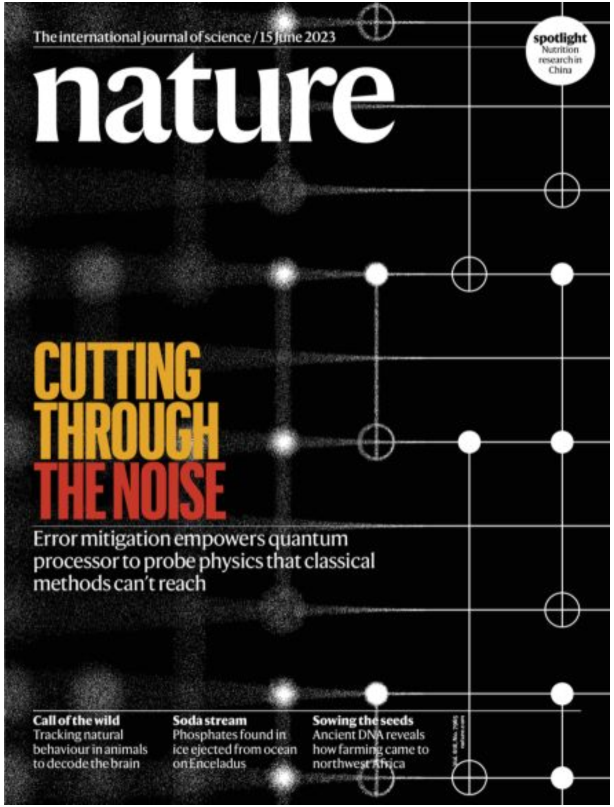
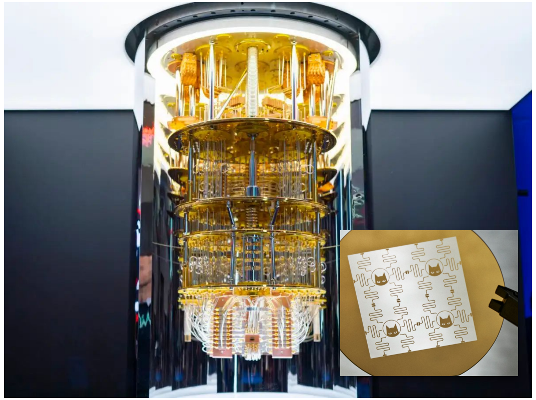
How to use a quantum computer

Chris Heunen

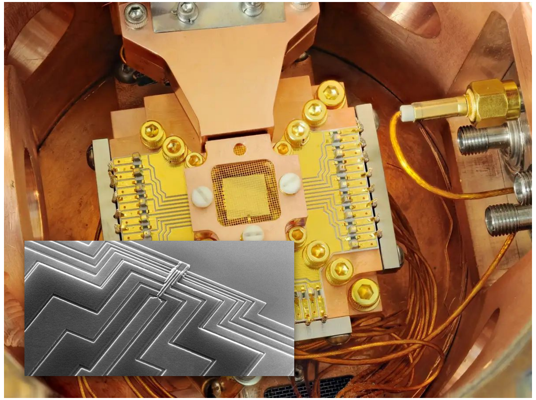
What is a quantum computer?


The international journal of science / 24 October 2019
index
Young universities
nature
QUANTUM SUPREMACY
Classical supercomputer outperformed by quantum chip for the first time
Secret history
Fossils rescue mammals from the shadow of dinosaurs
History of science
Industrial research – from Nylon to Nobels to now
Bacterial activity
How gut microbiota influences fear-related learning

The international journal of science / 15 June 2023
spotlight
Nutrition research in China
nature
CUTTING THROUGH THE NOISE
Error mitigation empowers quantum processor to probe physics that classical methods can't reach
Call of the wild
Tracking natural behaviour in animals to decode the brain
Soda stream
Phosphates found in ice ejected from ocean on Enceladus
Sowing the seeds
Ancient DNA reveals how farming came to northwest Africa

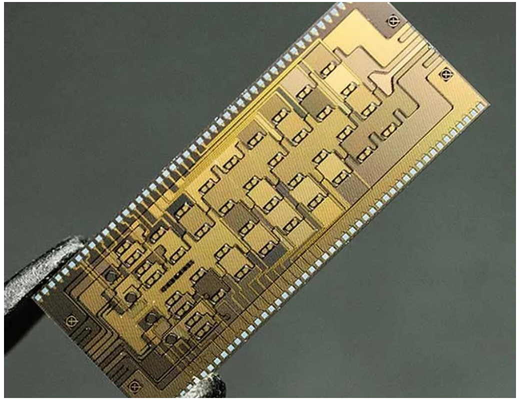
The international journal of science / 1 February 2024
nature
SIGNIFICANT OTTERS
How restored top predators helped slow down coastal damage
Hitting a nerve
Hijacked neurons boost cancer's ability to grow and spread
What's in a name?
Why current terms for metastatic cancers can delay treatment
Drawn threads
Mechanical process yields flexible fibres for wearable electronics



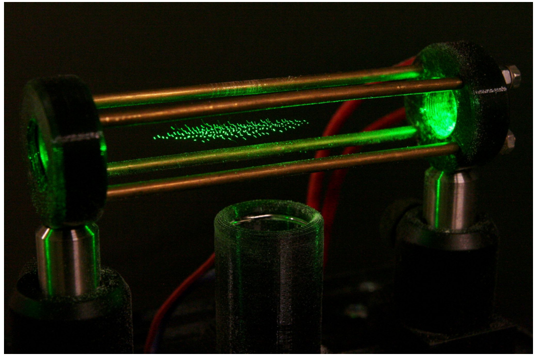
Superconducting



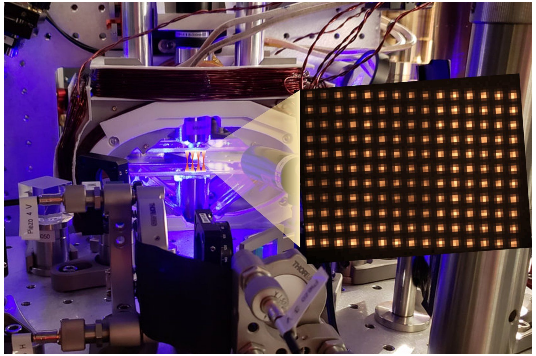
Spin



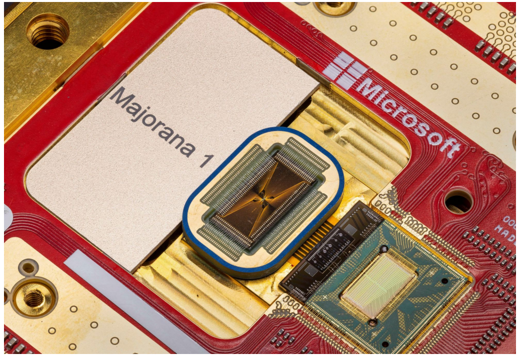
Photons



Trapped ions



Neutral atoms

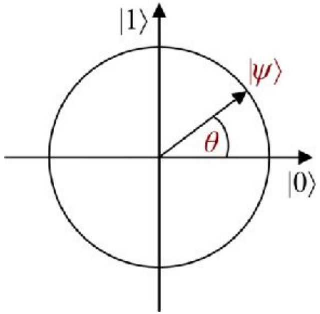


Majorana fermions

What is quantum information?

