Finding the true flow in measurement-based quantum computation

Ross Duncan

Laboratoire d'Informatique Quantique
Université Libre de Bruxelles

Joint work with

Simon Perdrix

CNRS, LIG,
Université de Grenoble

Duncan and Perdrix. "*Rewriting measurement-based quantum computations with generalised flow*". ICALP 2010, Part II, LNCS 6199

What is the point?

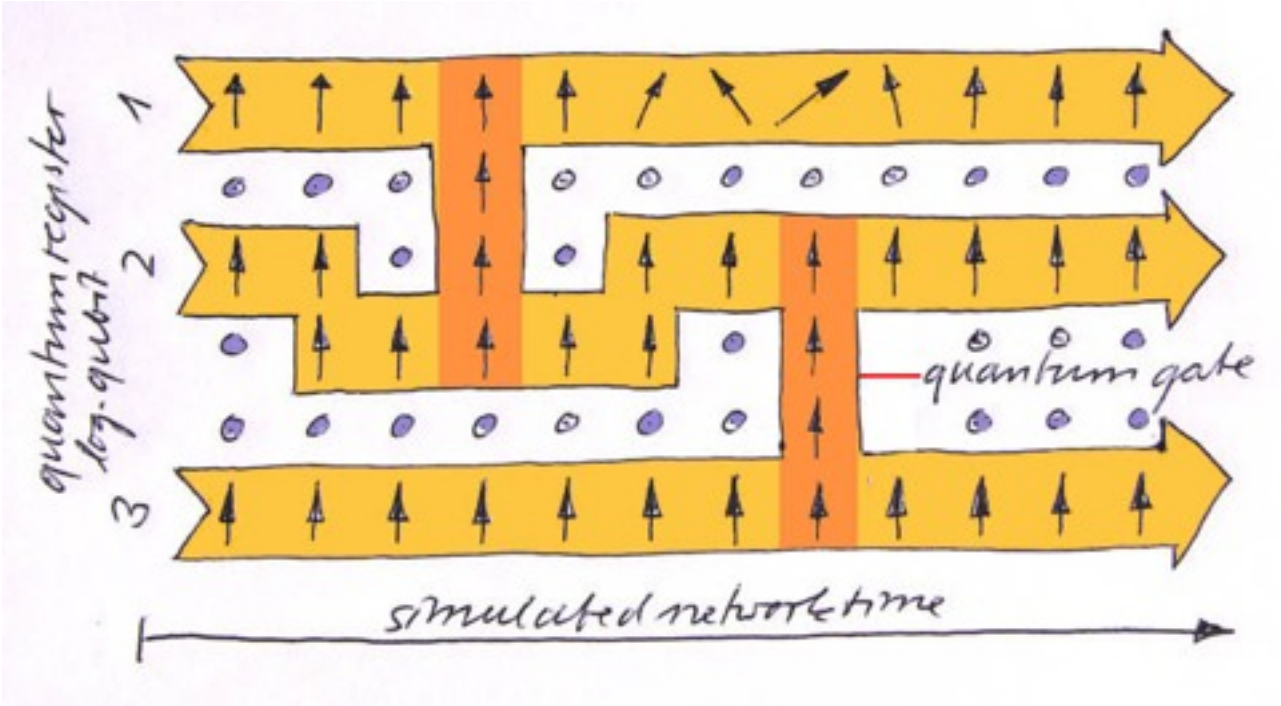
Measurement-based quantum computation

- compute by *measuring* a large entangled state
- can it be done deterministically?
- is the given program correct?

Graphical Rewriting

- translate MBQC into diagrammatic syntax
- equational theory based on high-level properties of QM
- rewrite graphs to translate back to quantum circuits

Information Flow in the 1-Way Quantum Computer



Using measurements to compute

Quantum Measurements

We can measure the state of qubit $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \qquad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$\begin{tikzcd}
 & \alpha\,|0\rangle + \beta\,|1\rangle \arrow[dl, "p=\alpha^2"'] \arrow[dl, "p=\beta^2"] \arrow[dr, "p=(\alpha+\beta)/2^2"] \arrow[dr, "p=(\alpha-\beta)/2^2"'] & \\
|0\rangle & |1\rangle & |+\rangle & |-\rangle
\end{tikzcd}$$

Quantum Measurements

We can measure the state of qubit $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$

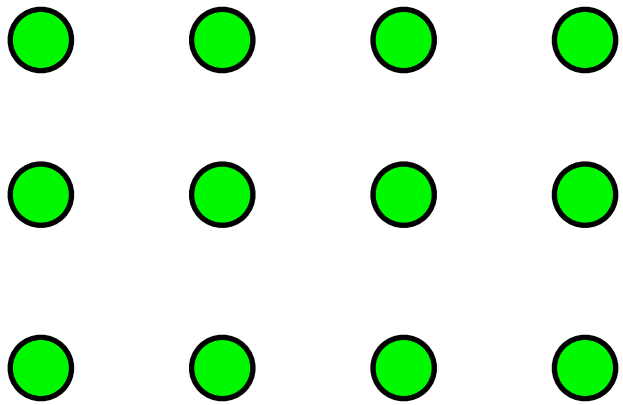
$$Z = \{|0\rangle\,, |1\rangle\} \qquad\qquad X = \{\frac{|0\rangle + |1\rangle}{\sqrt{2}}, \frac{|0\rangle + |1\rangle}{\sqrt{2}}\}$$

$$\begin{tikzcd}
 & \alpha\,|0\rangle + \beta\,|1\rangle \arrow[dl, "p=\alpha^2"'] \arrow[dl, "p=\beta^2"] \arrow[dr, "p=(\alpha+\beta)/2^2"] \arrow[dr, "p=(\alpha-\beta)/2^2"'] & \\
|0\rangle & |1\rangle & |+\rangle & |-\rangle
\end{tikzcd}$$

Raussendorf and Briegel, PRL (86) 2001
Raussendorf, Browne and Briegel, J. Mod. Opt. (46) 2003

The One-Way Model

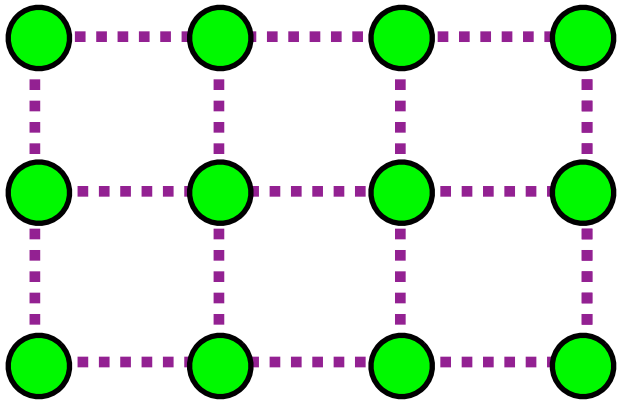
Some qubits, each initialised in the state $|+\rangle := \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$



Raussendorf and Briegel, PRL (86) 2001
Raussendorf, Browne and Briegel, J. Mod. Opt. (46) 2003

The One-Way Model

Some qubits, each initialised in the state $|+\rangle := \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$



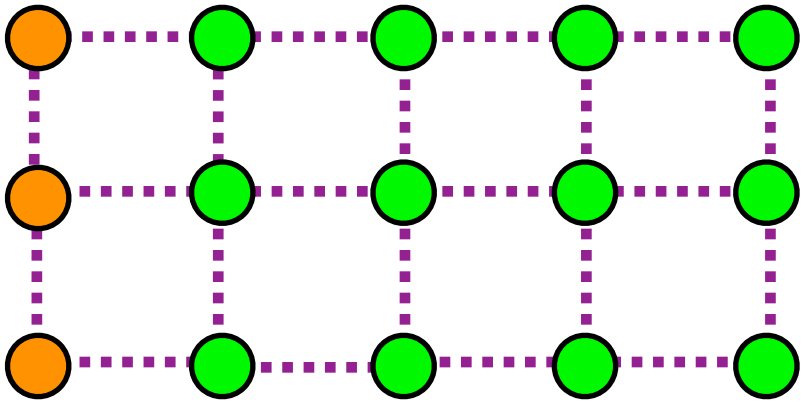
Entangling operation on pairs of qubits to make a **graph state.**

$$CZ = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

Raussendorf and Briegel. PRL (86) 2001
Raussendorf, Browne, Briegel J. Mod. Opt. (46) 2003

The One-Way Model

A graph state, coupled to some input qubits:



The only operation is to measure single qubits in the basis:

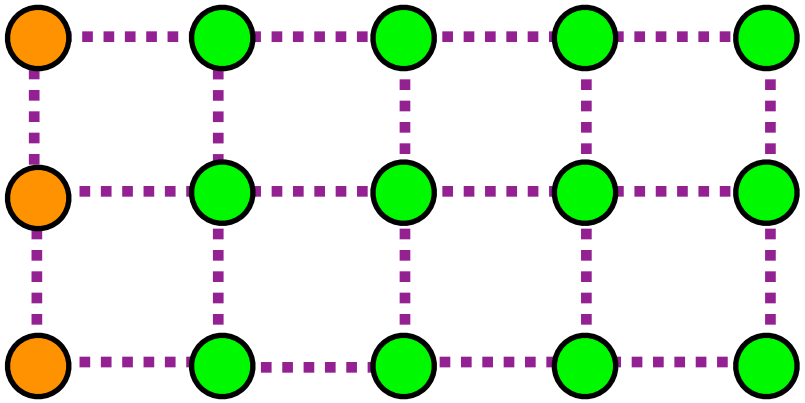
$$|+_\alpha\rangle := \frac{1}{\sqrt{2}}(|0\rangle + e^{i\alpha}|1\rangle)$$

$$|-_\alpha\rangle := \frac{1}{\sqrt{2}}(|0\rangle - e^{i\alpha}|1\rangle)$$

Raussendorf and Briegel. PRL (86) 2001
Raussendorf, Browne, Briegel J. Mod. Opt. (46) 2003

The One-Way Model

A graph state, coupled to some input qubits:

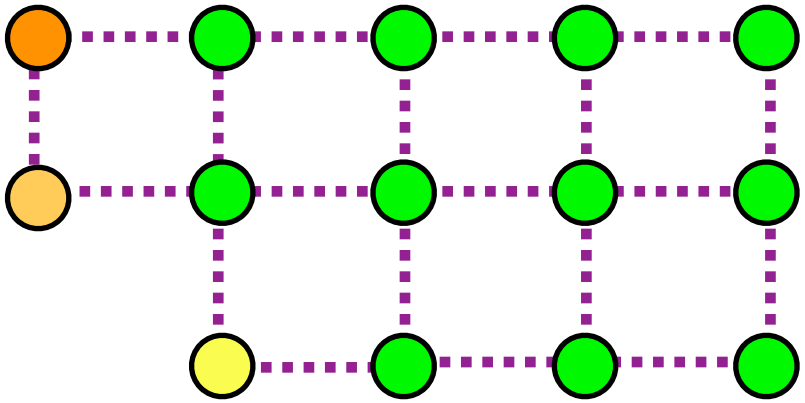


* Measured qubits are removed from the cluster;
* The outcome of measurement alters the remaining state.

Raussendorf and Briegel. PRL (86) 2001
Raussendorf, Browne, Briegel J. Mod. Opt. (46) 2003

The One-Way Model

A graph state, coupled to some input qubits:

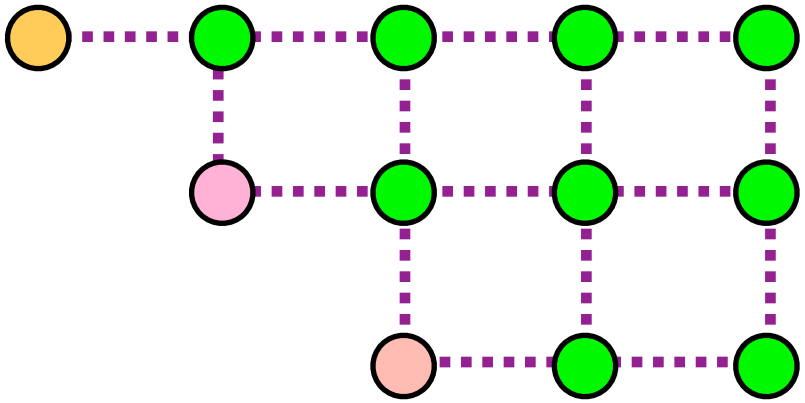


* Measured qubits are removed from the cluster;
* The outcome of measurement alters the remaining state.

Raussendorf and Briegel. PRL (86) 2001
Raussendorf, Browne, Briegel J. Mod. Opt. (46) 2003

The One-Way Model

A graph state, coupled to some input qubits:



* Measured qubits are removed from the cluster;
* The outcome of measurement alters the remaining state.

Raussendorf and Briegel. PRL (86) 2001
Raussendorf, Browne, Briegel J. Mod. Opt. (46) 2003

The One-Way Model

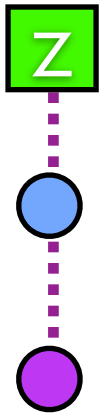
A graph state, coupled to some input qubits:

* Measured qubits are removed from the cluster;
* The outcome of measurement alters the remaining state.

Raussendorf and Briegel. PRL (86) 2001
Raussendorf, Browne, Briegel J. Mod. Opt. (46) 2003

The One-Way Model

A graph state, coupled to some input qubits:



* Measured qubits are removed from the cluster;
* The outcome of measurement alters the remaining state.

Finally, any output qubits, they can be corrected by a Pauli

Raussendorf and Briegel. PRL (86) 2001
Raussendorf, Browne, Briegel J. Mod. Opt. (46) 2003

The One-Way Model

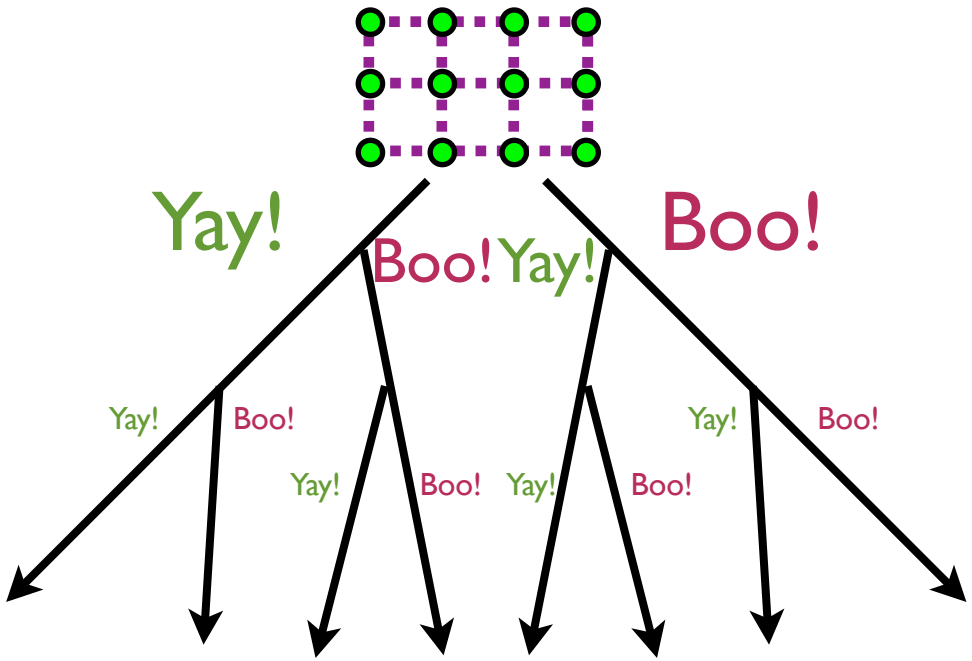
A graph state, coupled to some input qubits:

* Measured qubits are removed from the cluster;
* The outcome of measurement alters the remaining state.

Finally, any output qubits, they can be corrected by a Pauli

Non-determinism

Non-determinism of measurements leads to probabilistic branching

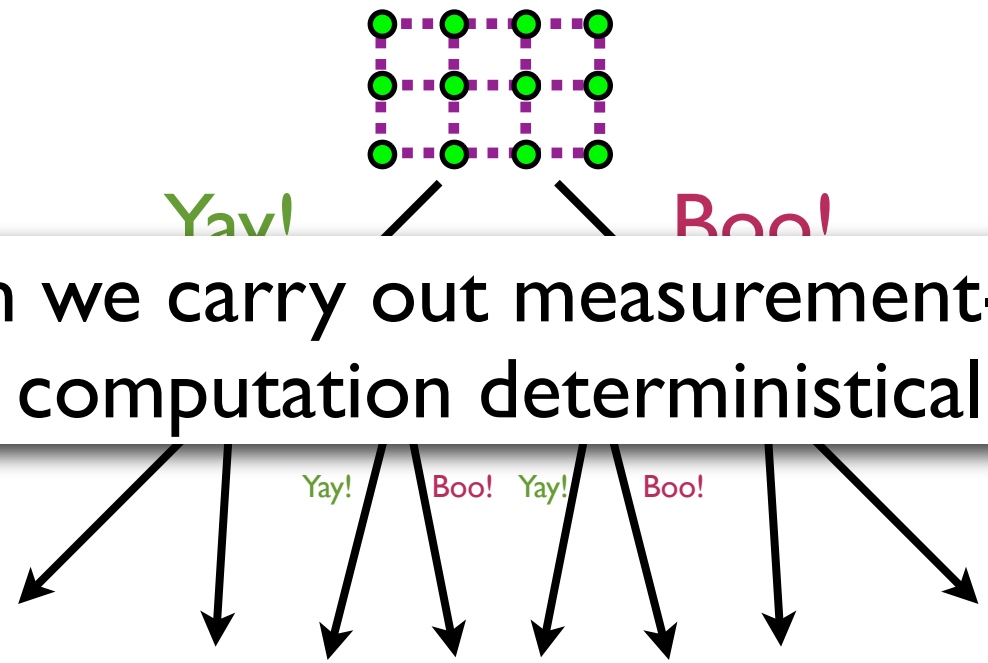


Attempt to control branching by using *adaptive measurements*:

(choice of measurements depend on the outcome of earlier ones)

Non-determinism

Non-determinism of measurements leads to probabilistic branching



Can we carry out measurement-based computation deterministically?

Attempt to control branching by using *adaptive measurements*:

(choice of measurements depend on the outcome of earlier ones)

Measurement Calculus

The measurement calculus is a formal syntax for programming the one-way model.

- N_i : initialise qubit i in the $|+\rangle$ state
- E_{ij}: entangle qubits i and j using a CZ operation
- ${}^s[M_i^\alpha]^t$: measure qubit i in the basis $|0\rangle \pm e^{(-1)^s\alpha + t\pi}\,|1\rangle$
- X_i^s, Z_i^s : apply a Pauli X or Z operator to qubit i

The boolean variables s, t are *signals*; their values are determined by the results of measuring the corresponding qubit.

