeness

Theorem (Completeness) [Clément,Heurtel,Mansfield,Perdrix,Valiron MFCS'22]

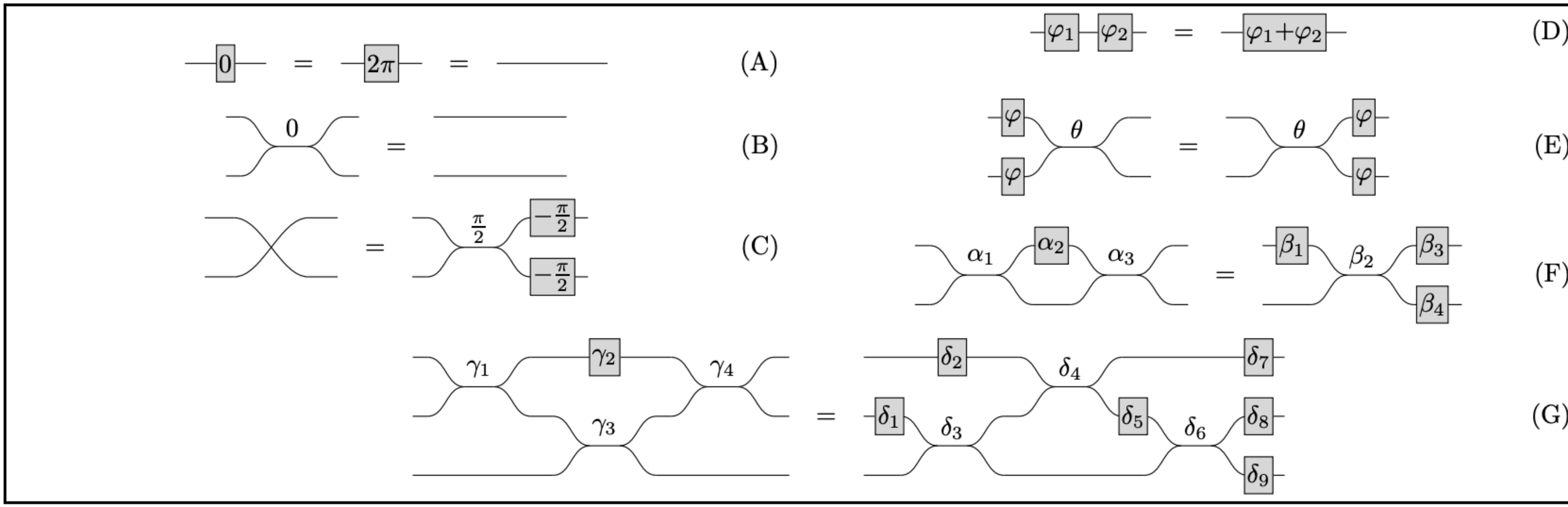
The following equational theory is complete, i.e. if $[\![C_1]\!] = [\![C_2]\!]$ then $\mathsf{LO}_v \vdash C_1 = C_2$



Completeness

Theorem (Completeness) [Clément,Heurtel,Mansfield,Perdrix,Valiron MFCS'22]

The following equational theory is complete, i.e. if $[\![C_1]\!] = [\![C_2]\!]$ then $\mathsf{LO}_v \vdash C_1 = C_2$