Qubits

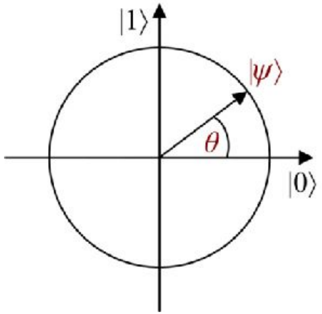
- Elements of unit circle in $\mathbb{C}^2$
- false = (1,0) true = (0,1)



- Also |+⟩ = (1,1)/√2 |-⟩ = (1,-1)/√2
- Manipulate by 2x2 unitary matrices
- Cannot clone or delete

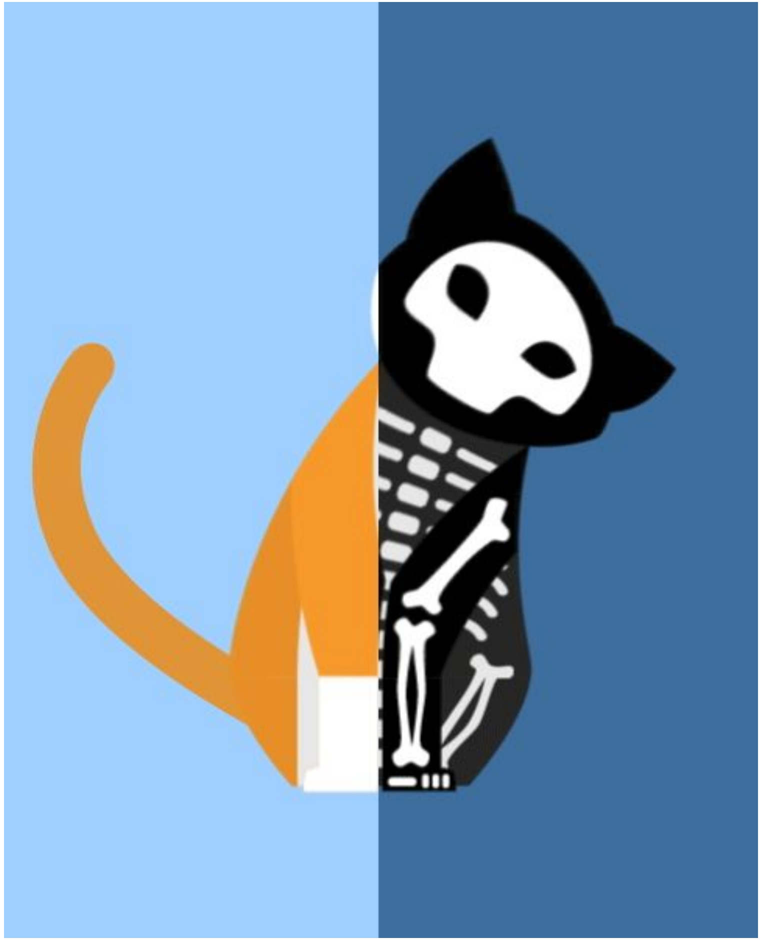
Measurement

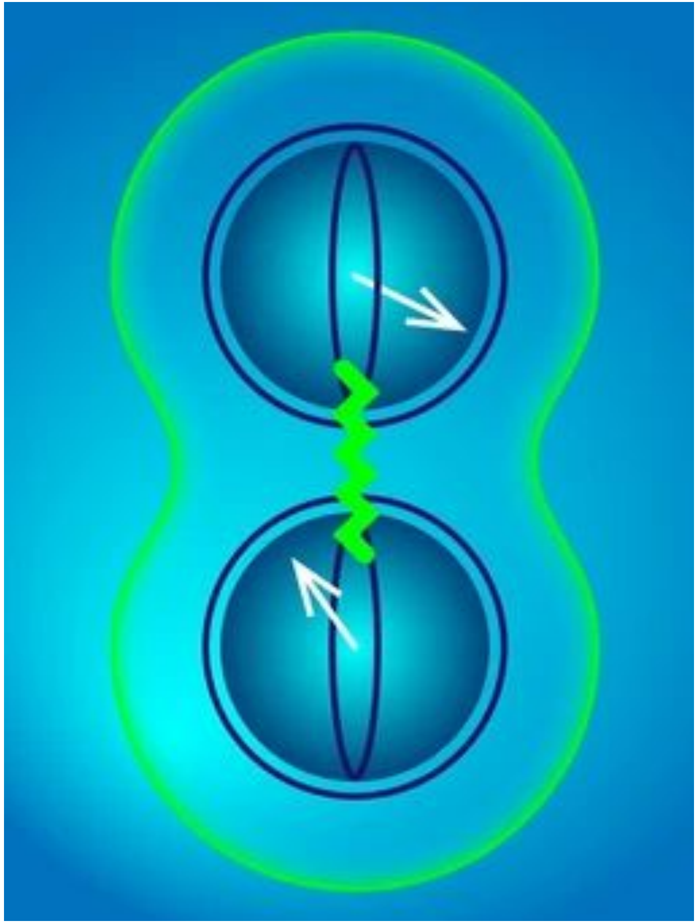
- Probabilistic operation qubit ⟶ bit



true with probability $|\langle\psi|1\rangle|^2$

- Depends on basis/angle
- Collapse after measurement





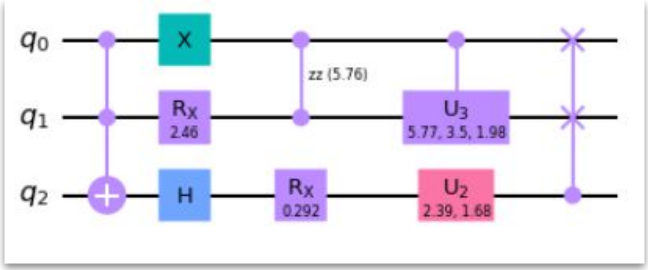
Multiple qubits

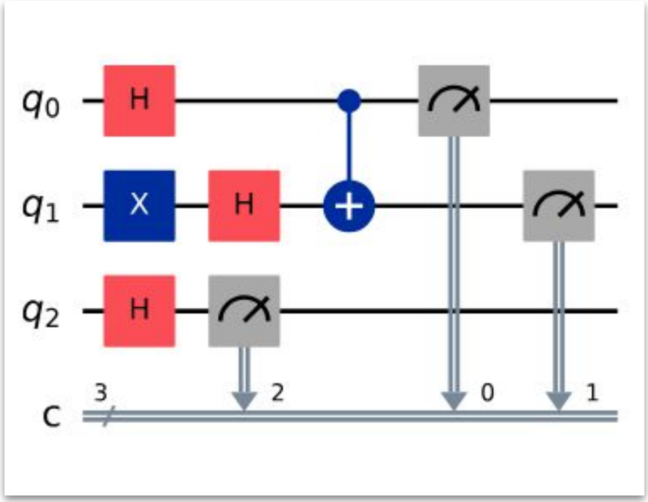
- Compound systems given by tensor product