- Standard form: operations occur in the order above
  - Every pattern can be standardised

Geometry of a pattern

The following pattern computes the CNOT gate:

$$\mathfrak{P} = \bar{X}_4^3 Z_4^2 Z_1^2 M_3^0 M_2^0 \bar{E}_{13} E_{23} E_{34} N_3 N_4$$

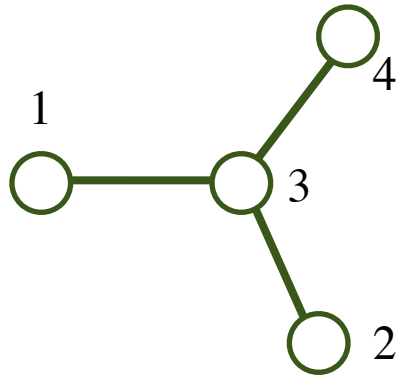
Each pattern implicitly defines a *geometry* (G, I, O)

Geometry of a pattern

The following pattern computes the CNOT gate:

$$\mathfrak{P} = \bar{X}_4^3 Z_4^2 Z_1^2 M_3^0 M_2^0 \boxed{E_{13} E_{23} E_{34}} N_3 N_4$$

Each pattern implicitly defines a *geometry* (G, I, O)



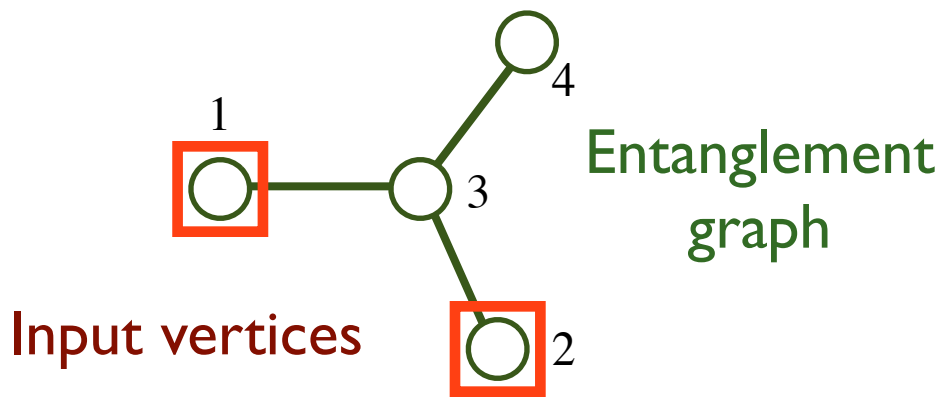
Entanglement graph

Geometry of a pattern

The following pattern computes the CNOT gate:

$$\mathfrak{P} = \bar{X}_4^3 Z_4^2 Z_1^2 M_3^0 M_2^0 E_{13} E_{23} E_{34} N_3 N_4$$

Each pattern implicitly defines a *geometry* (G, I, O)

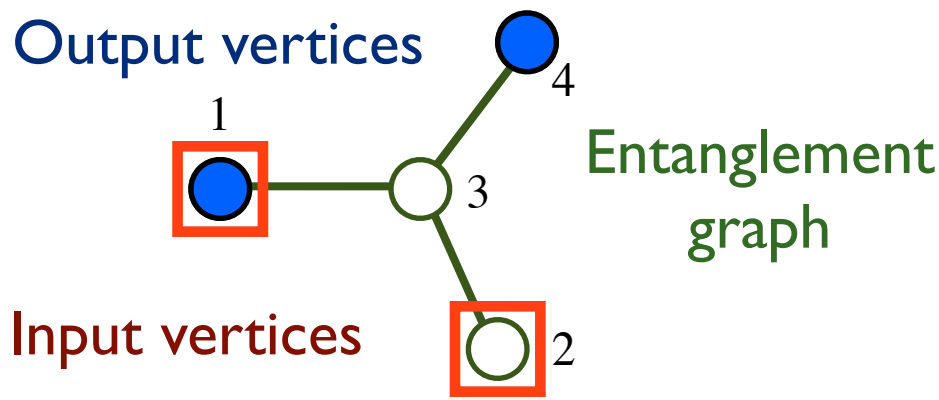


Geometry of a pattern

The following pattern computes the CNOT gate:

$$\mathfrak{P} = \bar{X}_4^3 Z_4^2 Z_1^2 \boxed{M_3^0 M_2^0} \boxed{E_{13} E_{23} E_{34}} \boxed{N_3 N_4}$$

Each pattern implicitly defines a *geometry* (G, I, O)



Flow and determinism

Defn: let (G, I, O) be a geometry; a *flow* on G is a pair $(f, <)$ where $f : O^C \to I^C$ is a function and $<$ is a partial order s.t.

- $f(u) \sim u$
- $u < f(u)$
- If $f(u) \sim v, v \neq u$ then $u < v$

Intuitively f(u) is the qubit which must correct the error produced by measuring u. The partial order guarantees that this is consistent with causality.

Thm: if a geometry has a flow there exists a uniformly deterministic pattern on it.

How to Measure

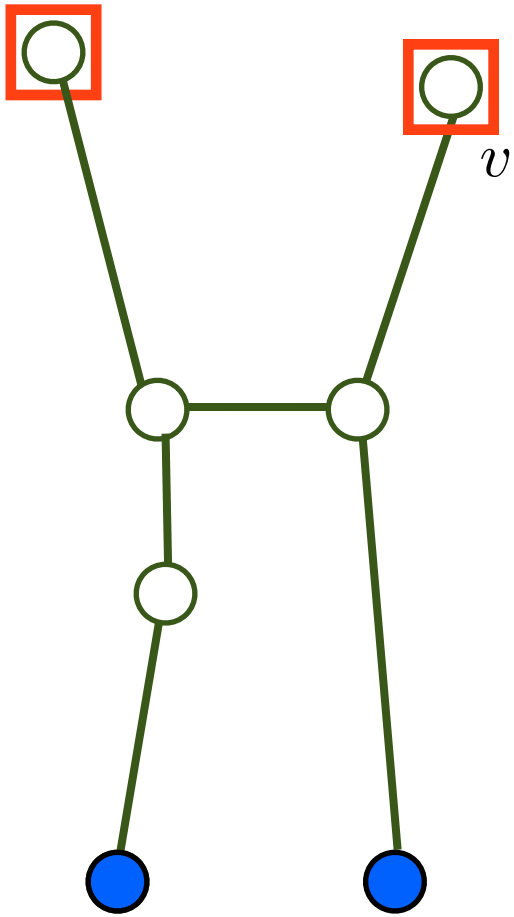
Suppose we have a measuring device for the standard 1-qubit basis:

How to Measure

Suppose we have a measuring device for the standard 1-qubit basis:

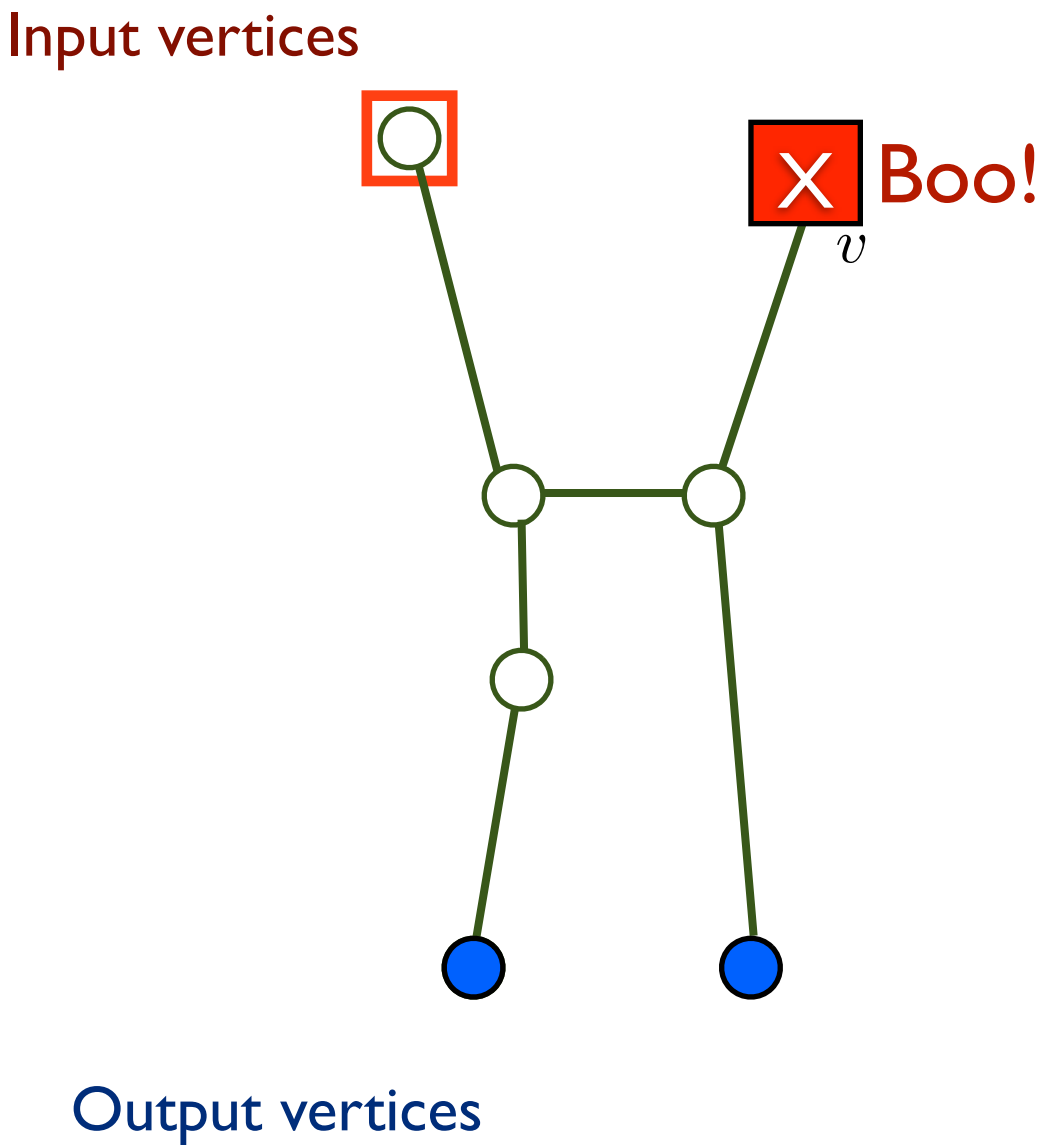
Flow

Input vertices

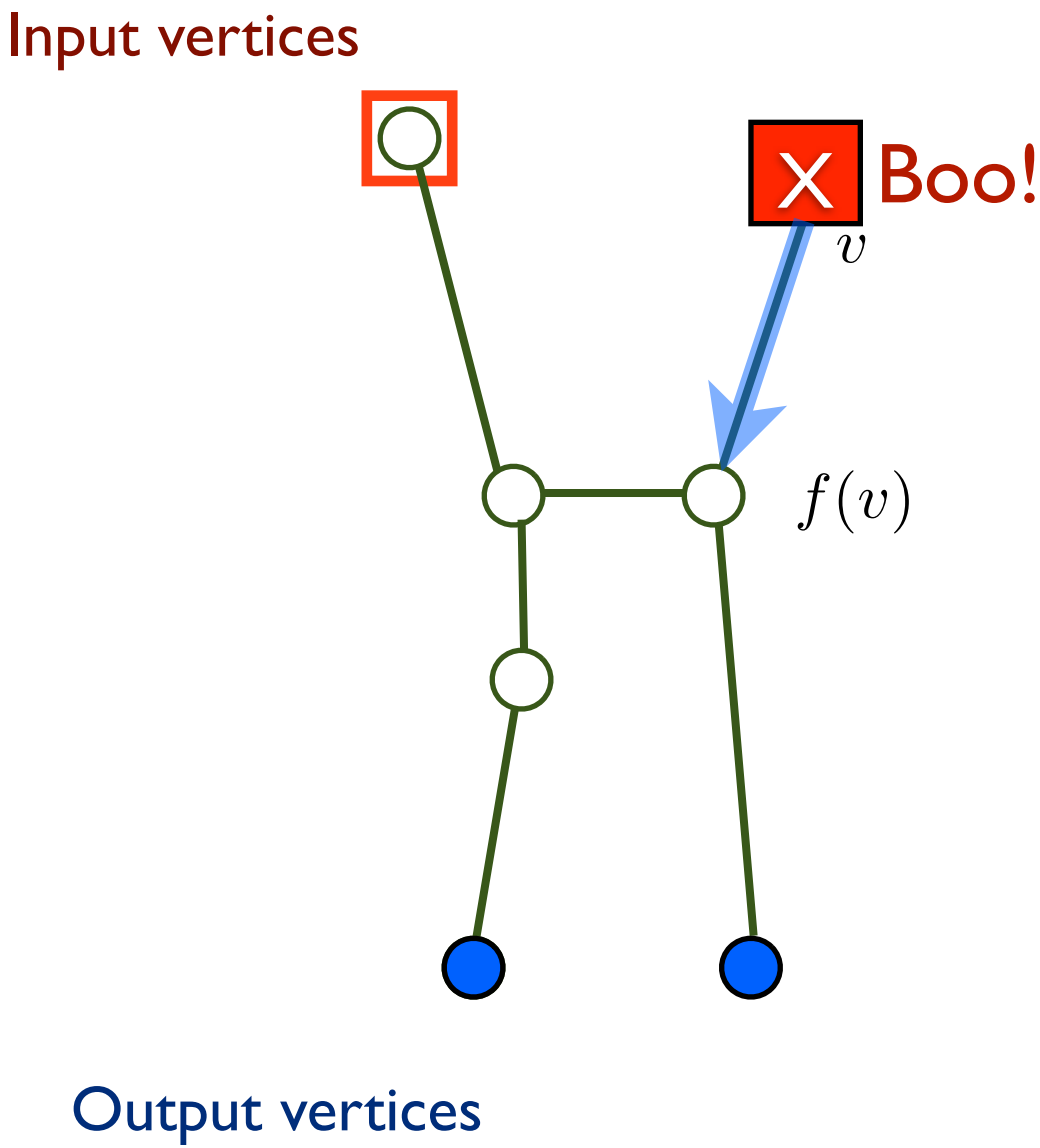


Output vertices

Flow

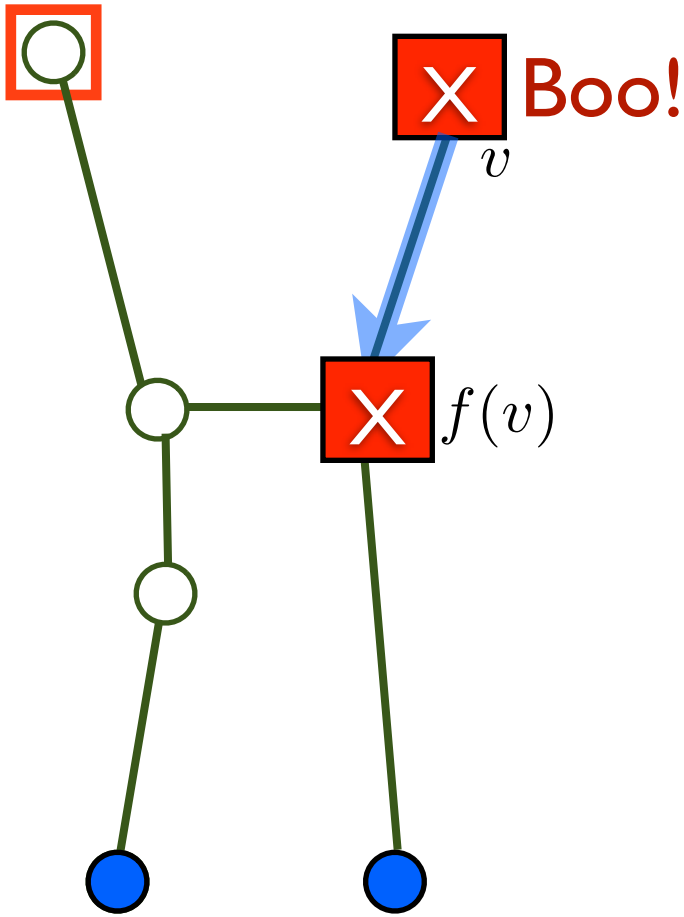


Flow



Flow

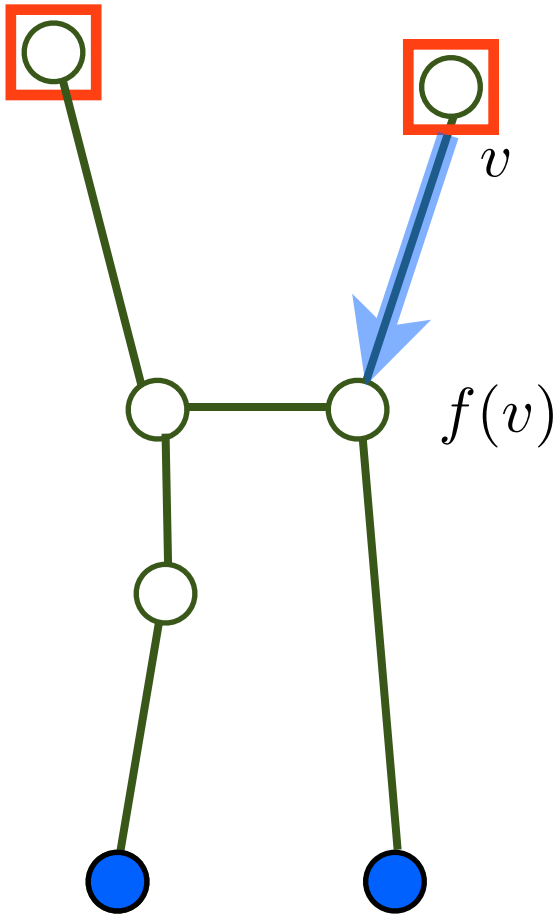
Input vertices



Output vertices

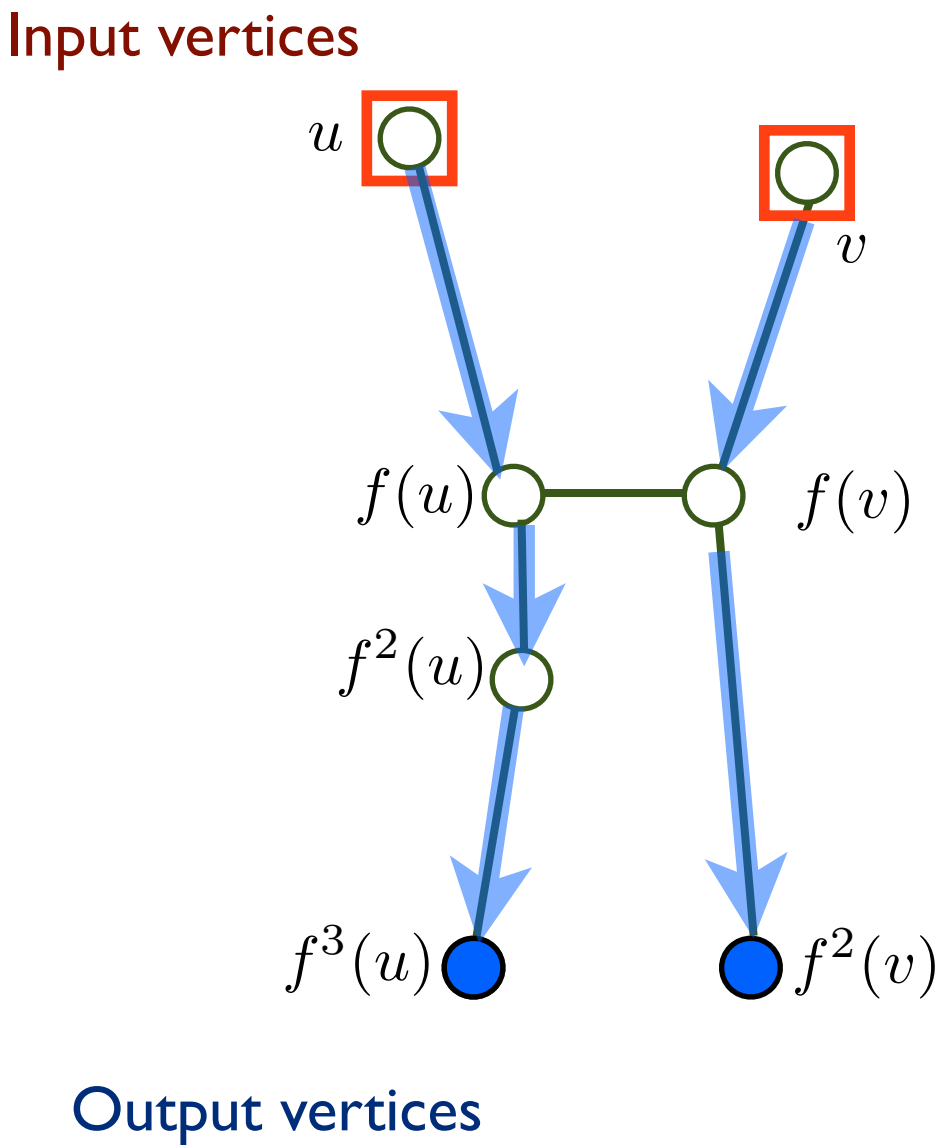
Flow

Input vertices



Output vertices

Flow



The sequence

$$u, f(u), f^2(u), \ldots$$

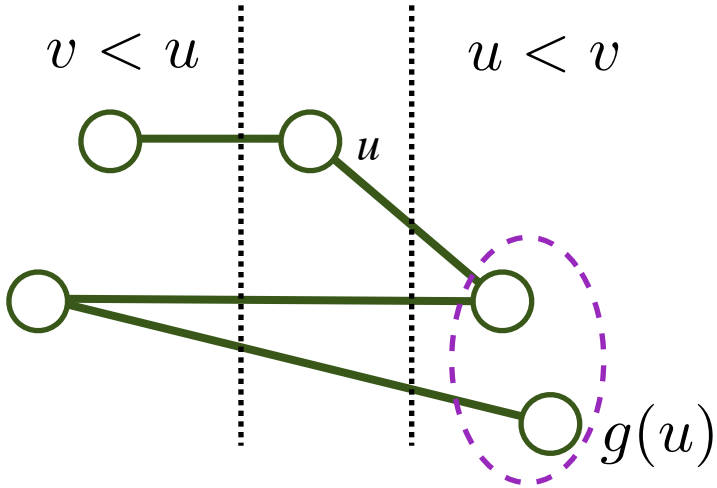
determines a path from inputs to outputs.