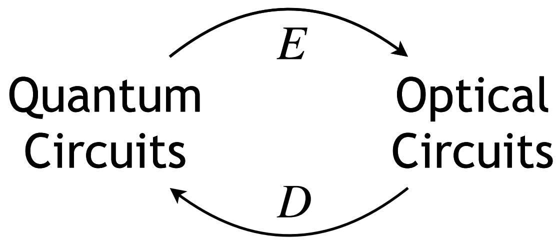


- Complete for Optical circuits
- Implemented in Perceval

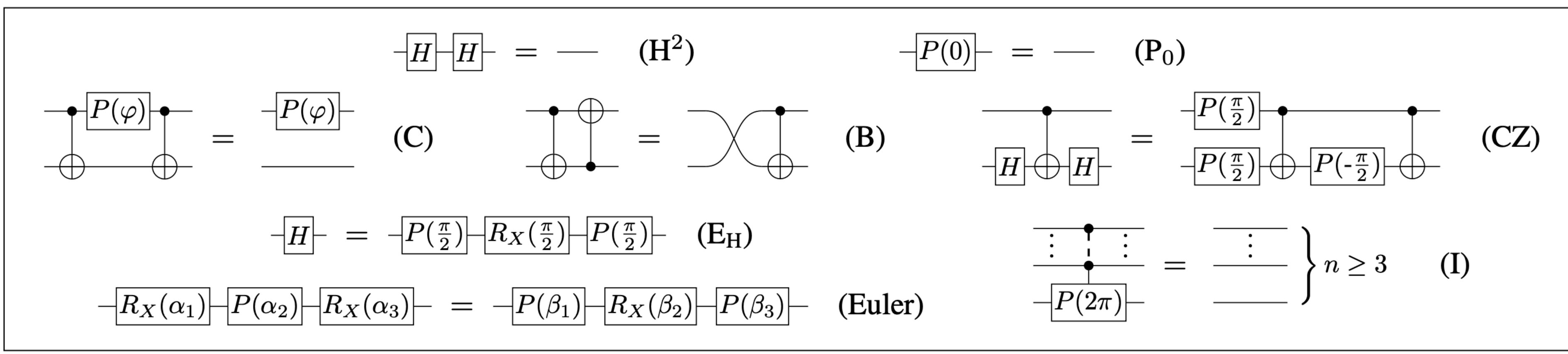
Completeness for Quantum Circuits

Parallel composition means:
- tensor product for Quantum Circuits
- direct sum for Optical Circuits

Completeness for Quantum Circuits



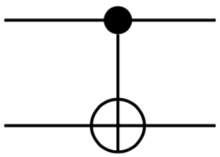
Parallel composition means:
- tensor product for Quantum Circuits
- direct sum for Optical Circuits



1. Clément, Heurtel, Mansfield, Perdrix, Valiron. LICS'23
2. Clément, Delorme, Perdrix, Vilmart. CSL'24
3. Clément, Delorme, Perdrix, LICS'24

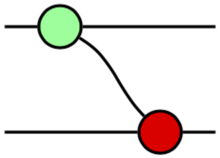
ZX-calculus [Coecke-Duncan'08]

CNot in circuit



elementary
quantum gate

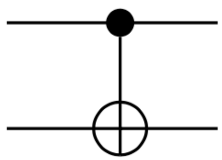
CNot in ZX



cf Miriam's talk

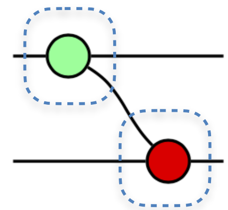
ZX-calculus [Coecke-Duncan'08]

CNot in circuit



elementary
quantum gate

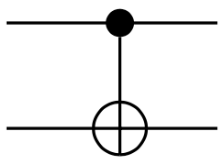
CNot in ZX



cf Miriam's talk

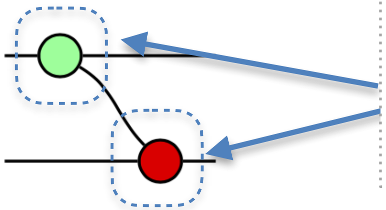
ZX-calculus [Coecke-Duncan'08]

CNot in circuit



elementary
quantum gate

CNot in ZX

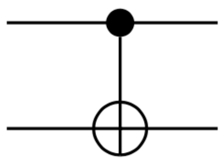


Mathematically well-defined
but not necessarily
(deterministically)
implementable

cf Miriam's talk

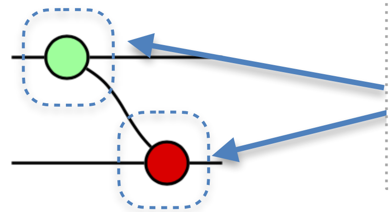
ZX-calculus [Coecke-Duncan'08]

CNot in circuit



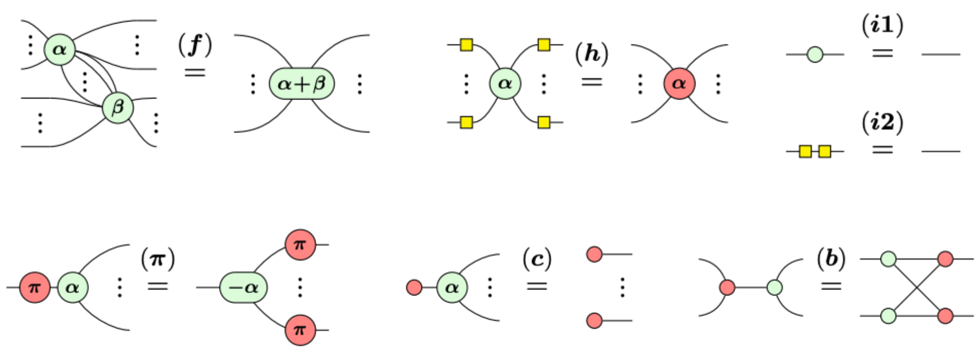
elementary
quantum gate

CNot in ZX



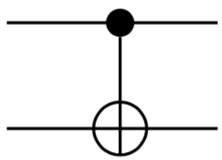
Mathematically well-defined
but not necessarily
(deterministically)
implementable

cf Miriam's talk



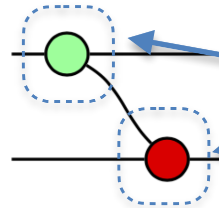
ZX-calculus [Coecke-Duncan'08]