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \otimes \begin{pmatrix} e & f \\ g & h \end{pmatrix} = \begin{pmatrix} ae & af & be & bf \\ ag & ah & bg & bh \\ ce & cf & de & df \\ cg & ch & dg & dh \end{pmatrix}$$

- State not determined by factors
- Dimension multiplies
- Measurement of one qubit influences other

How not to use a quantum computer





Circuits

- Time flows left to right
- Space goes up and down
- Qubits undergo unitary gates
- Gates can be controlled by multiple qubits
- Bits may influence control flow

$$^{*}\left(H = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix} \quad X = \begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix} \quad Y = \begin{bmatrix}0 & -i\\ i & 0\end{bmatrix} \quad Z = \begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix} \quad T = \begin{bmatrix}1 & 0\\ 0 & e^{i\pi/4}\end{bmatrix} \quad CX = \begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix} \right)$$

Universal:

- Computational universality:
  can compute every computable function

But:

- Hard to verify
- Hard to manipulate
- Hard to discover algorithms
- Hard to scale
- Hard to reuse classical infrastructure

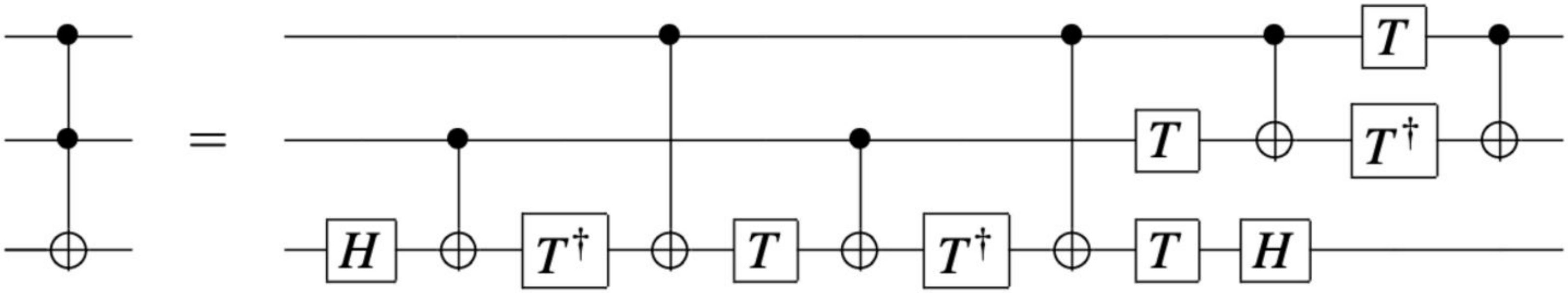
Scale

This circuit adds two 8-bit numbers:

Reasoning

Toffoli gate has many 'obvious' properties …

… that are completely obscured once expressed by elementary gates



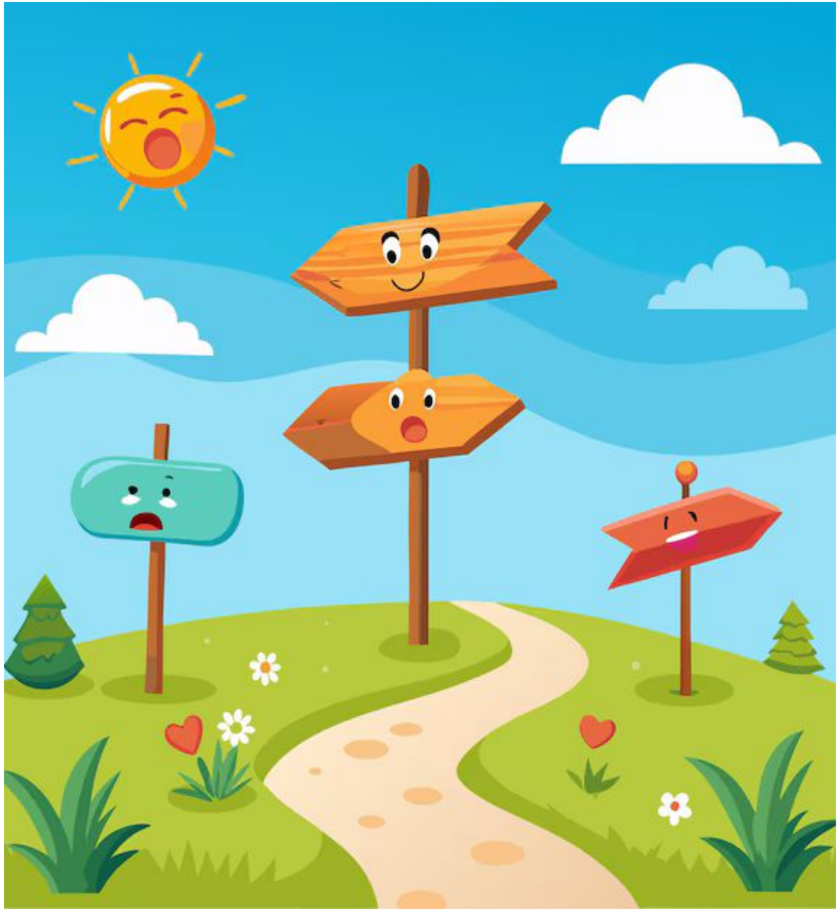
Quantum conditional

- $[\![\texttt{if } q \texttt{ then } U \texttt{ else } V]\!]$

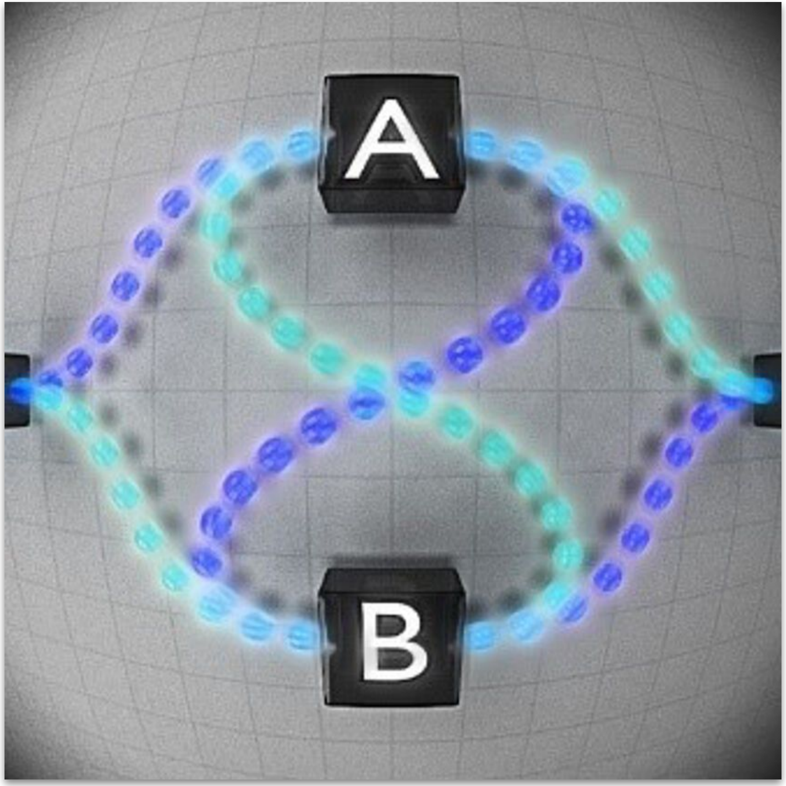
$$= \begin{pmatrix} [\![U]\!] & 0 \\ 0 & [\![V]\!] \end{pmatrix}$$