Effectively the flow path" is a "logical qubit" in a circuit equivalent to the pattern.

GFlow and determinism

Defn: let (G, I, O) be an open graph; a *gflow* on G is a pair $(g, <)$ where $g : O^C \to \mathcal{P}(I^C)$ is a function and $<$ is a partial order s.t.

- If $v \in g(u)$ then $u < v$
- $u \in \mathrm{Odd}(g(u))$
- If $v \in \mathrm{Odd}(g(u)) \setminus \{u\}$ then $u < v$



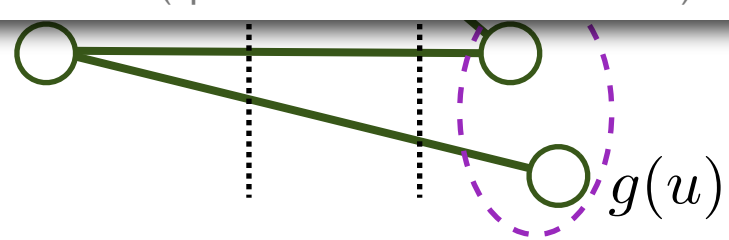
GFlow and determinism

Defn: let (G, I, O) be an open graph; a *gflow* on G is a pair $(g, <)$ where $g : O^C \to \mathcal{P}(I^C)$ is a function and $<$ is a partial order s.t.

- If $v \in g(u)$ then $u < v$
- $u \in \mathrm{Odd}(g(u))$
- If $v \in \mathrm{Odd}(g(u)) \setminus \{u\}$ then $u < v$

$v < u$ $\qquad$ $u < v$

Thm: a geometry has a gflow *if and only if* there exists a uniformly deterministic pattern on it. (upto some minor technical caveat)



Determinism in MBQC

Measurement
Pattern
"low-level program"

Determinism in MBQC

Measurement Pattern

"low-level program"

implicitly defines

Geometry

"entangled resource"

"graph state"

Determinism in MBQC

Measurement Pattern

"low-level program"

implicitly defines

Geometry

"entangled resource"

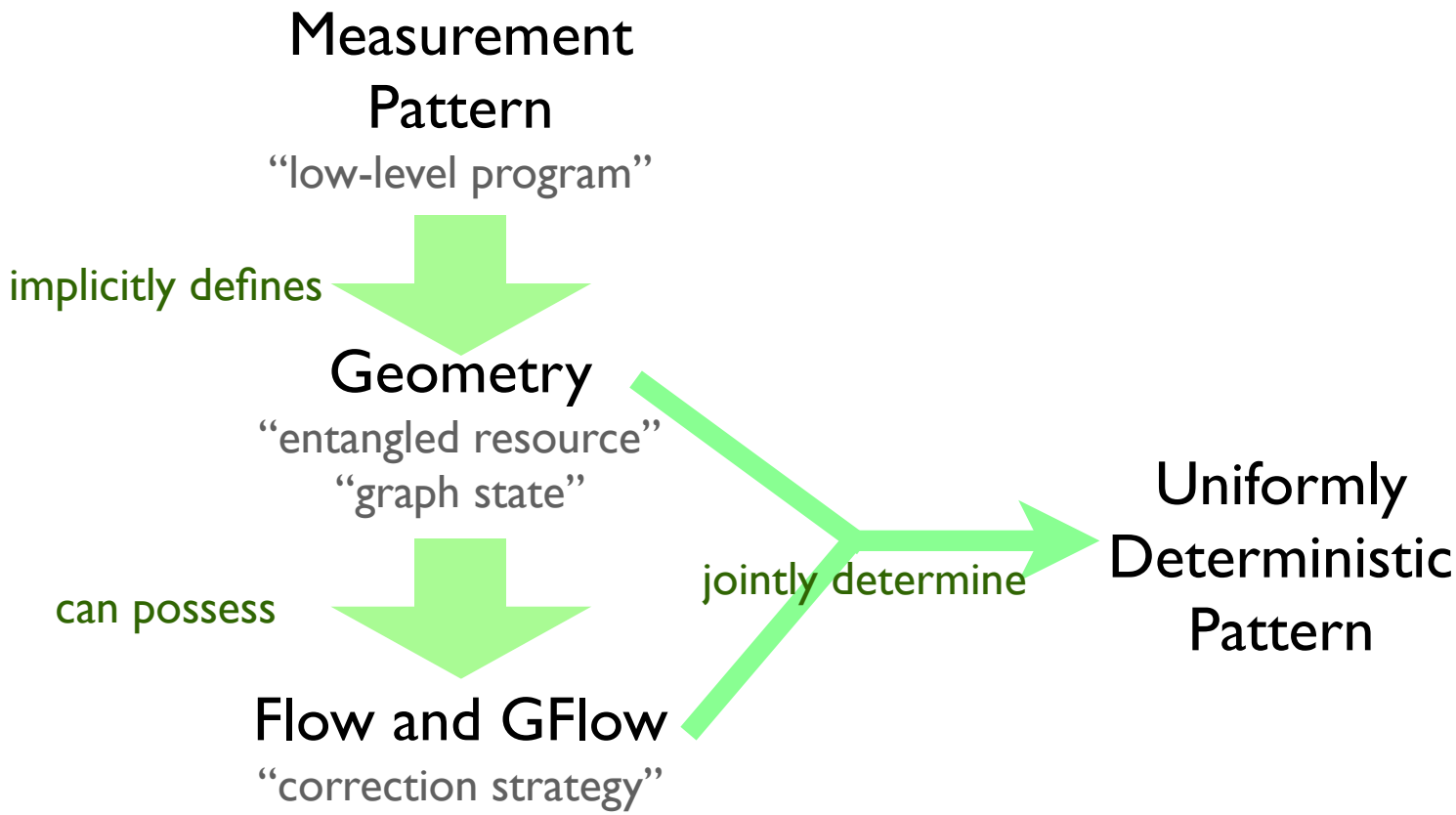
"graph state"

can possess

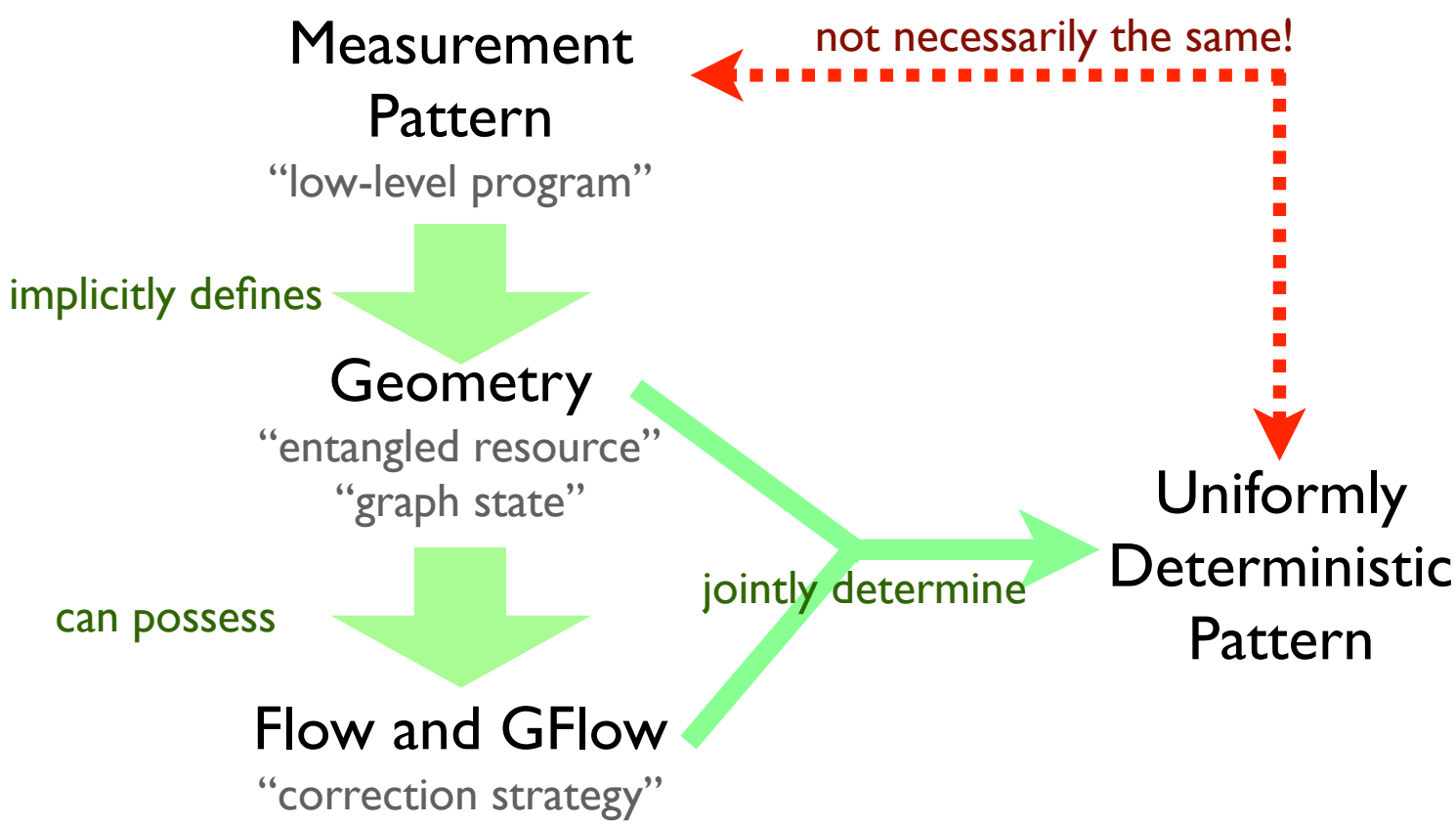
Flow and GFlow

"correction strategy"

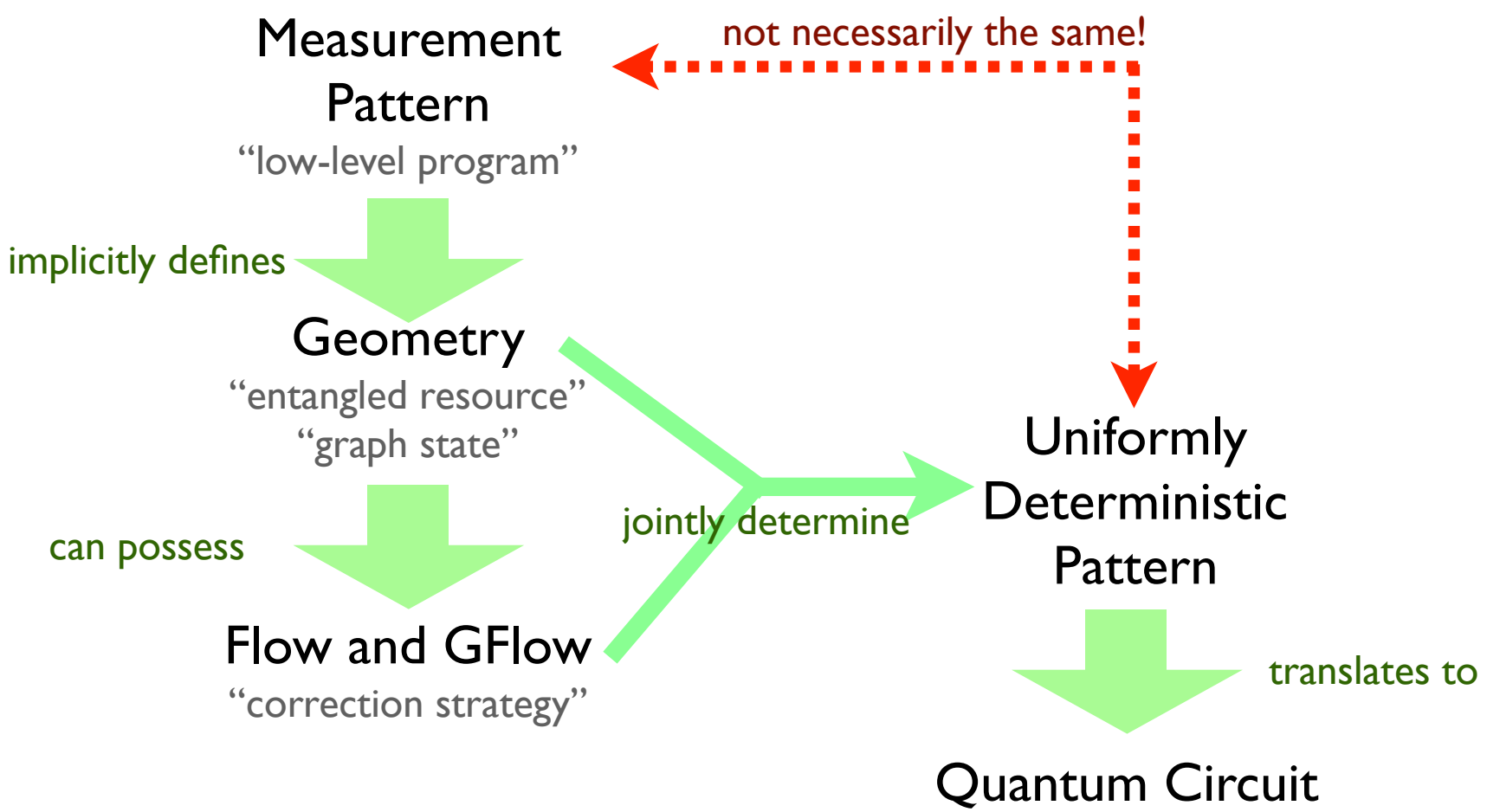
Determinism in MBQC



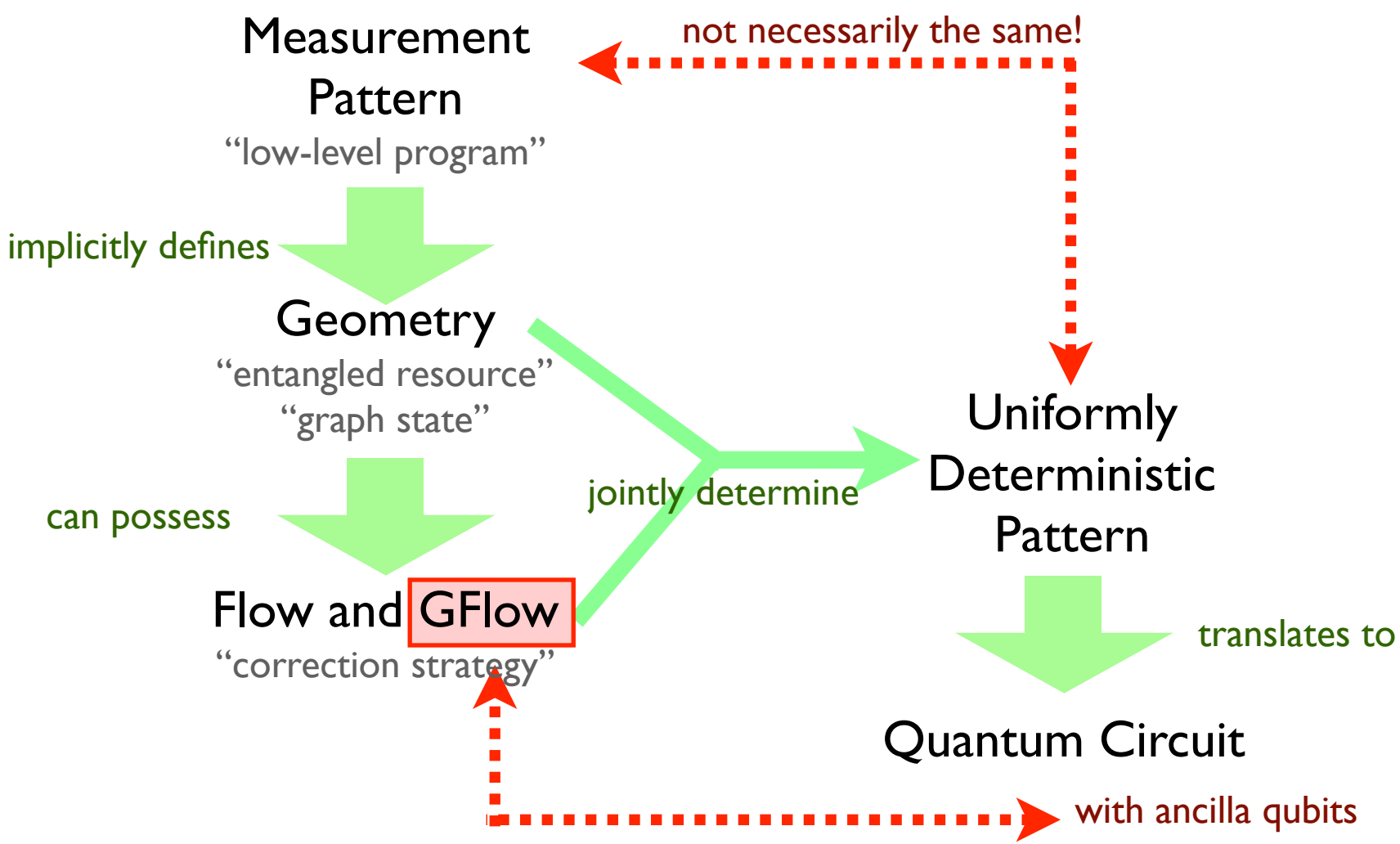
Determinism in MBQC



Determinism in MBQC



Determinism in MBQC



Summary of this talk:

Measurement Pattern

"low-level program"

Geometry

"entangled resource"

"graph state"

Flow and GFlow

"correction strategy"

Summary of this talk:

Measurement Pattern

"low-level program"

Geometry

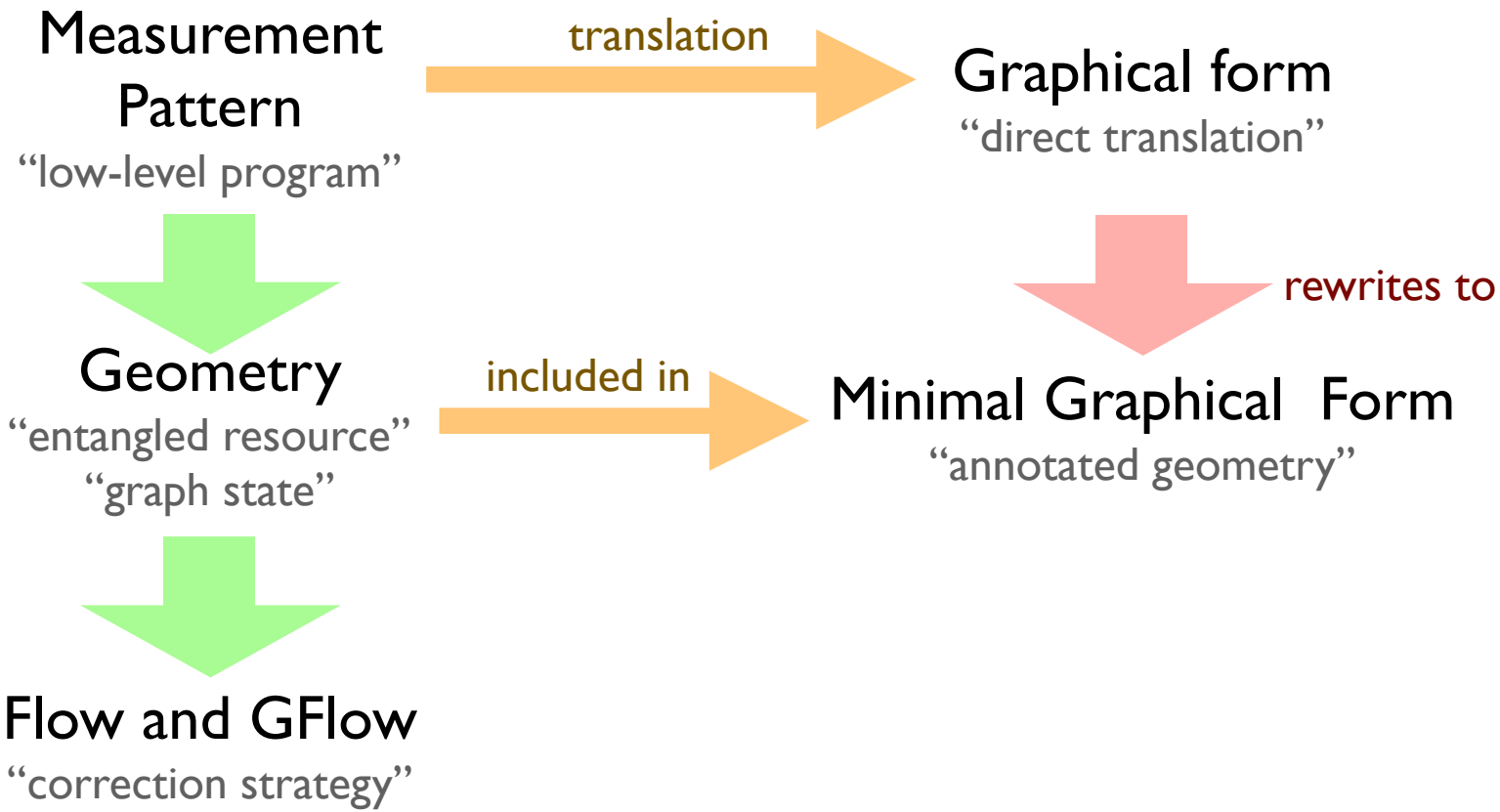
"entangled resource"