CNot in circuit

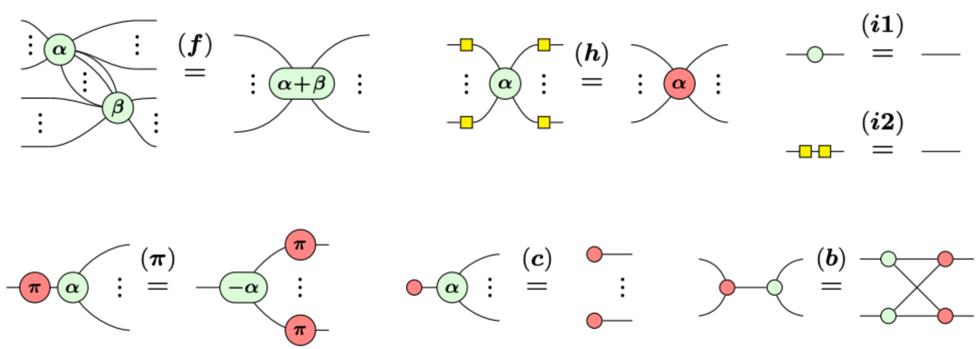


elementary
quantum gate

CNot in ZX



Mathematically well-defined
but not necessarily
(deterministically)
implementable



cf Miriam's talk

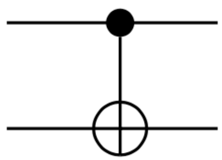
Completeness results
- Clifford (classical simulatable) [Backens'14]
- Clifford+T (approx. Universal) [Jeandel, Perdrix, Vilmart'17]
- Universal [Ng, Wang'17]

⋮

- Universal, nearly minimal [Vilmart'19]

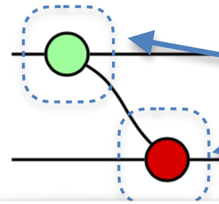
ZX-calculus [Coecke-Duncan'08]

CNot in circuit

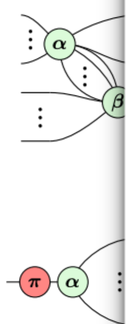


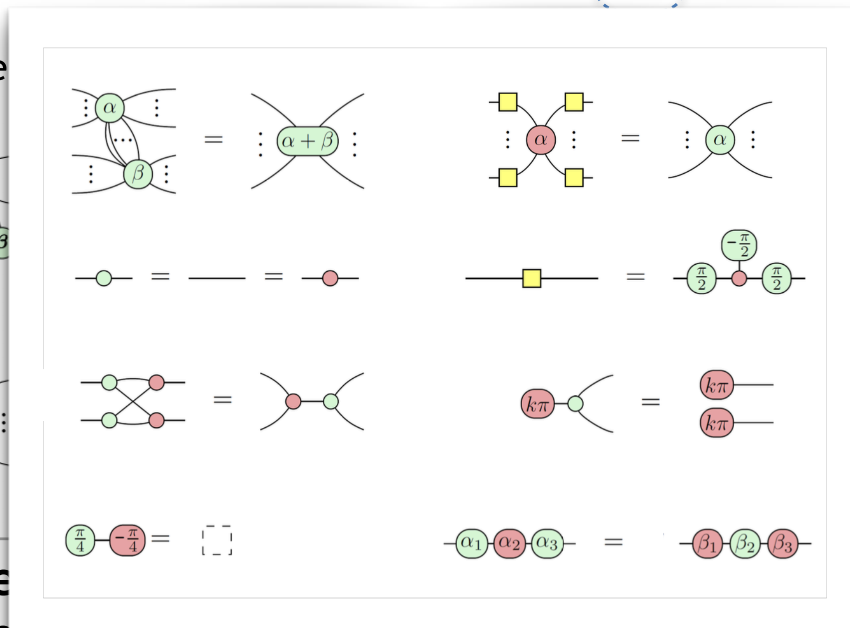
elementary
quantum gate

CNot in ZX



Mathematically well-defined
but not necessarily
(deterministically)
implementable





cf Miriam's talk

Completene
- Clifford (classical simulatable) [Backens'14]
- Clifford+T (approx. Universal) [Jeandel, Perdrix, Vilmart'17]
- Universal [Ng, Wang'17]

  ⋮

- Universal, nearly minimal [Vilmart'19]