- Both branches in superposition

- Not monotone
  (unlike probabilistic computing)



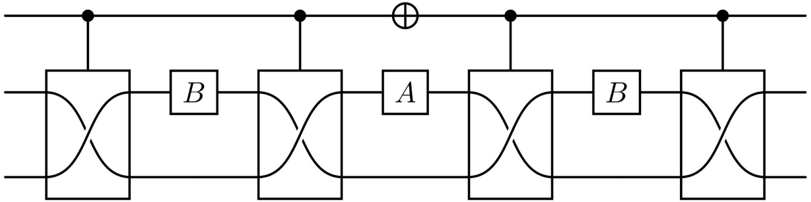
Coherent Control



- Quantum switch:

  `if q then AB else BA`

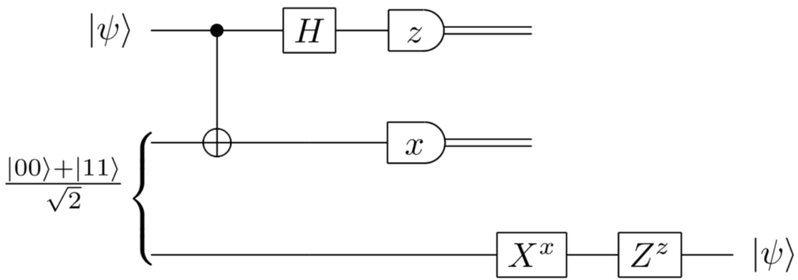
- Possible as circuit
  (with multiple uses of oracles A and B)



- Possible experimentally
  (with single use of oracles)
- Impossible as circuit
  (with single use of oracles)

Teleportation

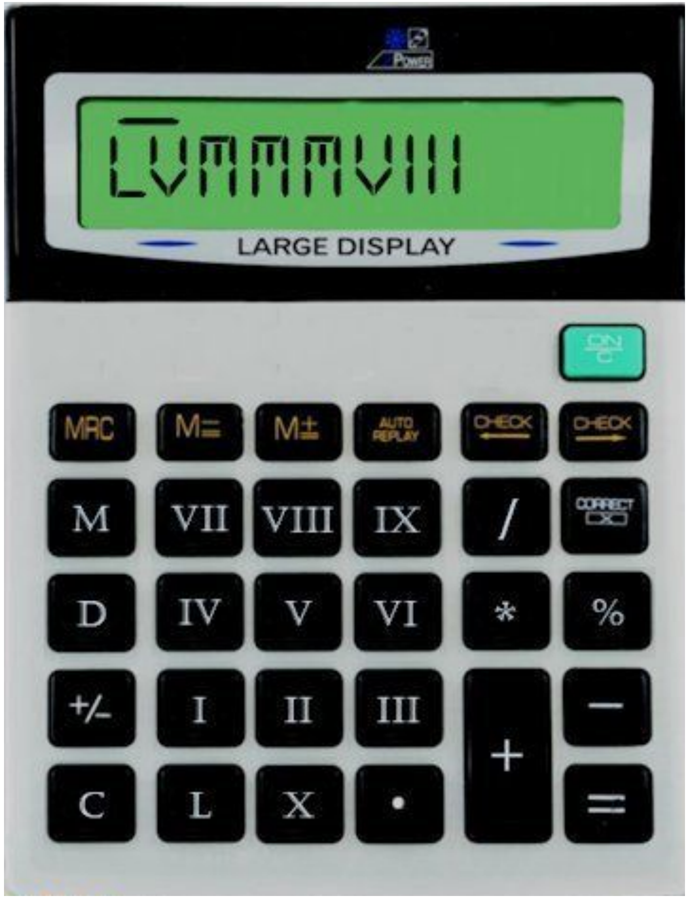
- Use entanglement as communication channel



- Transmit 2 bits (finite amount of information) to teleport 1 qubit (uncountable)

- Pay cost ahead of time

How to use quantum information



Notation

Notation isn't just a way to write ideas.

It stimulates ideas you can have.

Abstraction

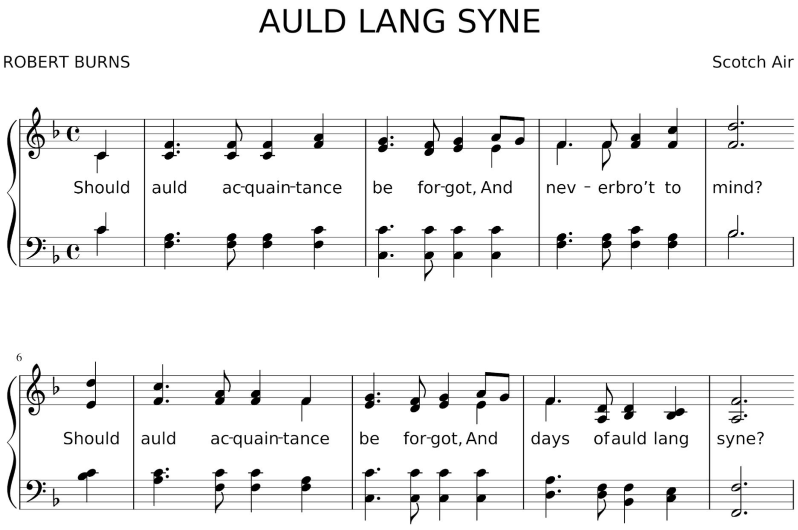
- *Scalable*
- *Automatable*
- *Optimisable*
- *Understandable*

Abstraction

- *Scalable*
- *Automatable*
- *Optimisable*
- *Understandable*

Abstraction

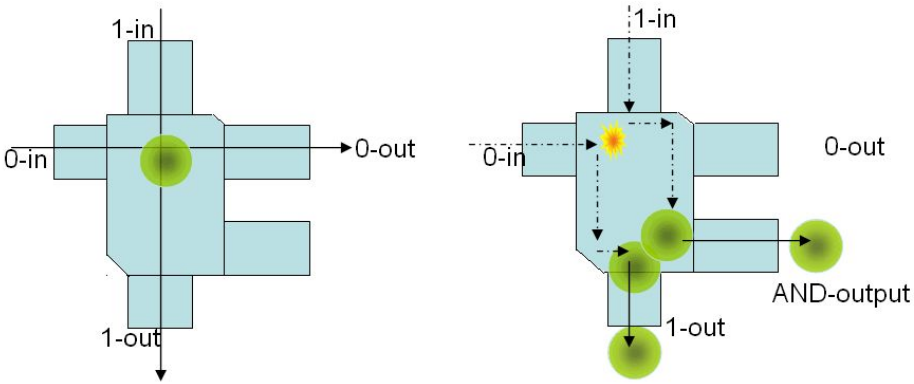
- *Scalable*
- *Automatable*
- *Optimisable*
- *Understandable*