"graph state"

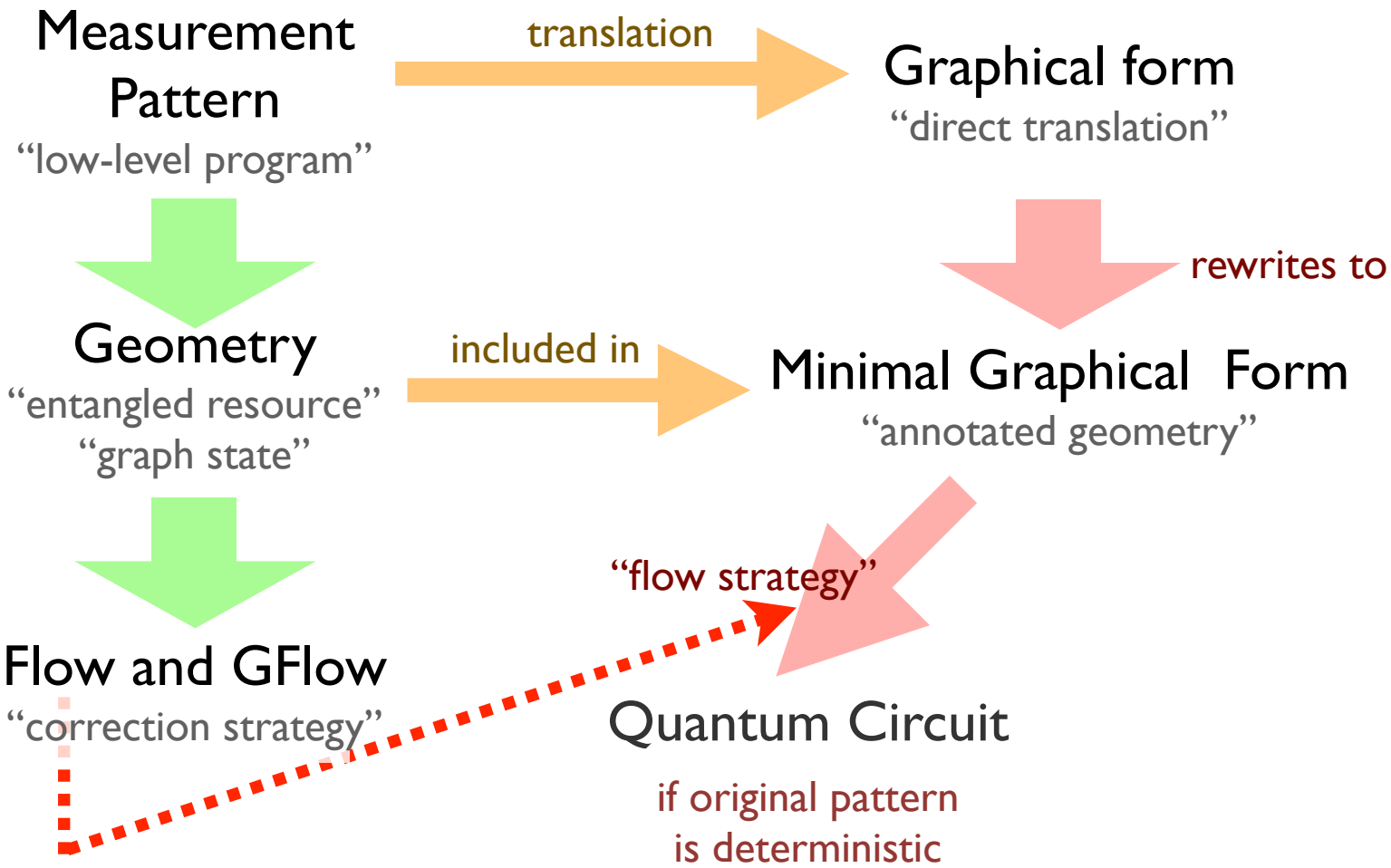
Flow and GFlow

"correction strategy"

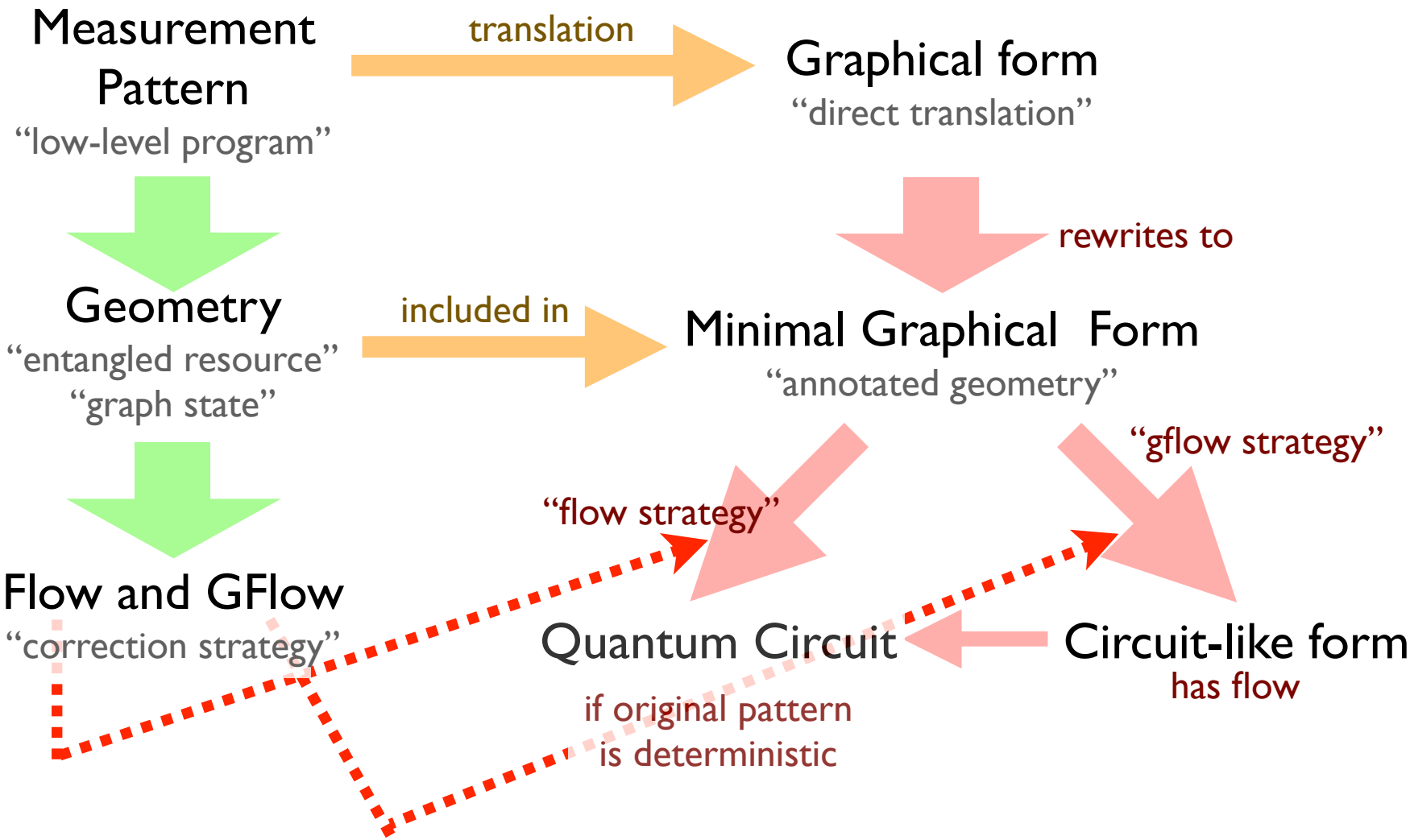
Summary of this talk:



Summary of this talk:



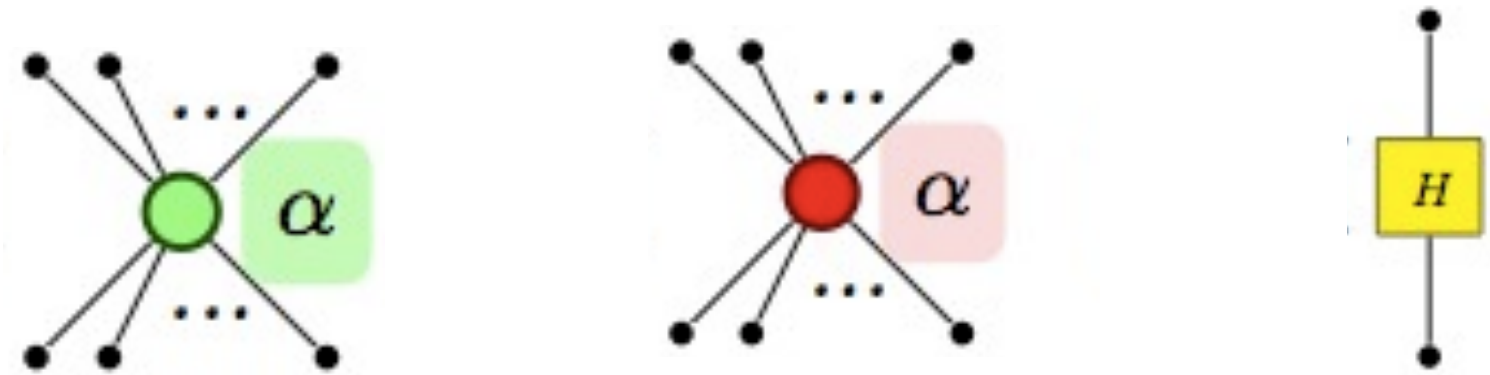
Summary of this talk:



Diagrams, Algebras and Circuits

Pessimistic Diagram Convention

Generators for diagrams

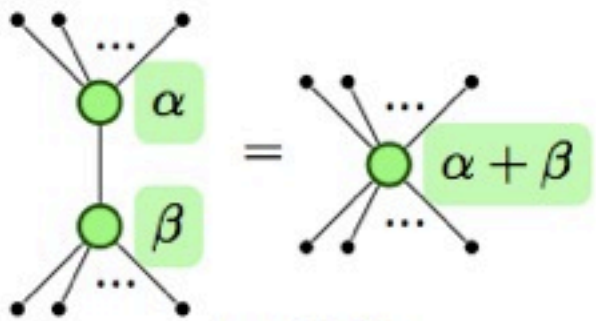


Defn: A diagram is an undirected open graph generated by the above vertices.

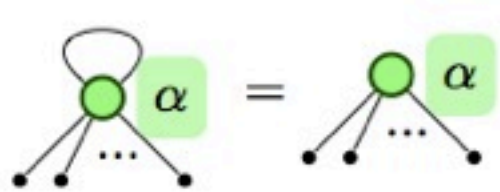
Defn: let $\mathbb{D}$ be the dagger compact category of diagrams s.t.

$$(\cdot)\dagger : \alpha \mapsto -\alpha$$

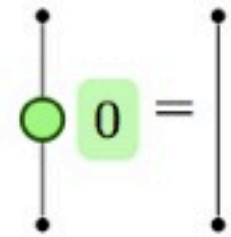
Equations



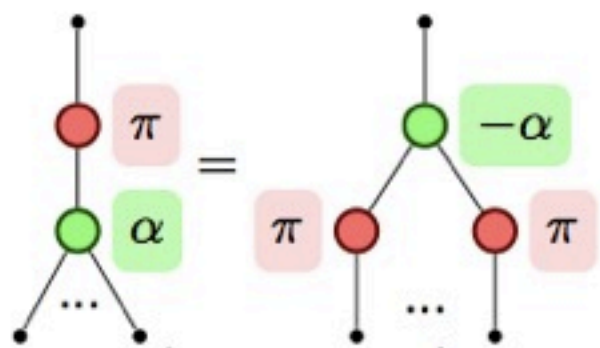
(spider)



(anti-loop)

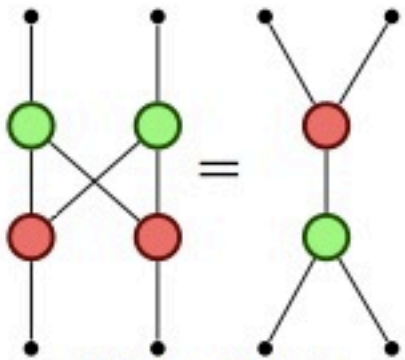


(identity)

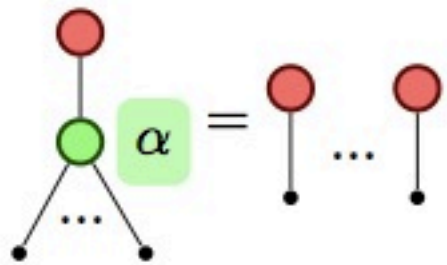


(π-commute)

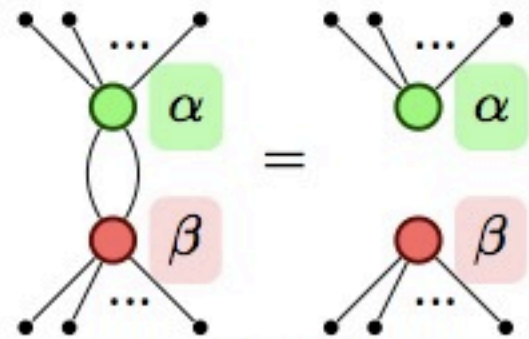
Equations



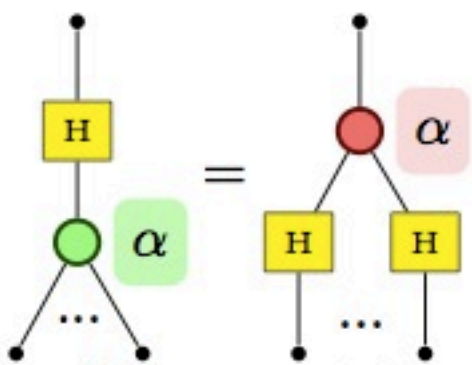
(bialgebra)



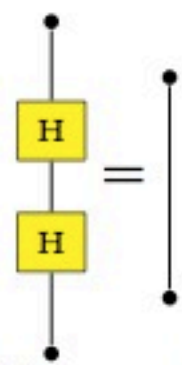
(copying)



(hopf)

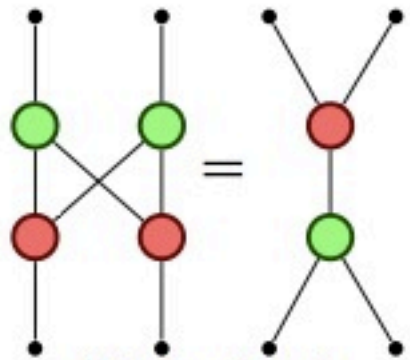


(H-commute)

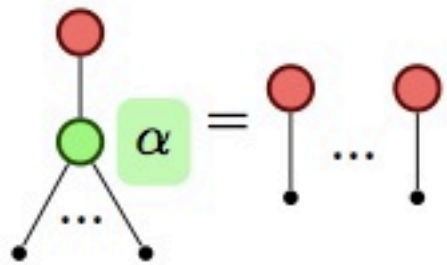


(H-cancel)

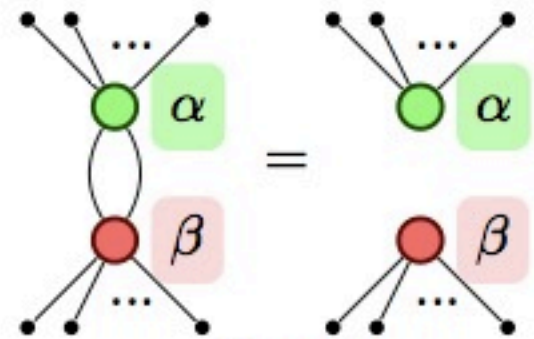
Equations



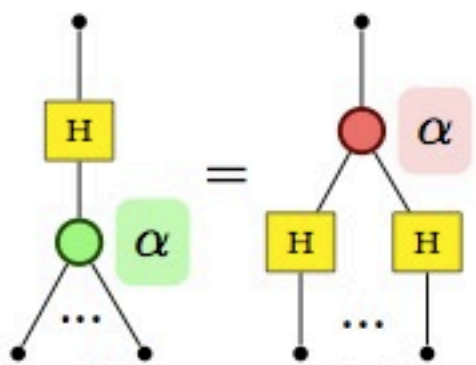
(bialgebra)



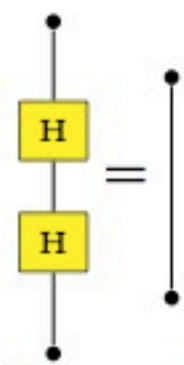
(copying)



(hopf)



(H-commute)



(H-cancel)

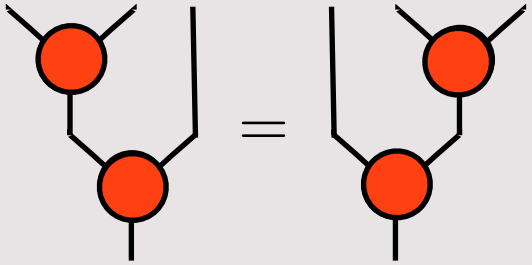
.... plus the same again with the colours exchanged