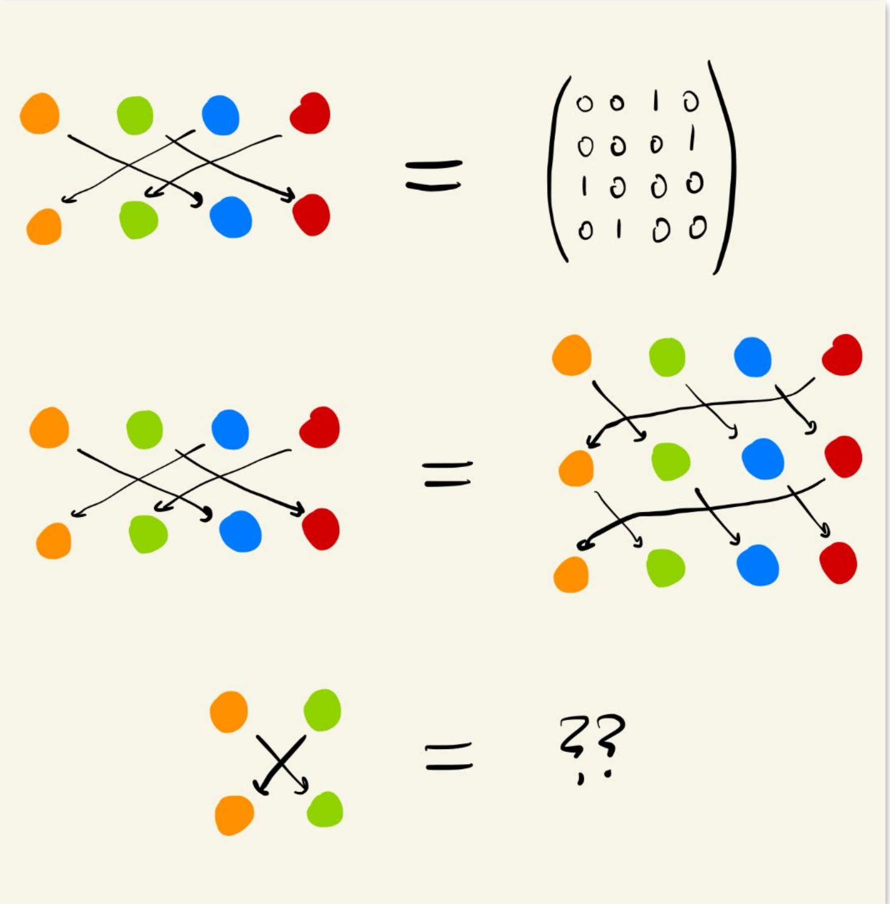


How to use a quantum computer

Billiard Ball Computing

Universal for reversible computation [Fredkin & Toffoli, 1982]





Permutations

Permutation with *n* elements is
n x *n* matrix with boolean entries

Permutations give semantics for
reversible classical programs

Not all polynomials x^2+f have solutions:
not every permutation has a square root

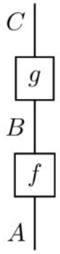
Rig categories

Categorification of natural numbers: finite sets and bijections

$$\frac{f\colon A \to B \qquad g\colon B \to C}{g \circ f\colon A \to C}$$

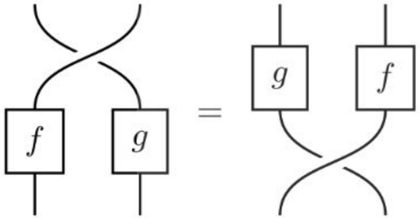
$$\frac{}{\mathrm{id}\colon A \to A}$$

$$(h \circ g) \circ f = h \circ (g \circ f)$$
$$\mathrm{id} \circ f = f = f \circ \mathrm{id}$$



$$\frac{f\colon A \to B \qquad f'\colon A' \to B'}{f \otimes f'\colon A \otimes A' \to B \otimes B'}$$

$$(A \otimes B) \otimes C \simeq A \otimes (B \otimes C)$$
$$I \otimes A \simeq A \simeq A \otimes I$$
$$A \otimes B \simeq B \otimes A$$



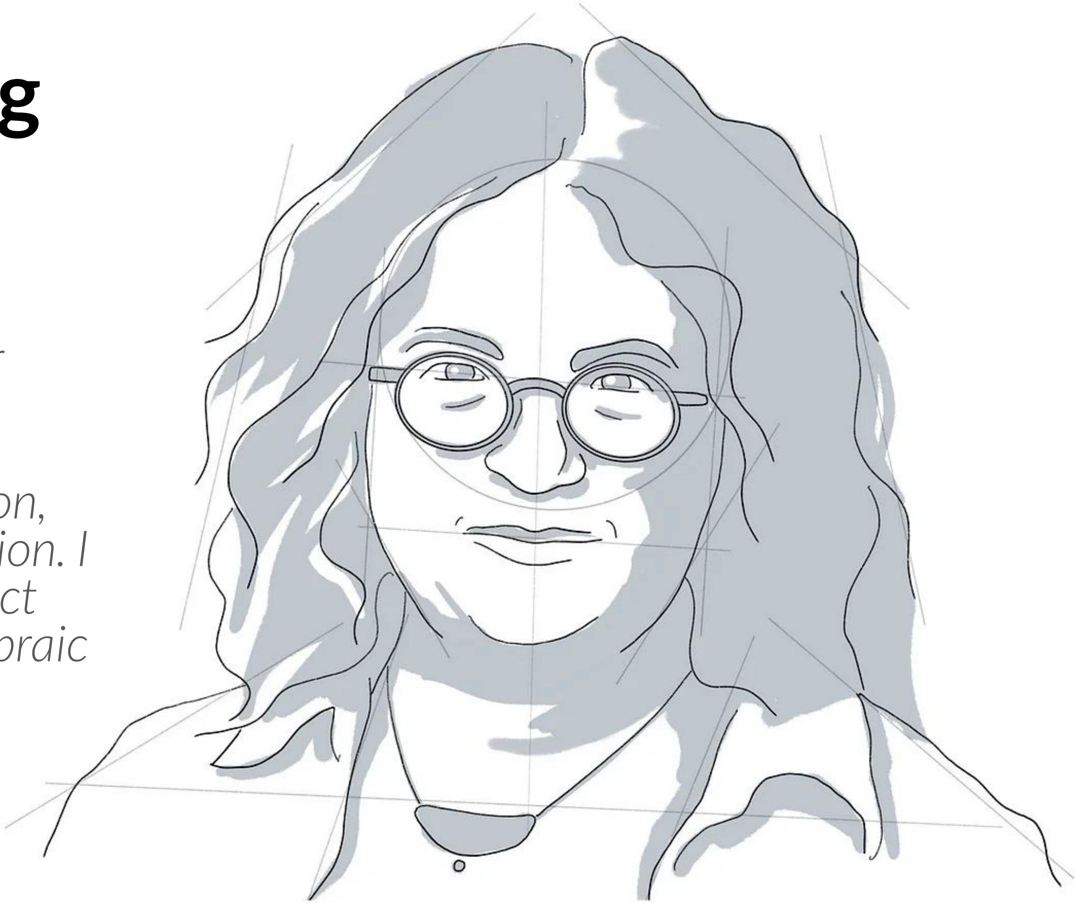
$$\frac{f\colon A \to B \qquad f'\colon A' \to B'}{f \oplus f'\colon A \oplus A' \to B \oplus B'}$$

$$(A \oplus B) \oplus C \simeq A \oplus (B \oplus C)$$
$$O \oplus A \simeq A \simeq A \oplus O$$
$$A \oplus B \simeq B \oplus A$$

$$A \otimes (B \oplus C) \simeq (A \otimes B) \oplus (A \otimes C)$$
$$A \otimes O \simeq O$$

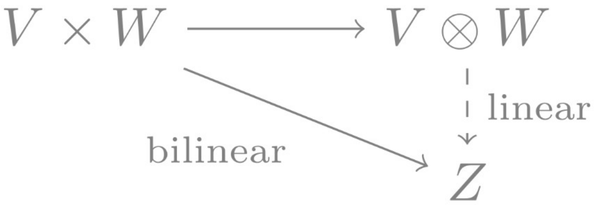
Quantum computing as a completion

"It's really something that is special for quantum computation because it's somehow 'complete' — quantum computation is some kind of completion, mathematically, of classical computation. I think of this as maybe similar to the fact that the complex numbers are an algebraic closure of the real numbers."

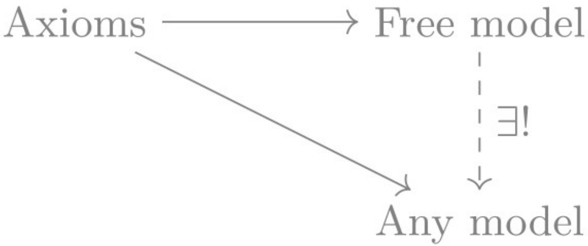


Universality

Categorical universality
characterise property up to isomorphism
by behaviour rather than construction



Finite sets and permutations are initial rig category



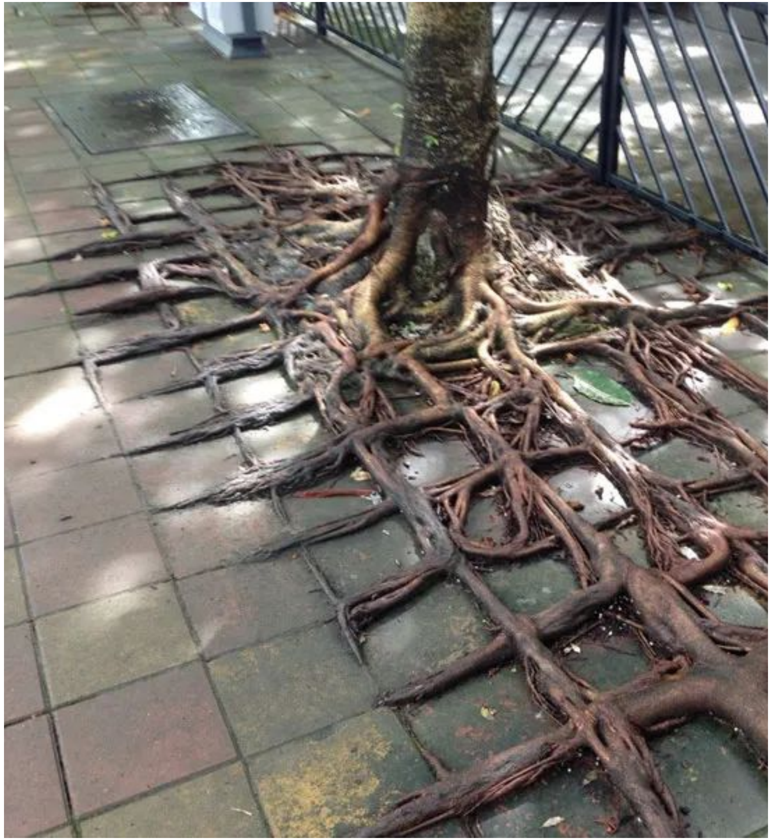
A Few Square Roots

Add two generators

$w : 1 \leftrightarrow 1$
$v : 1+1 \leftrightarrow 1+1$

And impose three equations

v^2 = swap
w^8 = id
vsv = svs



where $s = id+w^2$

Can build Clifford+T: $T = id + w$ $S = id + w^2$ $Z = id + w^4$ $H = w^7 v s v$

Freeq

Theorem: Free model Π exists

Theorem: If $\mathbb{D}$ is dyadic rationals, ζ is 8^{th} root of unity, then Π = Unitary($\mathbb{D}[\zeta]$)

Theorem: There is inclusion $[\![-]\!]\colon \Pi \to$ Unitary($\mathbb{C}$), and it is dense

Theorem: $[\![f]\!]=[\![g]\!]$ iff $(\!(f)\!)=(\!(g)\!)$ for all interpretations $(\!(f)\!)$

Theorem: There is faithful $F(\Pi) \to \mathrm{FCstar}_{cp}$ for universal construction F

Conclusion

- How to use a quantum computer?
  I don’t know.

- If scalable, optimisable, and understandable,
  then programming language must be abstract.

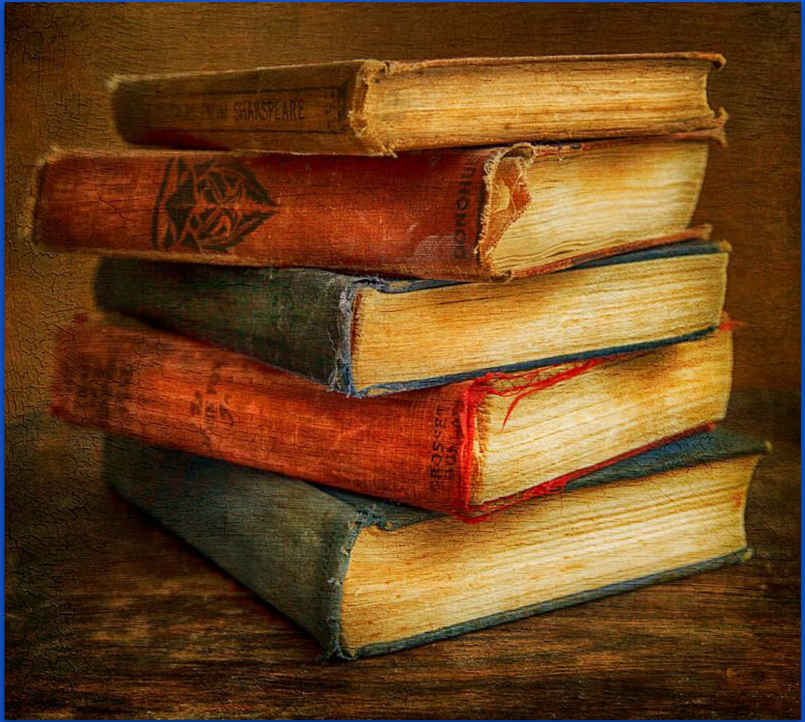
- Universality: no alarms and no surprises.
  Can discover primitives and algorithms.