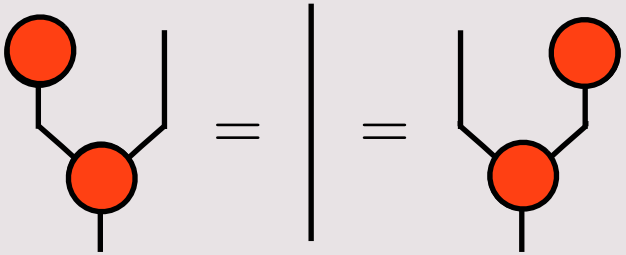
Algebraic Properties

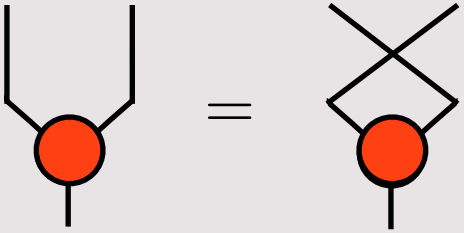
$m =$ 

$u =$ 

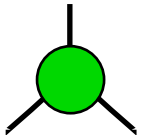
Monoid Laws

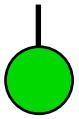




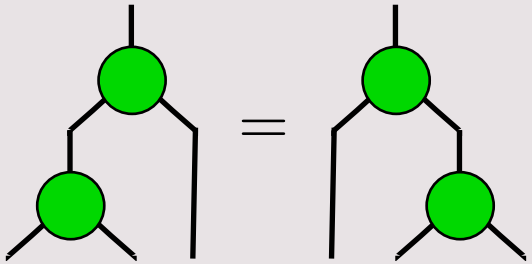


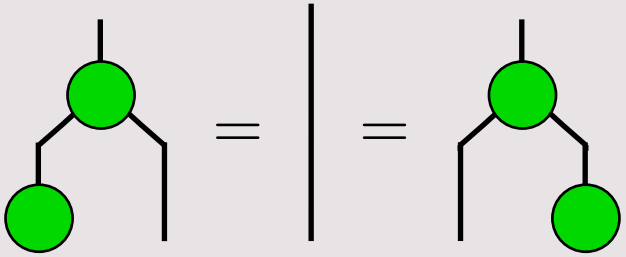
Algebraic Properties

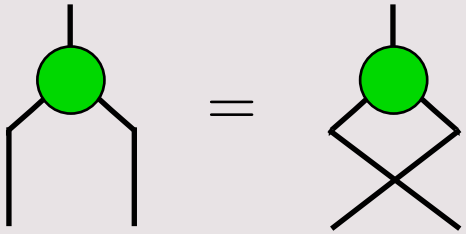
$\delta =$ 

$\epsilon =$ 

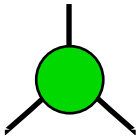
Comonoid Laws

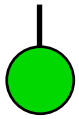






Algebraic Properties

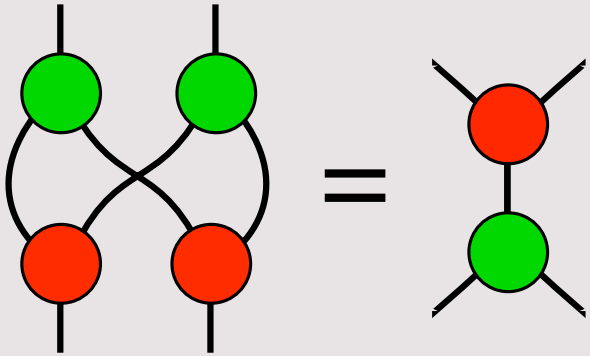
$\delta =$ 

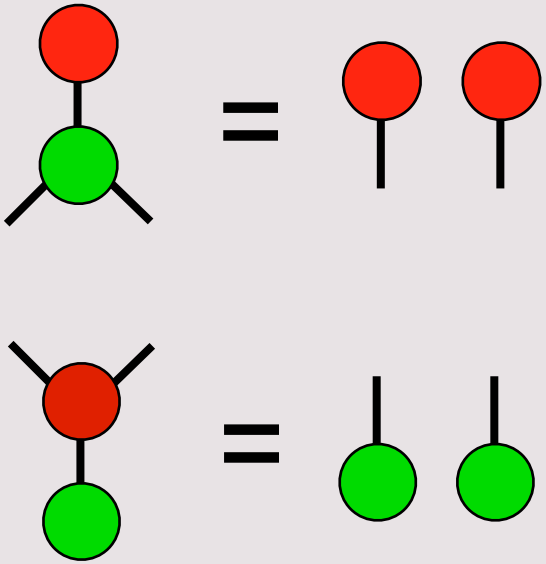
$\epsilon =$ 

$m =$ 

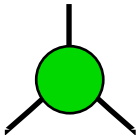
$u =$ 

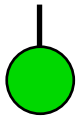
Bialgebra Laws





Algebraic Properties

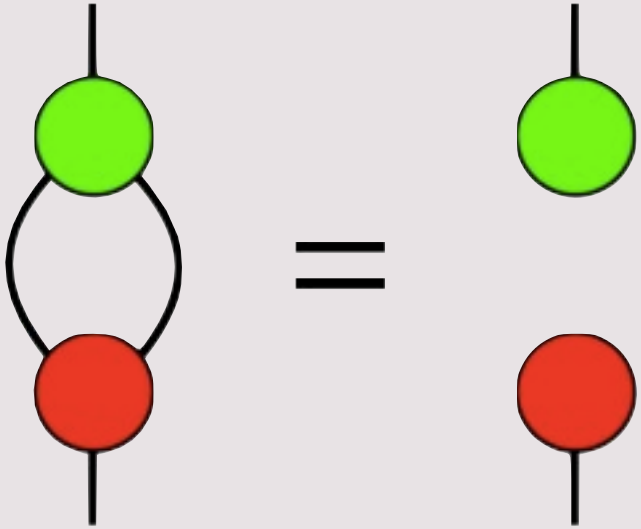
$\delta =$ 

$\epsilon =$ 

$m =$ 

$u =$ 

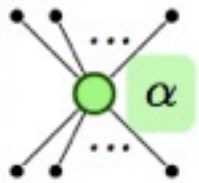
Hopf Algebra Laws



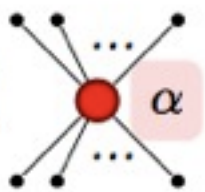
(with trivial antipode)

Semantics for diagrams

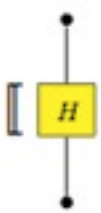
Defn: Let $[\![\cdot]\!] : \mathbb{D} \to \boldsymbol{FDHilb}$ be the traced monoidal functor such and that $[\![A]\!] = \mathbb{C}^{2^n}$ and define its action on generators by:



$$[\![\quad]\!] = \begin{cases} |0\rangle^{\otimes m} \mapsto |0\rangle^{\otimes n} \\ |1\rangle^{\otimes m} \mapsto e^{i\alpha}\,|1\rangle^{\otimes n} \end{cases}$$



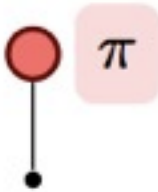
$$[\![\quad]\!] = \begin{cases} |+\rangle^{\otimes m} \mapsto |+\rangle^{\otimes n} \\ |-\rangle^{\otimes m} \mapsto e^{i\alpha}\,|-\rangle^{\otimes n} \end{cases}$$



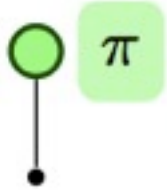
$$[\![\quad]\!] = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Representing Qubits

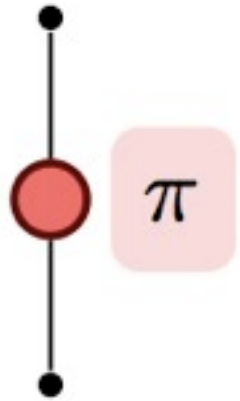
$[\![$  $]\!] = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle$

$[\![$  $]\!] = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle$

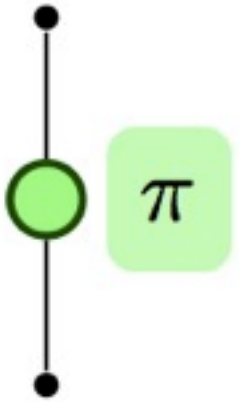
$[\![$  $]\!] = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = |+\rangle$

$[\![$  $]\!] = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = |-\rangle$

Representing Logic Gates

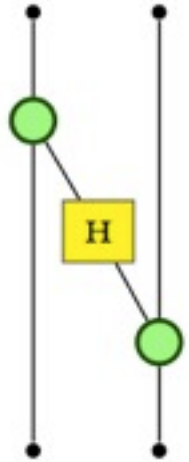
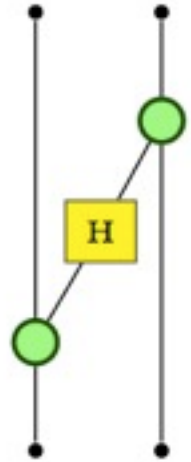


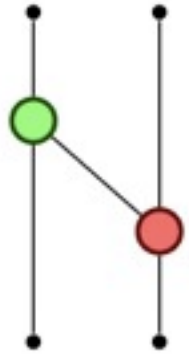
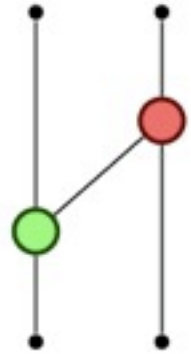
$$[\![\quad \pi \quad]\!] = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$



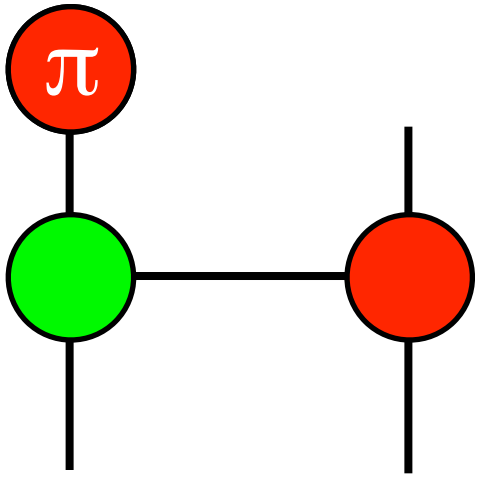
$$[\![\quad \pi \quad]\!] = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Representing Logic Gates

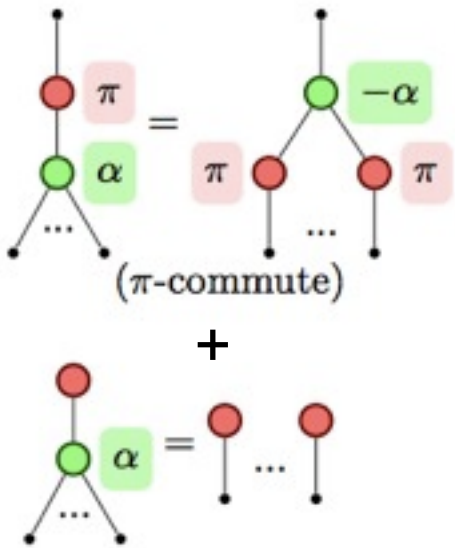
$\wedge Z = [\![$  $]\!] = [\![$  $]\!] = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$

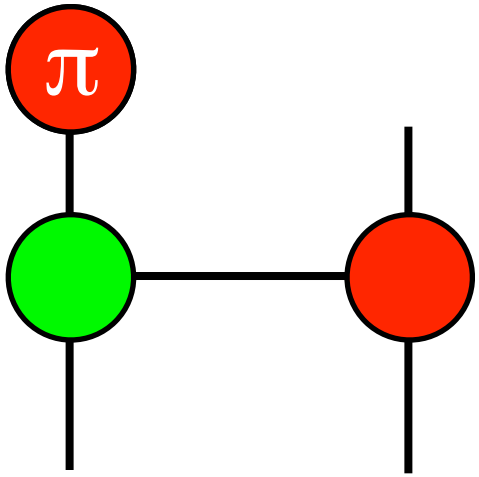
$\wedge X = [\![$  $]\!] = [\![$  $]\!] = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$

Reasoning via rewrites:

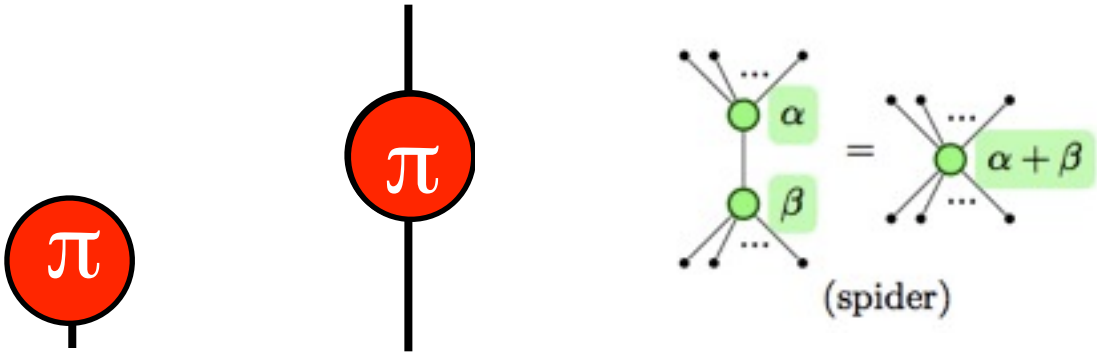


Reasoning via rewrites:



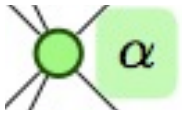


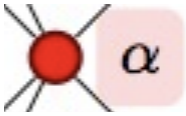
Reasoning via rewrites:



Which diagrams are circuits?

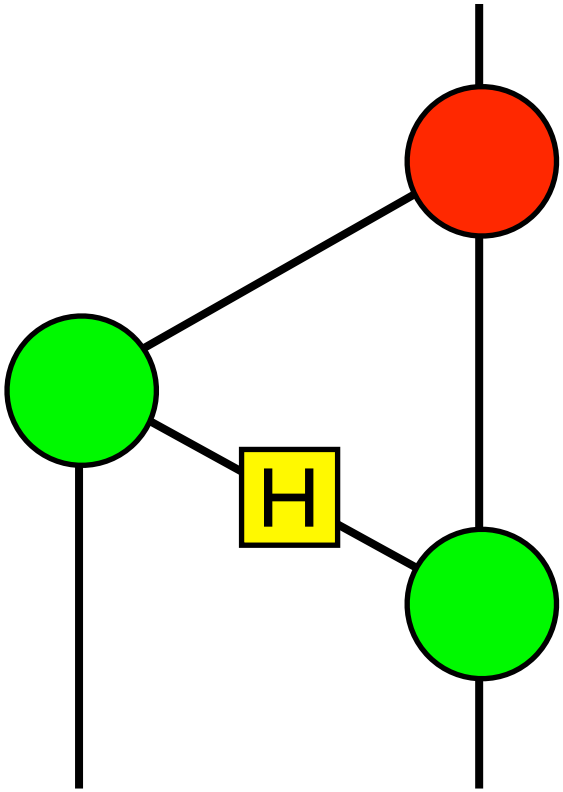
Defn: a diagram is called **circuit-like** if:

- All of its 

, 

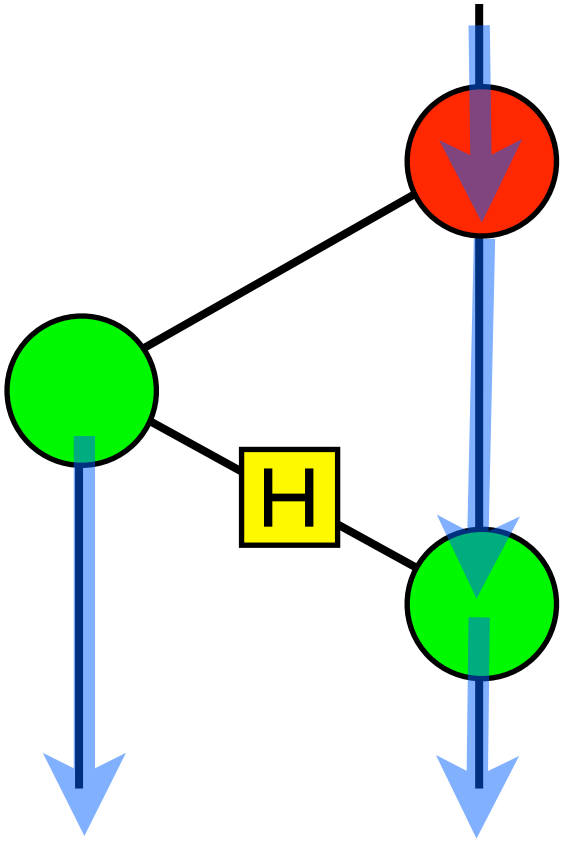
, and boundary vertices can be covered by set of disjoint paths P, each of which ends in an output.
- Every cycle in the diagram which overlaps with 2 paths in P traverses an edge in the opposite direction to P.
- It is 3-coloured.

Which diagrams are circuits?

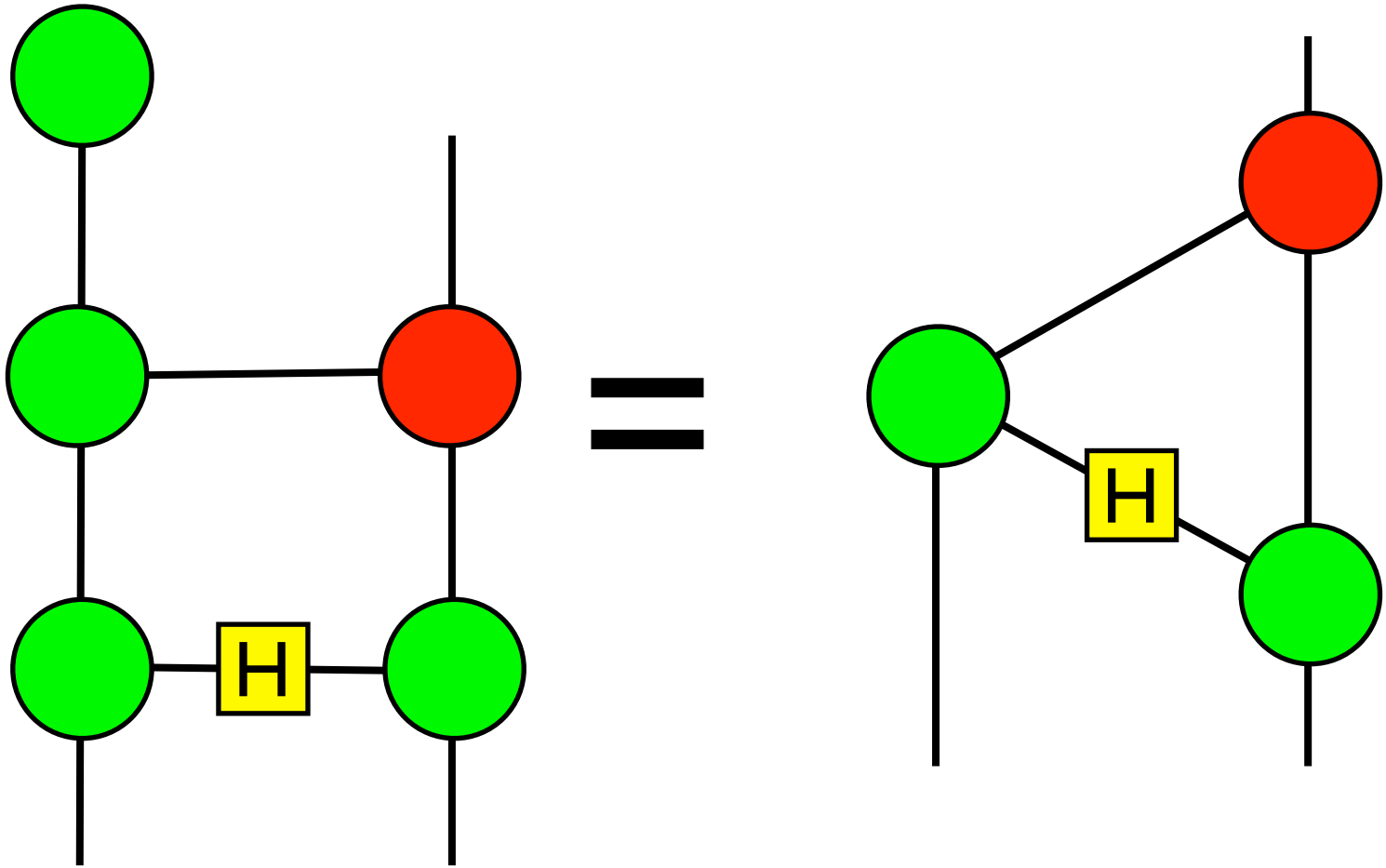


Which diagrams are circuits?

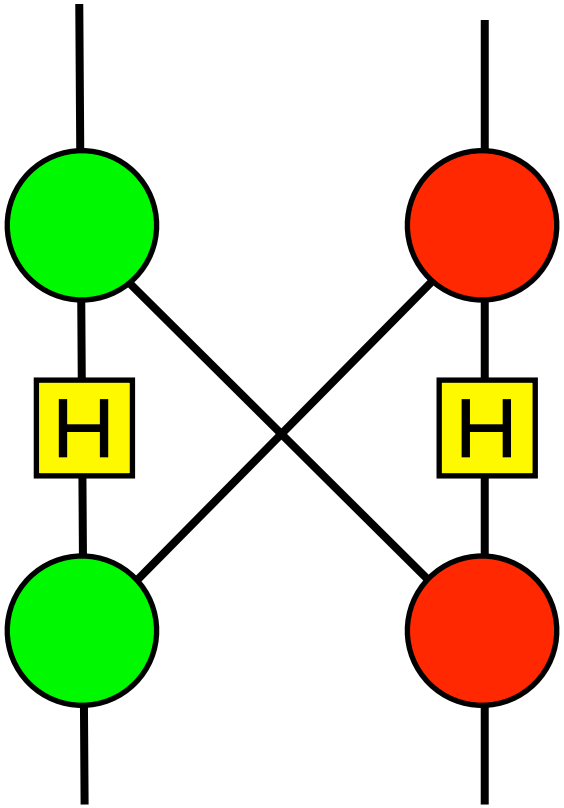
YES!



Which diagrams are circuits?