- Can uncover nature of quantum information.

- Suggestion: rig categories, square roots.
  Next: syntax, optimisation. Hamiltonians.

Many exciting questions for next 100 years!

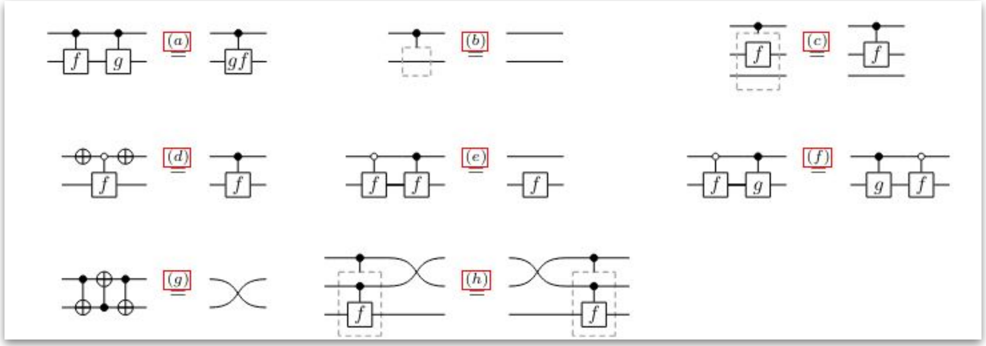
References



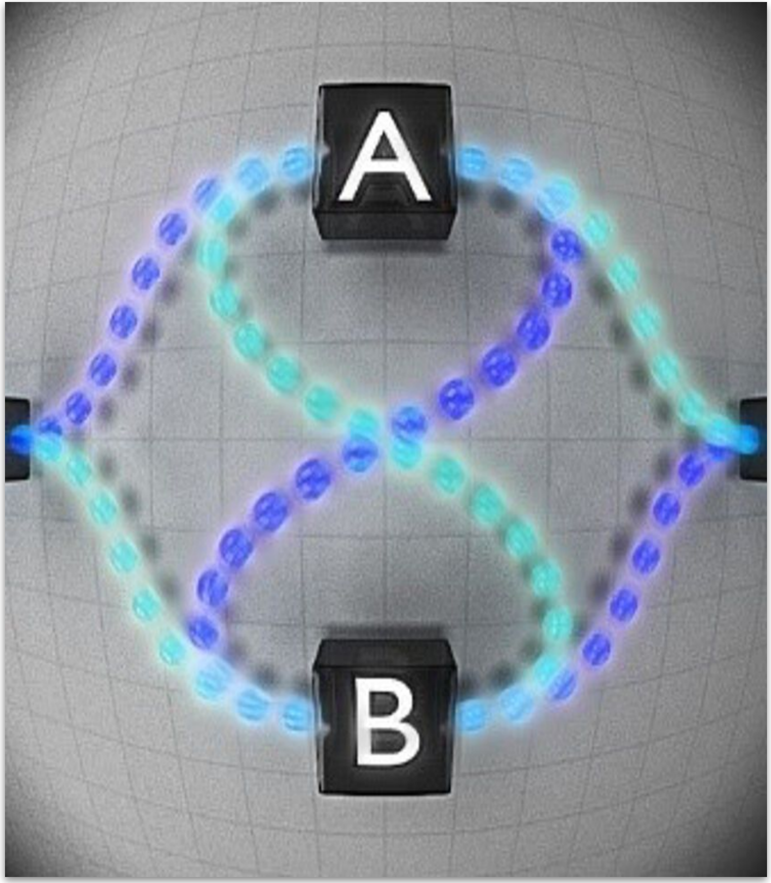
- “Categories for Quantum Theory”
  Oxford University Press, 2019
- “Axioms for the Category of Hilbert Spaces”
  Proceedings of the National Academy of Sciences, 2022
- “Weakly Measured While Loops: Peaking at Quantum States”
  Quantum Science and Technology, 2022
- “With a Few Square Roots, Quantum Computing is as Easy as Pi”
  Principles of Programming Languages, 2024
- “Qurts: Automatic Quantum Uncomputation by Affine Types with Lifetimes”
  Principles of Programming Languages, 2025
- “Quantum Circuits are Just a Phase”
  Principles of Programming Languages, 2026
- “One Rig to Control Them All”
  arXiv, 2026

Taking back control

- Start with any ‘circuit theory’
- Add control



- Get complete, structured language



It’s just a phase

- Many (most?) quantum algorithms come down to eigenspace decomposition and eigenvector manipulation
- Lift to primitive
- Universal

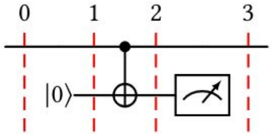
$$
\begin{aligned}
\mathsf{Z} &:= \text{if let } |1\rangle \text{ then } \mathsf{Ph}(\pi) & \mathsf{X} &:= \text{if let } |-\rangle \text{ then } \mathsf{Ph}(\pi) \\
\mathsf{T} &:= \text{if let } |1\rangle \text{ then } \mathsf{Ph}(\pi/4) & \mathsf{Y} &:= \text{if let } \mathsf{S} \cdot |-\rangle \text{ then } \mathsf{Ph}(\pi) \\
\mathsf{H} &:= \text{if let } \mathsf{Y}^{1/4} \cdot |1\rangle \text{ then } \mathsf{Ph}(\pi) & \mathsf{CX} &:= \text{if let } |1\rangle \otimes \mathsf{id}_1 \text{ then } \mathsf{X}
\end{aligned}
$$

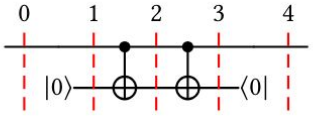
- Grover now one-liner:

$$\text{if let } |\omega_1\rangle \otimes \cdots \otimes |\omega_n\rangle \text{ then } \mathsf{Ph}(\pi)$$

More ancillae, more problems

- Cannot reuse dirty qubits



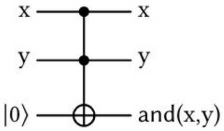


- Rust type system: ownership, borrowing

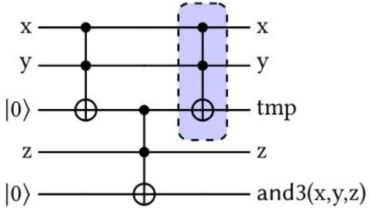
```
fn main() {
  let x = [1,2,3];
  let y = f(x); // value of x is moved
  print!("{x},{y}"); // so x lost access
}
fn f<T>(x : T) -> T {
  x
}
```

```
fn main() {
  let x = [1,2,3];
  let y = f(&x); // x is borrowed
  print!("{x},{y}"); // x can be read
}
fn f<T>(x : T) -> T {
  x
}
```

- Compiler automatically inserts uncomputation




```
fn and3<'a>(
  x:&'a qbit,
  y:&'a qbit,
  z:&'a qbit
) -> #'a qbit {
  let tmp = and(x,y);
  let ref = &tmp;
  and(ref,z)
}
```



Becoming measured

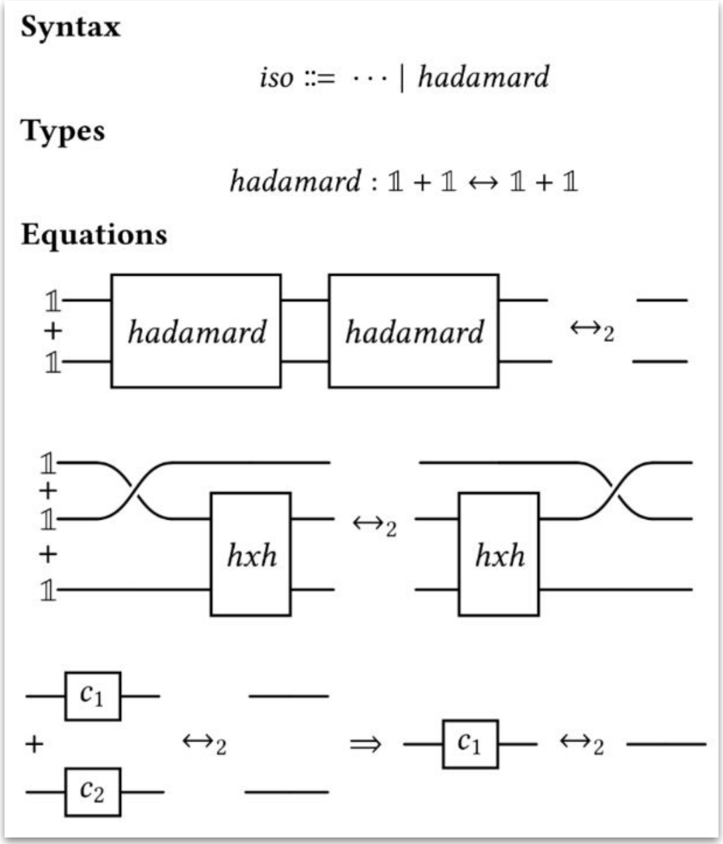
- classical computation
  = classical reversible computation
  + information effects

- Can copy and delete classical bits with

  erase : b $\rightsquigarrow$ 1
  create : 1 $\rightsquigarrow$ b

- Dynamic quantum programming language

- Compiles measurements anywhere to single standard measurement at end



Programming with equations

- Word problem decidable
- Normalisation by evaluation
- Fewer generators than Π
- **Theorem (completeness):** $[\![f]\!]=[\![g]\!]$ iff $(\!(f)\!)=(\!(g)\!)$

Orthogonal group presentations

Proposition: there is finite presentation for unitary matrices $O_n(\mathbb{Z}[1/\sqrt{2}])$

$$(-1)^2_{[a]} \approx \varepsilon$$

$$X^2_{[a,b]} \approx \varepsilon$$

$$(-1)_{[a]}(-1)_{[b]} \approx (-1)_{[b]}(-1)_{[a]}$$

$$(-1)_{[a]}X_{[b,c]} \approx X_{[b,c]}(-1)_{[a]}$$

$$X_{[a,b]}X_{[c,d]} \approx X_{[c,d]}X_{[a,b]}$$

$$(-1)_{[a]}X_{[a,b]} \approx X_{[a,b]}(-1)_{[b]}$$

$$X_{[b,c]}X_{[a,b]} \approx X_{[a,b]}X_{[a,c]}$$

$$X_{[a,c]}X_{[b,c]} \approx X_{[b,c]}X_{[a,b]}$$

$$H^2_{[a,b]} \approx \varepsilon$$

$$(-1)_{[a]}H_{[b,c]} \approx H_{[b,c]}(-1)_{[a]}$$

$$X_{[a,b]}H_{[c,d]} \approx H_{[c,d]}X_{[a,b]}$$

$$H_{[a,b]}H_{[c,d]} \approx H_{[c,d]}H_{[a,b]}$$

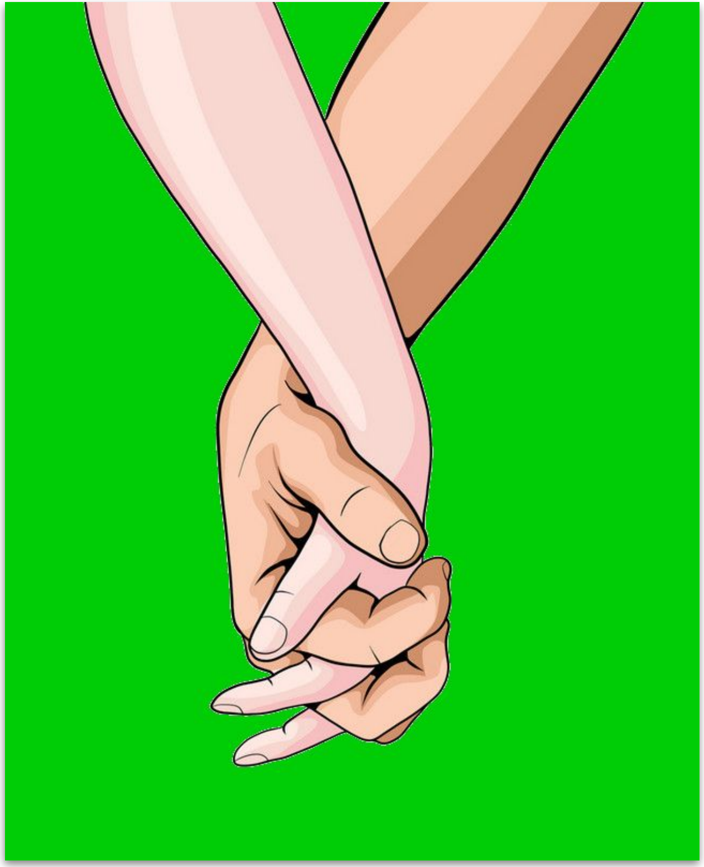
$$H_{[b,c]}X_{[a,b]} \approx X_{[a,b]}H_{[a,c]}$$

$$H_{[a,c]}X_{[b,c]} \approx X_{[b,c]}H_{[a,b]}$$

$$(-1)_{[a]}(-1)_{[b]}H_{[a,b]} \approx H_{[a,b]}(-1)_{[a]}(-1)_{[b]}$$

$$(-1)_{[b]}H_{[a,b]} \approx H_{[a,b]}X_{[a,b]}$$

$$X_{[b,c]}H_{[a,b]}X_{[a,c]}H_{[a,d]}H_{[a,b]}X_{[a,c]}H_{[a,d]} \approx H_{[a,b]}X_{[a,c]}H_{[a,d]}H_{[a,b]}X_{[a,c]}H_{[a,d]}X_{[c,d]}$$



Bit commitment

- Alice commits to value hidden from Bob
- Cryptographic primitive, essential in
  - zero-knowledge proofs
  - secret sharing
  - secure multi-party computation
- Impossible with quantum values