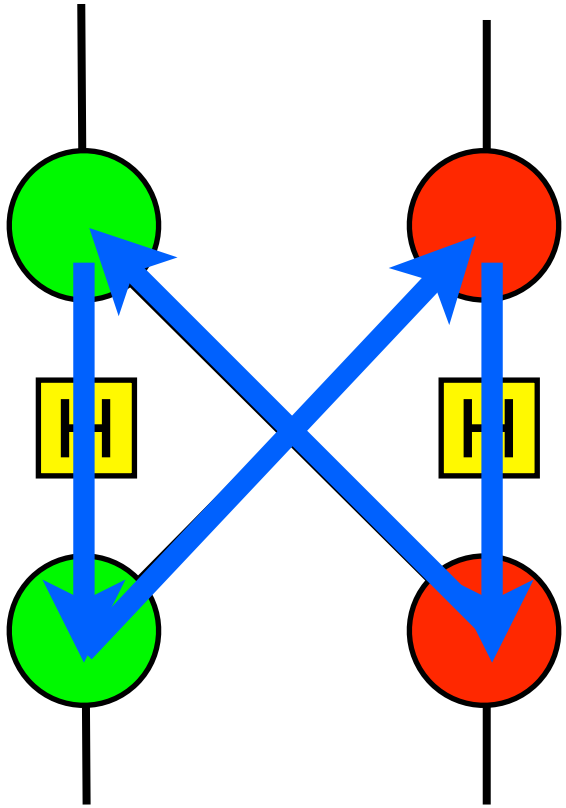


Which diagrams are circuits?



Which diagrams are circuits?

NO :(



Which diagrams are circuits?

Thm: if a diagram D is circuit-like then $[\![D]\!]$ is a unitary embedding.

Being circuit-like is a little stronger than being a unitary embedding: we require a particular gate set, and a particular encoding of the gate set as diagrams, and the minimality w.r.t. to spider. Up to these conditions, the theorem is *if and only if*.

Back to the 1-way model

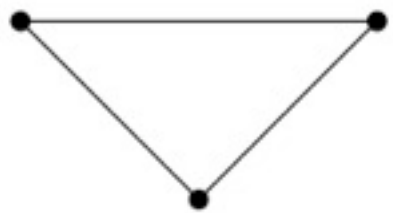
Graphs state computing computed graphically.

Graph States

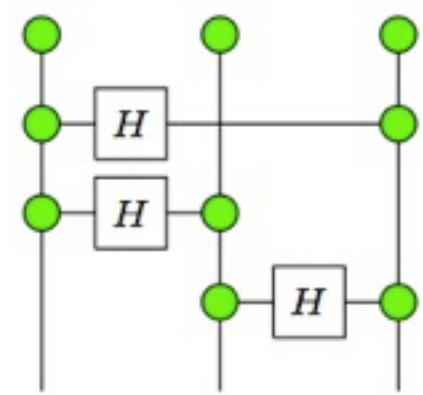
Defn: Given a simple undirected graph G we define a state

$$|G\rangle = \left(\prod_{(u,v)\in E(G)} \wedge Z_{u,v} \right) \left(\bigotimes_{u\in V(G)} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}_u \right)$$

where V(G) and E(G) are the sets of vertices and edges of G.



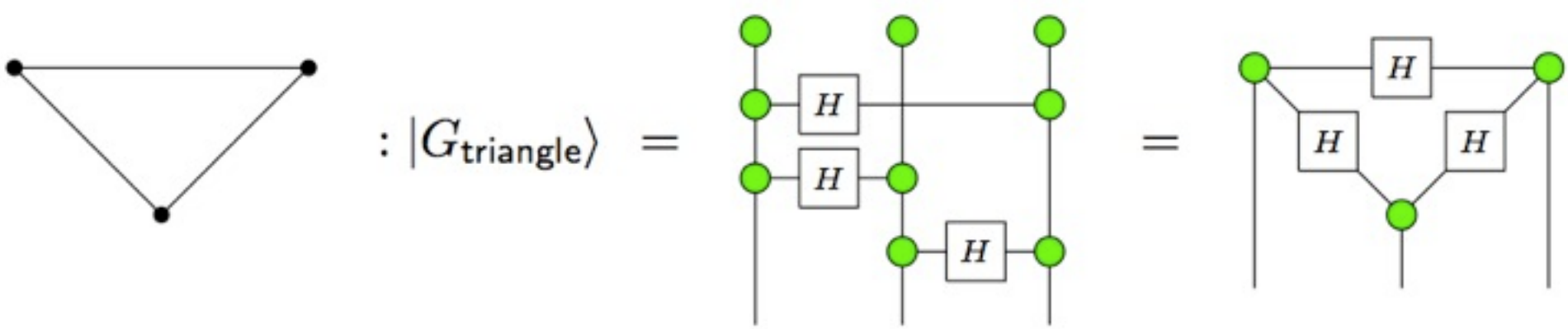
$: |G_{\text{triangle}}\rangle \;=$



Graph States

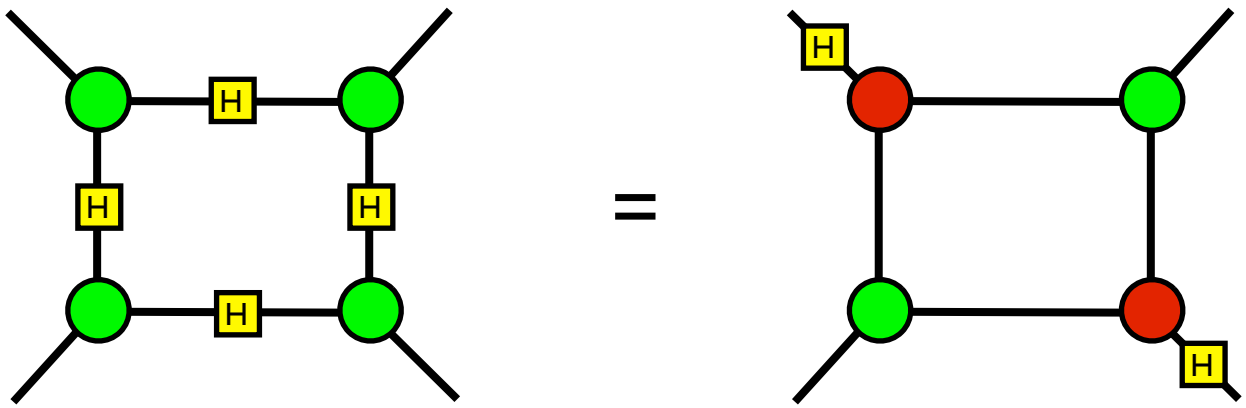
Hence a graph state defined by a graph G is represented by the same graph G with:

- green dots at each vertex corresponding to each qubit
- Hs on every edge, correspoding to the CZ interaction
- one output wire at each qubit



Graph States

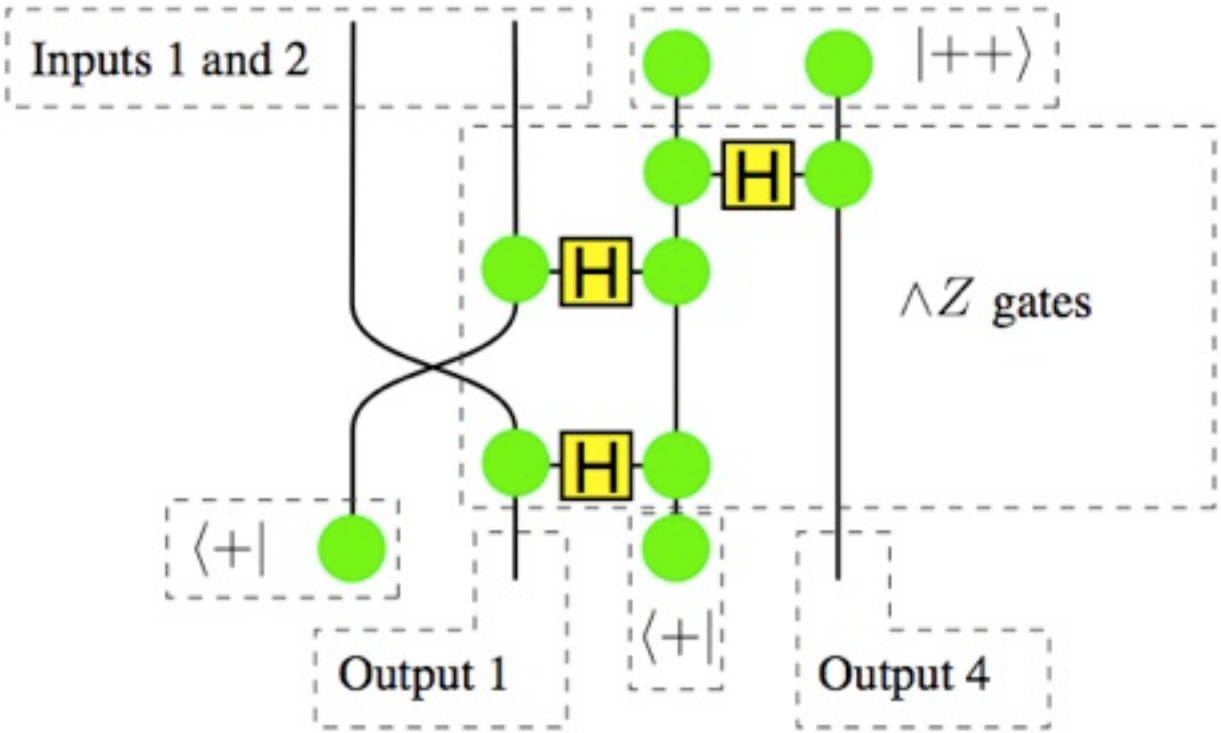
If a graph state G can be 2-coloured, we can eliminate the H:



This will be useful later!

Example: CNOT gate

$$M_2^0 M_4^0 E_{13} E_{23} E_{34} N_3 N_4$$



Example: CNOT gate

$$M_2^0 M_4^0 E_{13} E_{23} E_{34} N_3 N_4$$

Determinism in the one-way model

We can use the existence of a flow on a graph state to convert a one-way pattern to a quantum circuit:

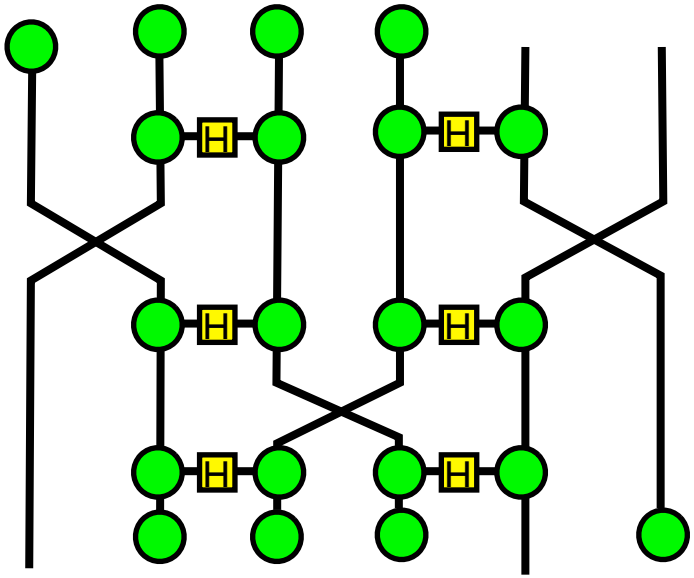
- Convert the pattern/geometry to diagram

$$M_5^0 M_4^0 M_3^0 M_2^0 E_{13} E_{23} E_{34} E_{15} E_{45} E_{56} N_{3456}$$

Determinism in the one-way model

We can use the existence of a flow on a graph state to convert a one-way pattern to a quantum circuit:

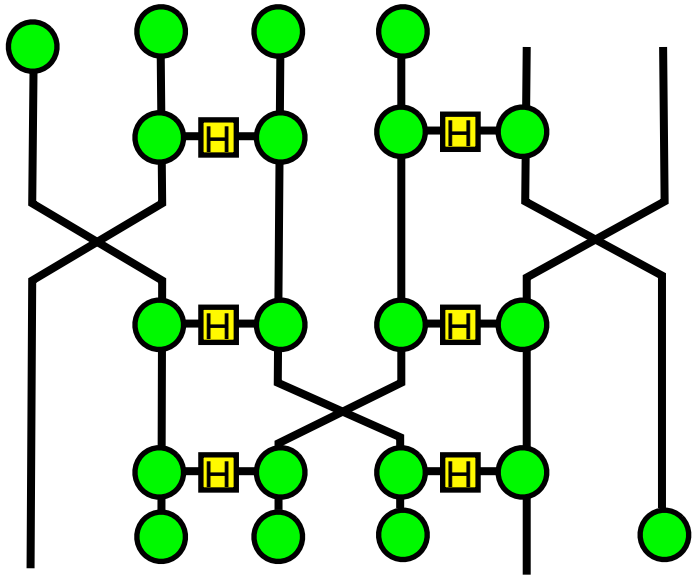
- Convert the pattern/geometry to diagram



Determinism in the one-way model

We can use the existence of a flow on a graph state to convert a one-way pattern to a quantum circuit:

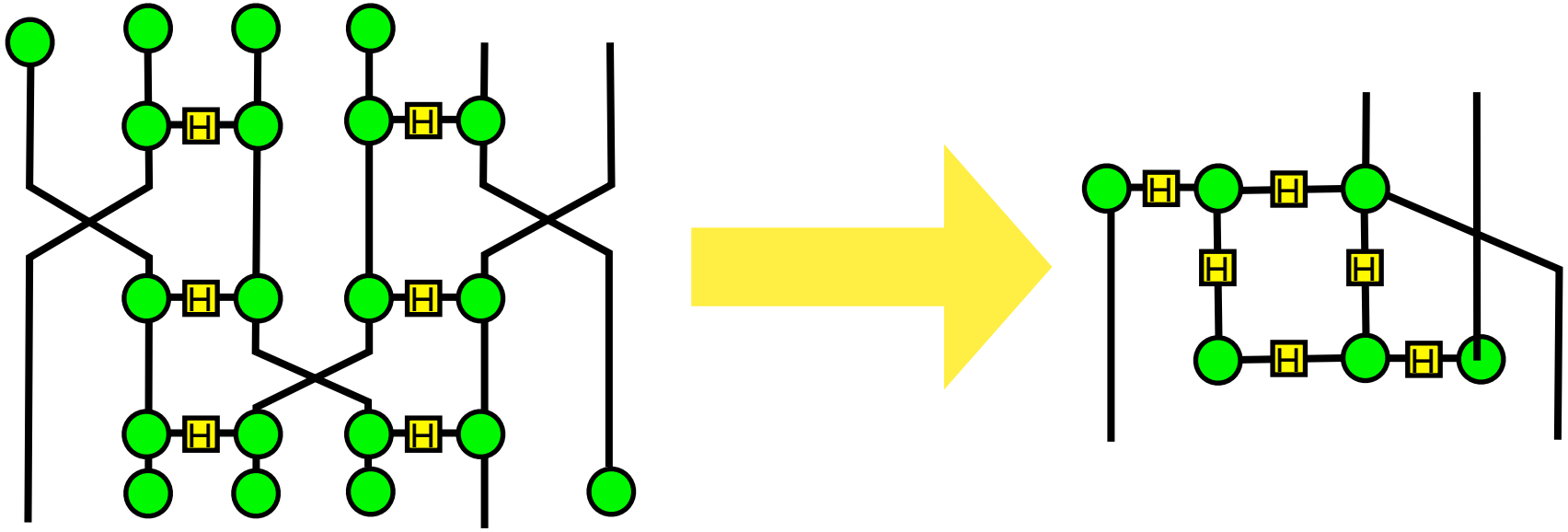
- apply the spider rule to obtain geometry



Determinism in the one-way model

We can use the existence of a flow on a graph state to convert a one-way pattern to a quantum circuit:

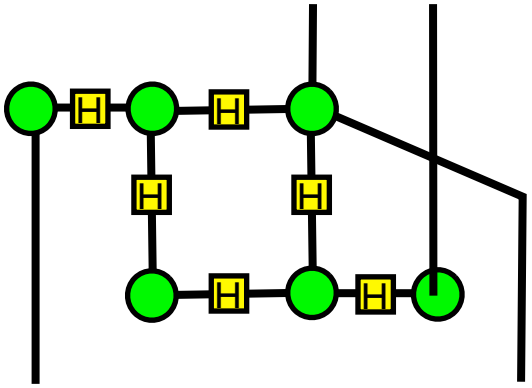
- apply the spider rule to obtain geometry



Determinism in the one-way model

We can use the existence of a flow on a graph state to convert a one-way pattern to a quantum circuit:

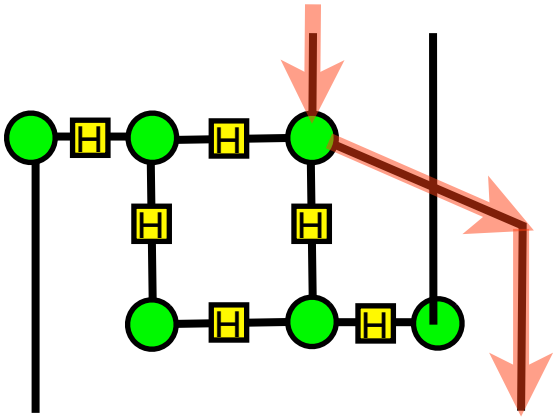
- Look at flow:



Determinism in the one-way model

We can use the existence of a flow on a graph state to convert a one-way pattern to a quantum circuit:

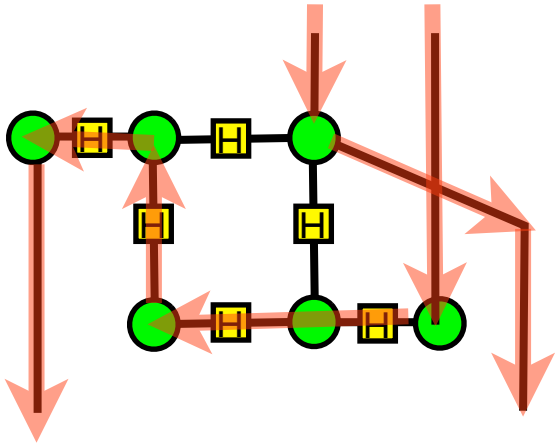
- Look at flow:



Determinism in the one-way model

We can use the existence of a flow on a graph state to convert a one-way pattern to a quantum circuit:

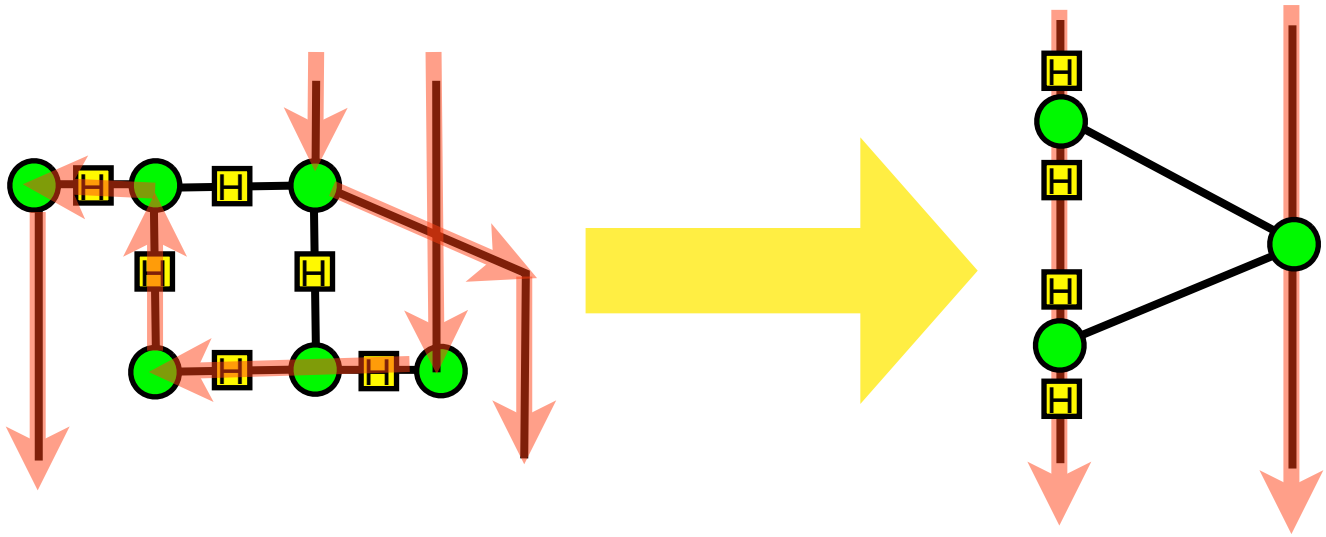
- Look at flow:



Determinism in the one-way model

We can use the existence of a flow on a graph state to convert a one-way pattern to a quantum circuit:

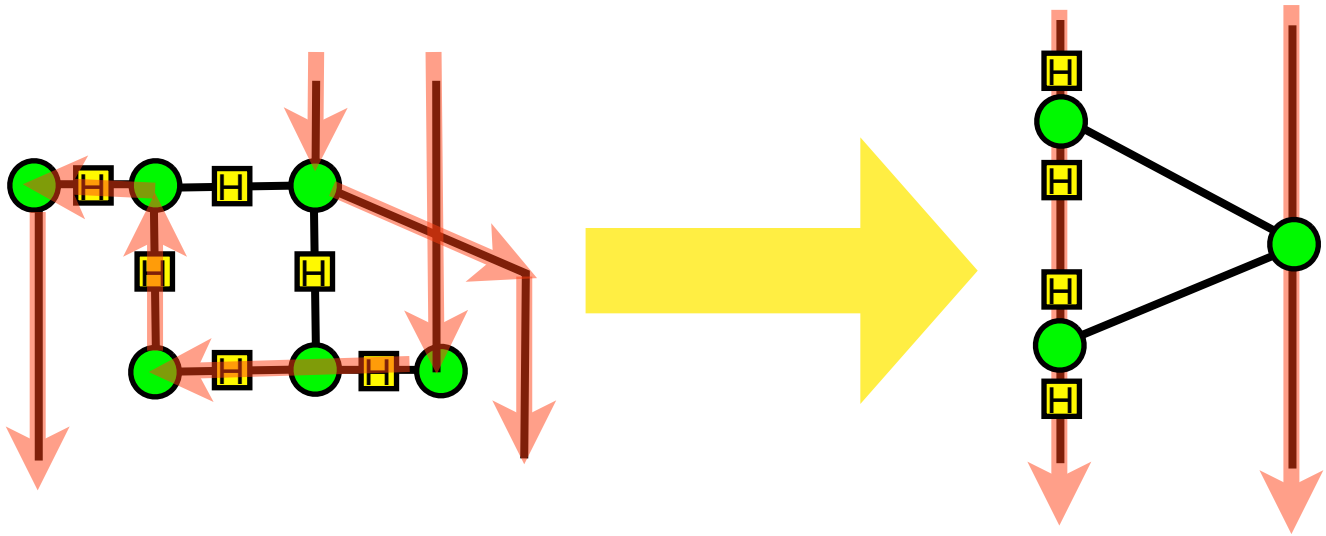
- Derive circuit by topology change:



Determinism in the one-way model

We can use the existence of a flow on a graph state to convert a one-way pattern to a quantum circuit:

- Derive circuit by topology change:

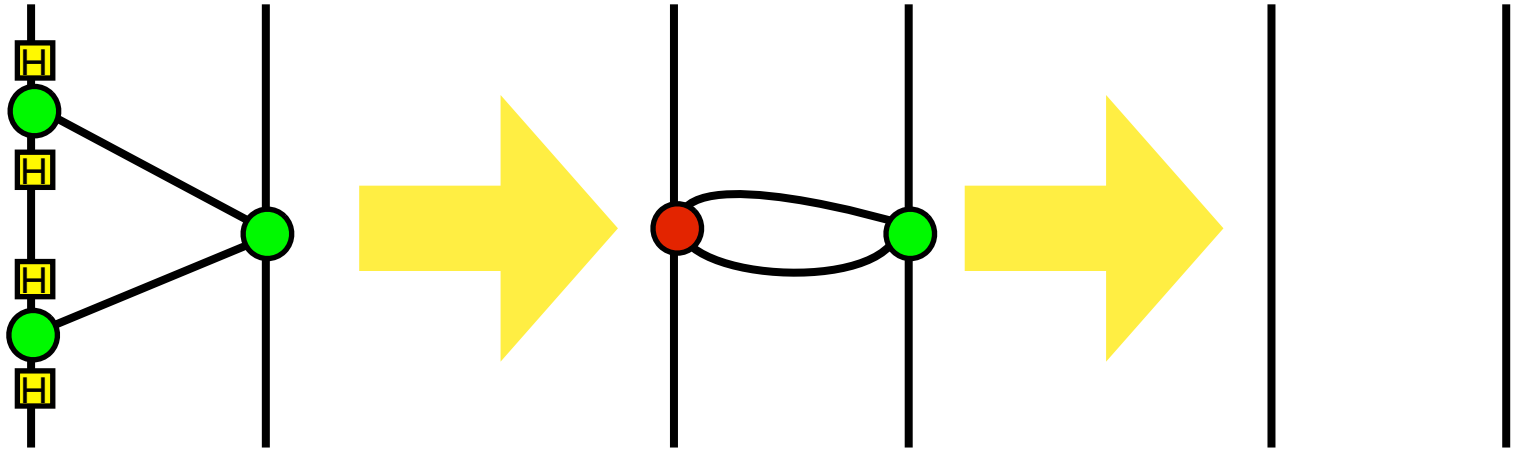


Prop: if the underlying graph state has a flow, then its diagram is circuit-like

Determinism in the one-way model

We can use the existence of a flow on a graph state to convert a one-way pattern to a quantum circuit:

- Simplify circuit:



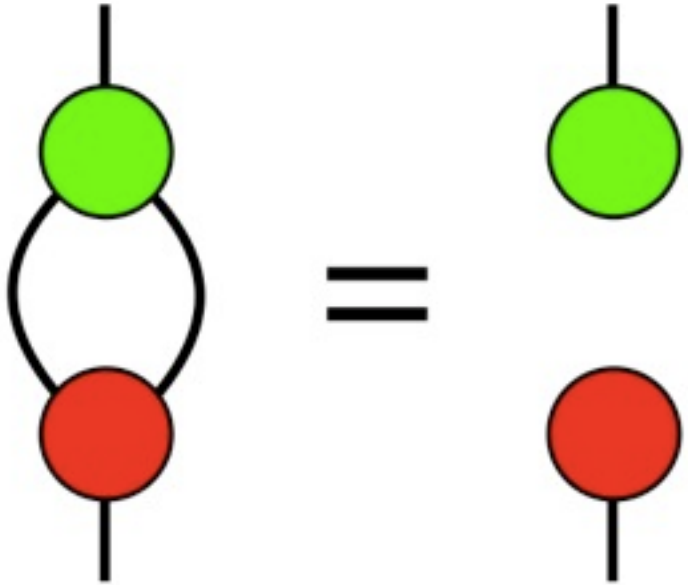
GFlow Strategy

What about **GFlow**?

- If a geometry has a GFlow, there exists a strategy that will transform the diagram into a diagram with flow, *and no ancilla qubits or acausal loops*.
- This strategy is based on the Hopf algebra normal form equivalence for graph states.

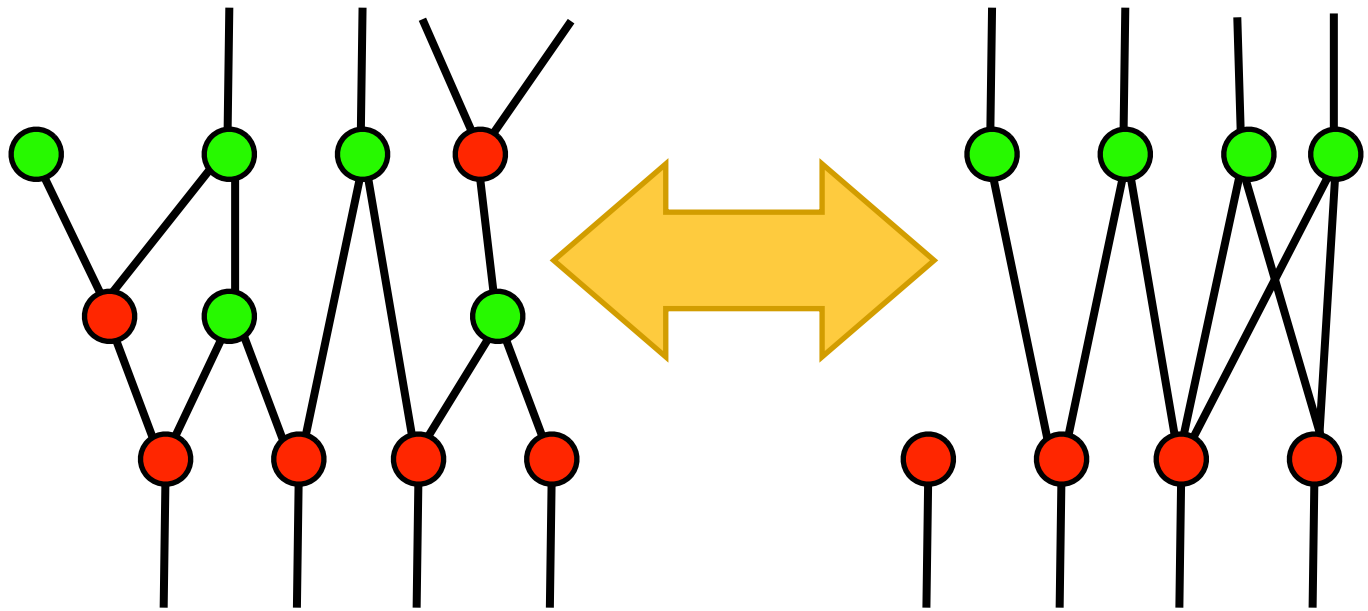
Hopf Algebra Equivalence

Recall that the red and green operators form a Hopf Algebra with a trivial antipode:



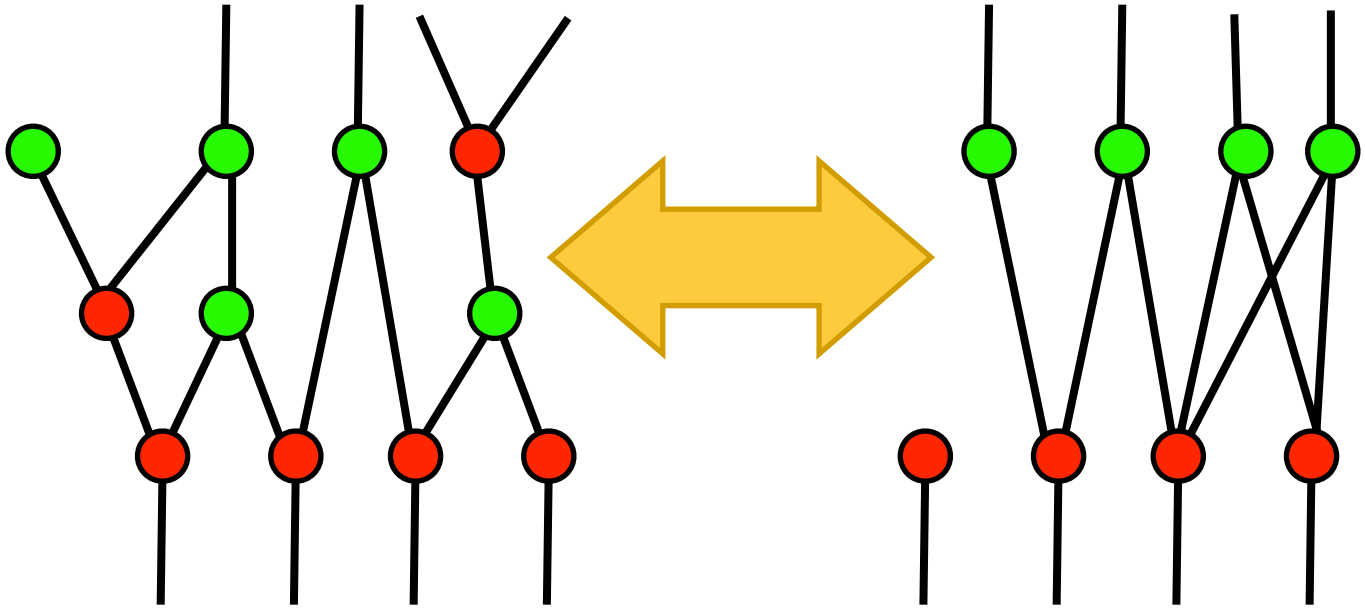
Hopf Algebra Equivalence

Thm: any Hopf algebra expression can be put into normal form:



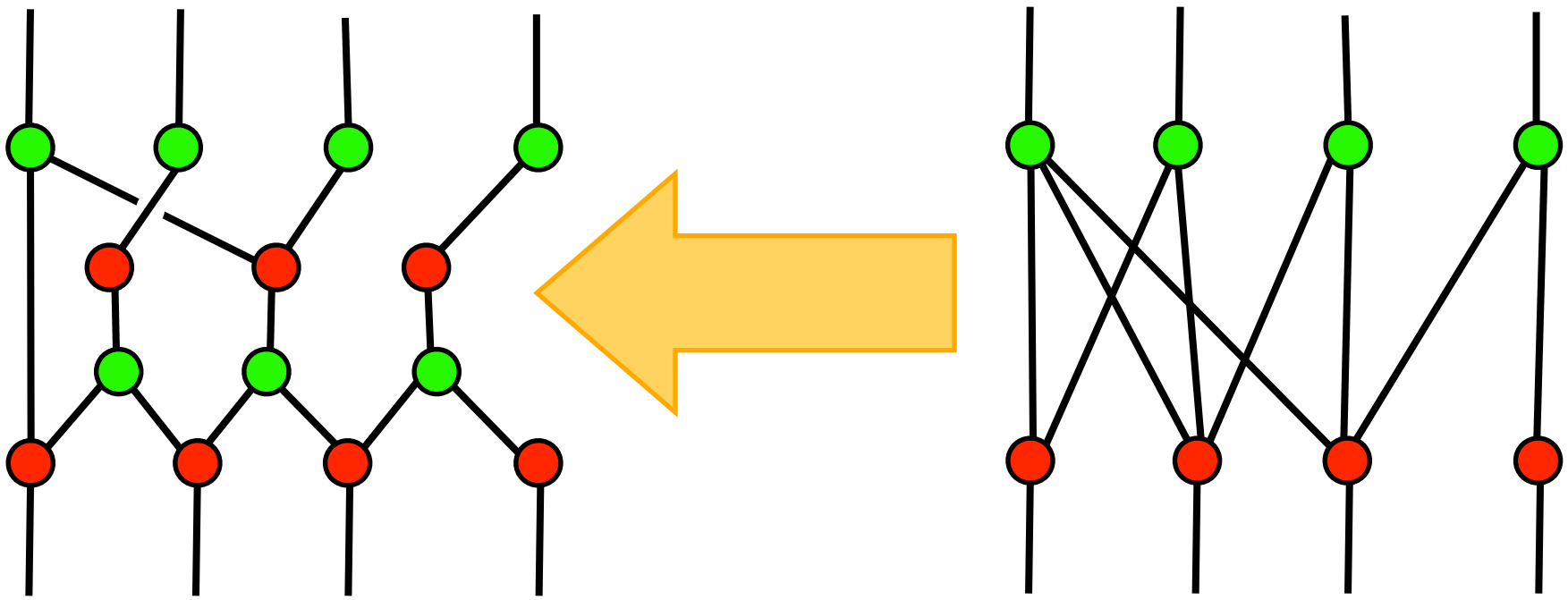
Hopf Algebra Equivalence

Cor: the corresponding "graph states" are equivalent.



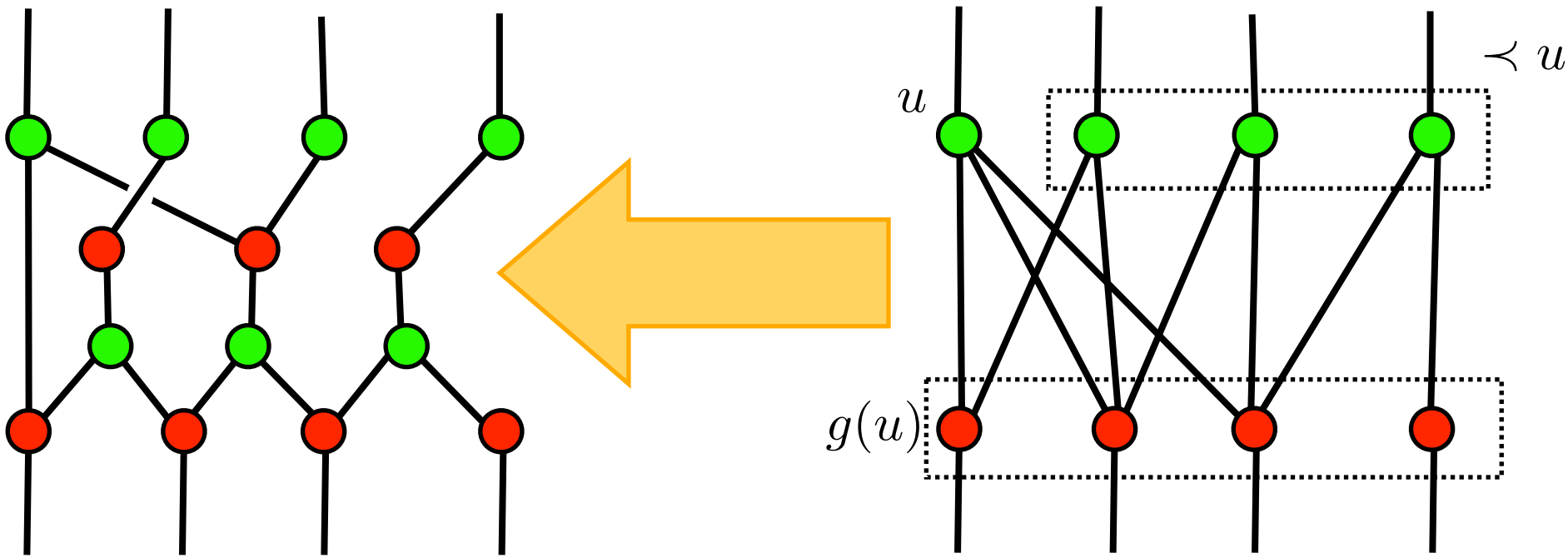
Hopf Algebra Equivalence

Strategy: use this to refine the GFlow:



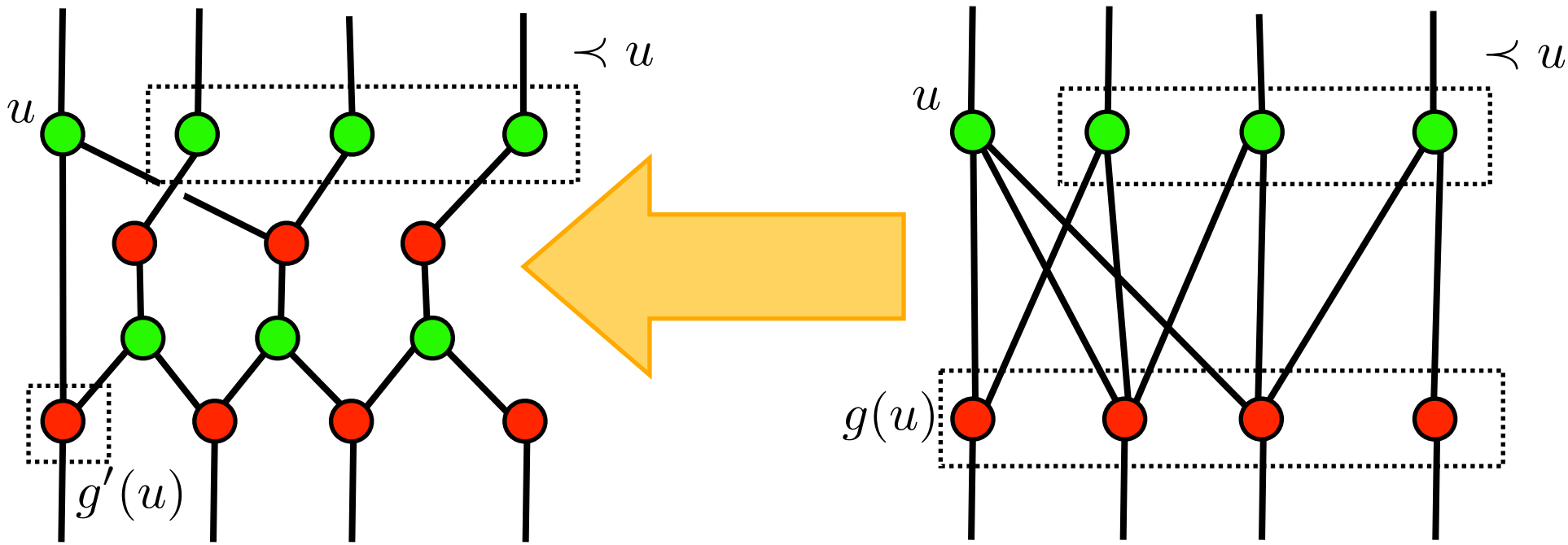
Hopf Algebra Equivalence

Strategy: use this to refine the GFlow:



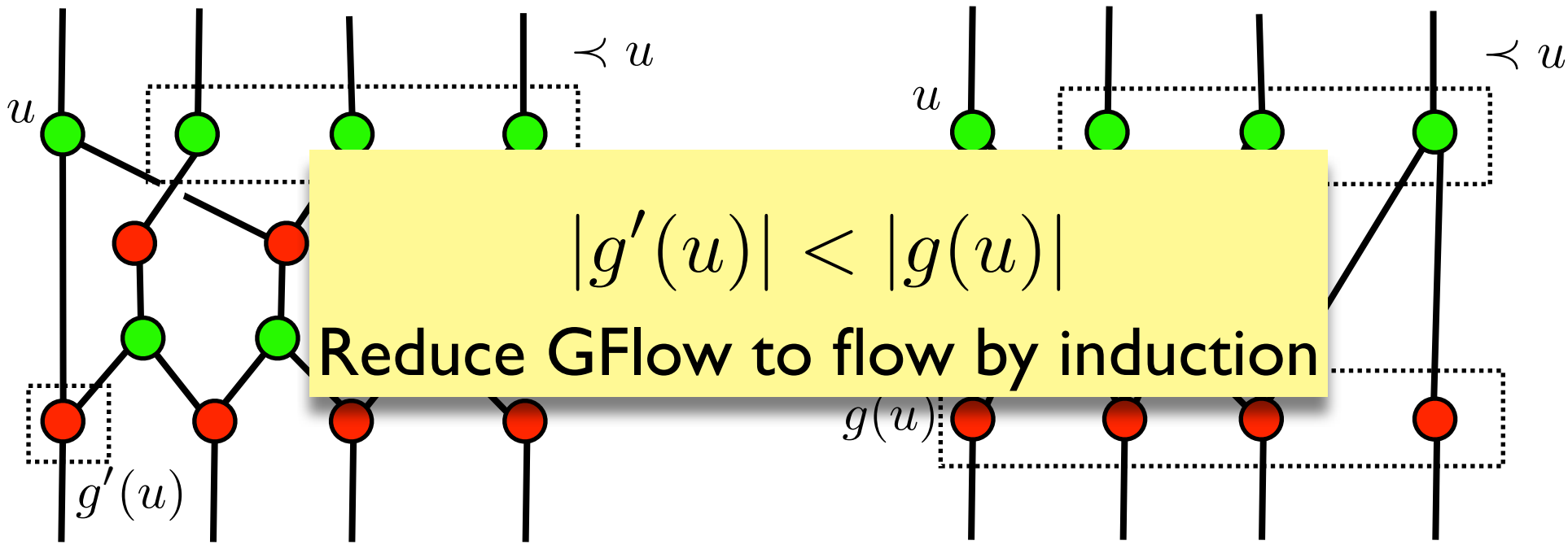
Hopf Algebra Equivalence

Strategy: use this to refine the GFlow:



Hopf Algebra Equivalence

Strategy: use this to refine the GFlow:



$$|g'(u)| < |g(u)|$$

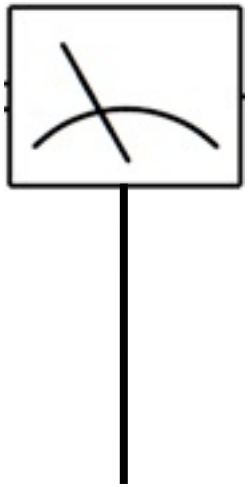
Reduce GFlow to flow by induction

“But quantum measurements are non-deterministic!”

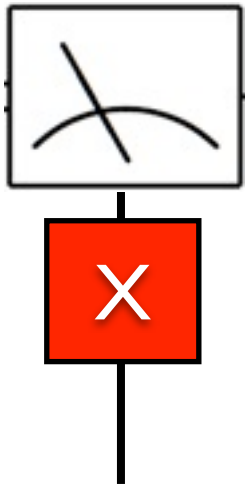
I know that, you dolt!

Remember this?

Yay!



$|0\rangle$

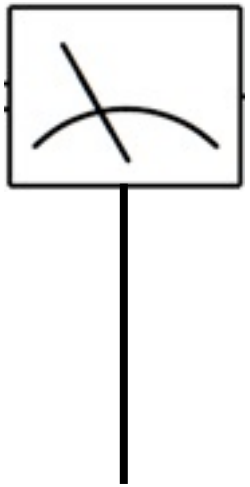


$|1\rangle$

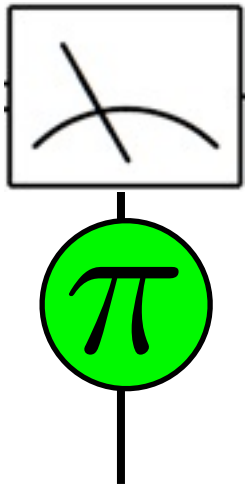
Yay!

Should be this:

Yay!



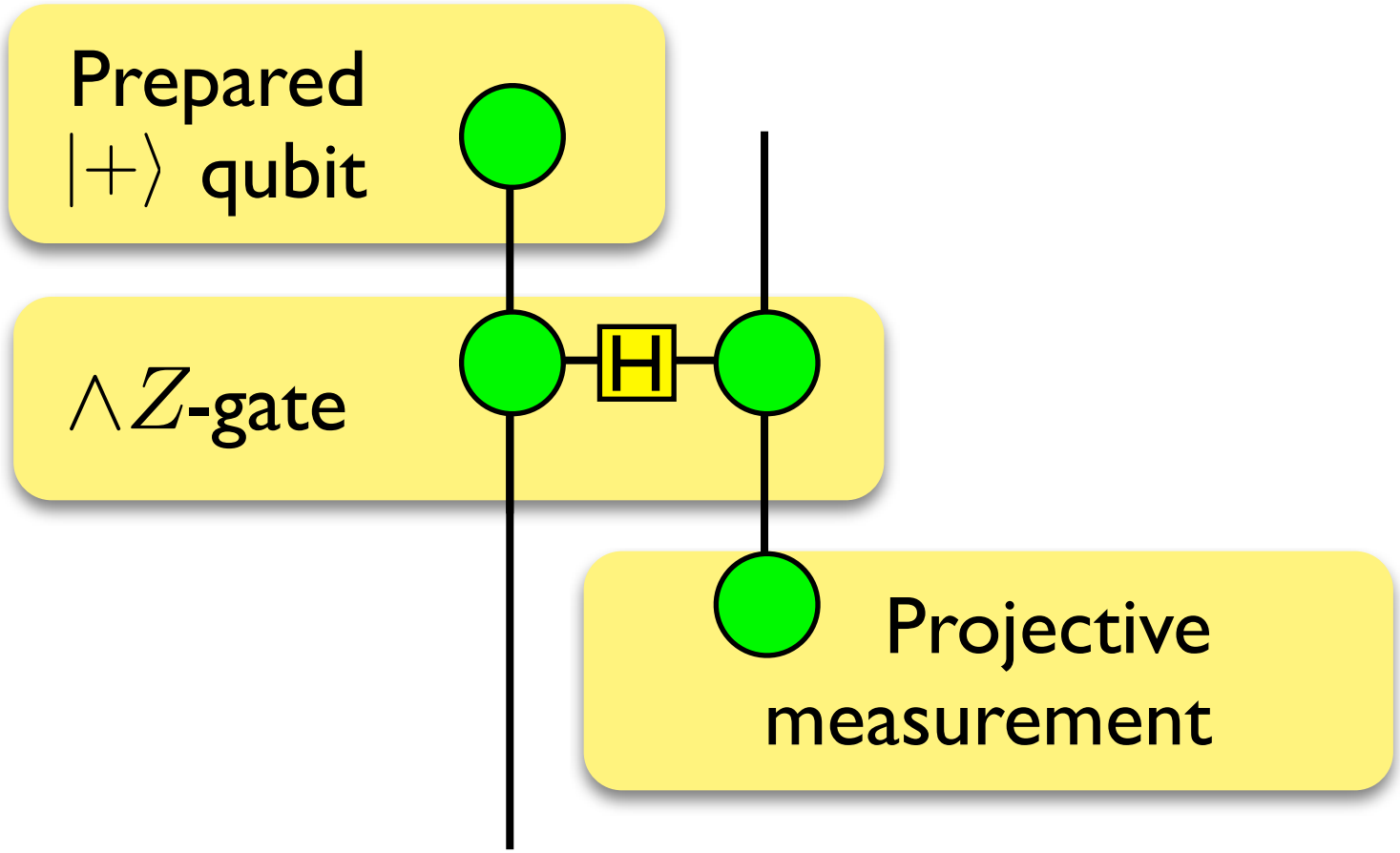
$|+\rangle$



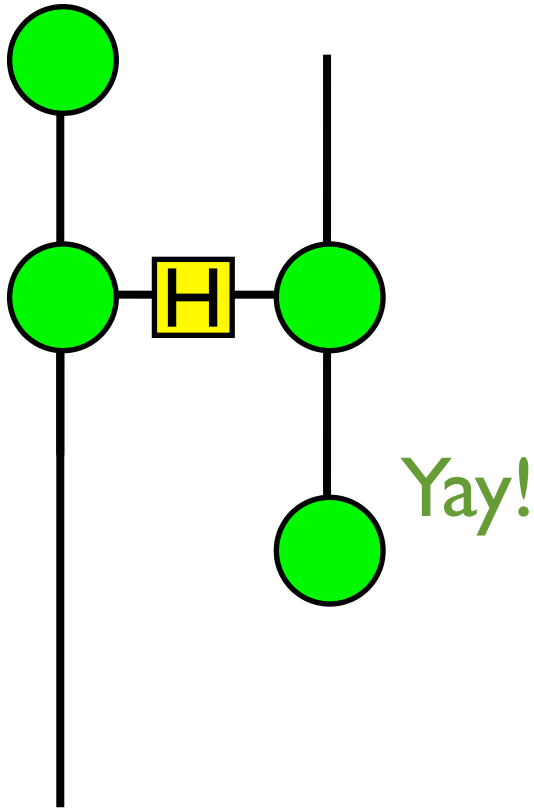
$|-\rangle$

Yay!

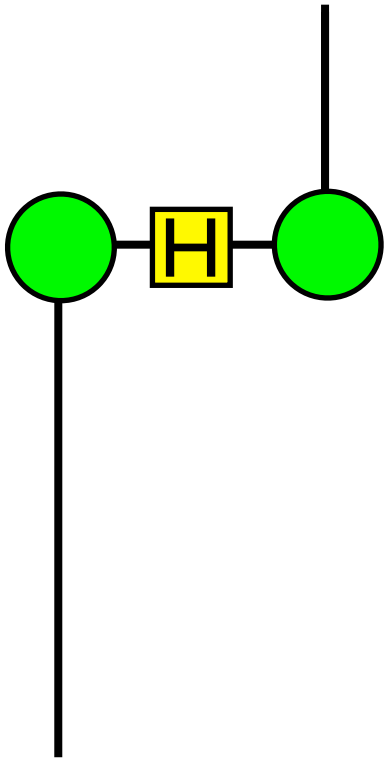
Example: Hadamard



Example: Hadamard

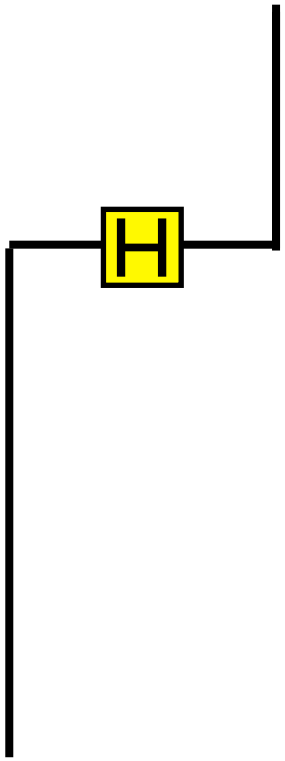


Example: Hadamard



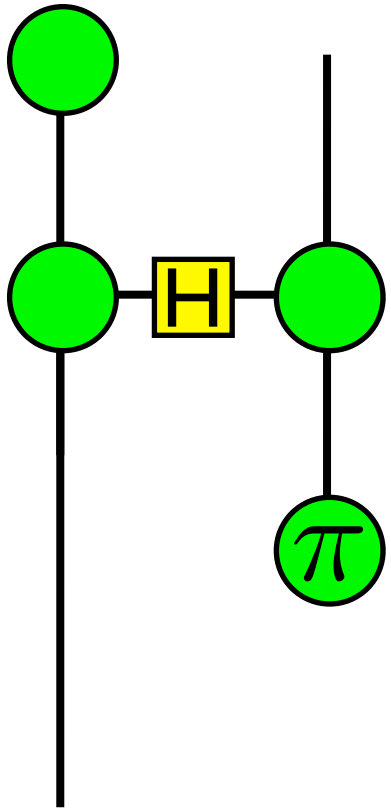
Yay!

Example: Hadamard

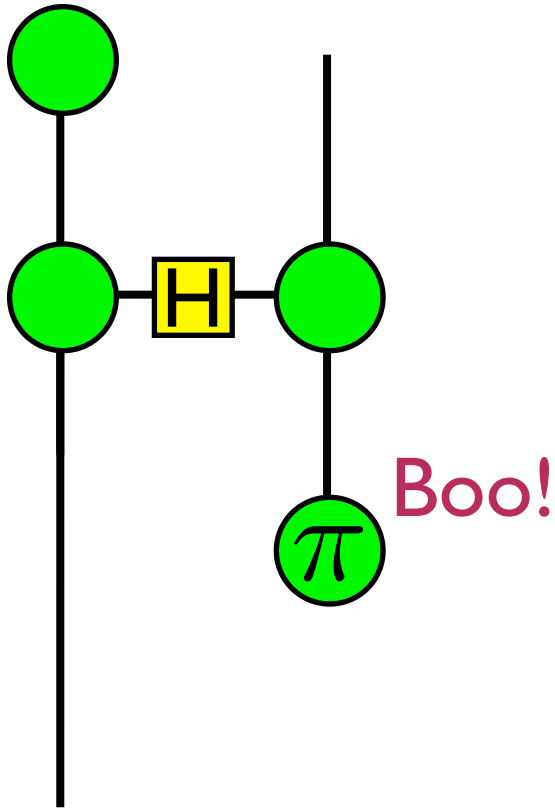


Yay!

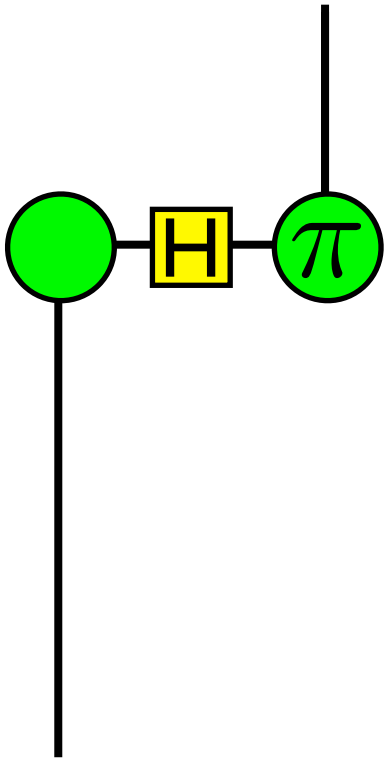
Example: Hadamard



Example: Hadamard

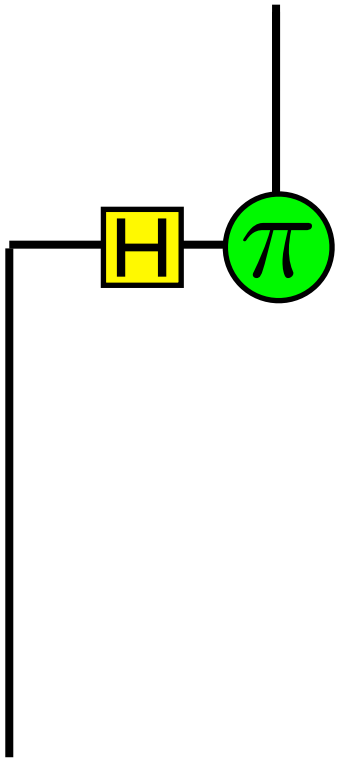


Example: Hadamard



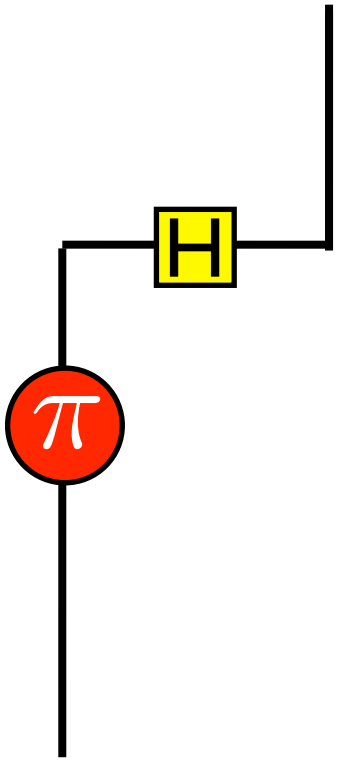
Boo!

Example: Hadamard



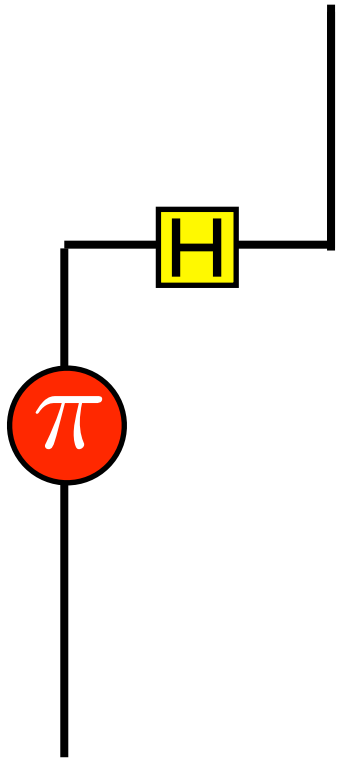
Boo!

Example: Hadamard



Boo!

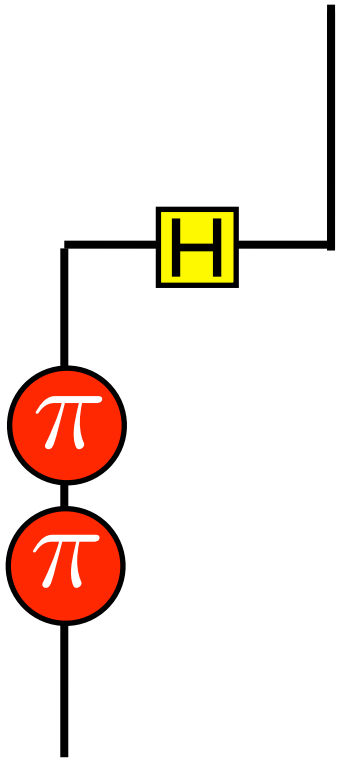
Example: Hadamard



Boo!

Add correction here:

Example: Hadamard

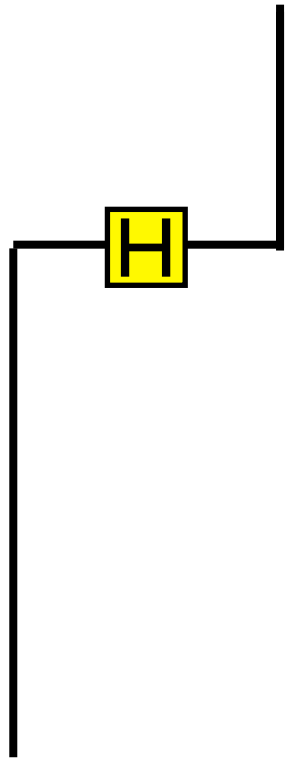


Add correction here:

Boo!

Example: Hadamard

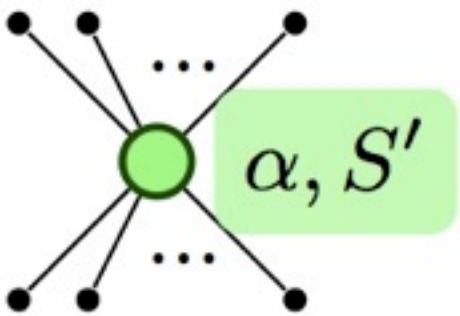
Add correction here:

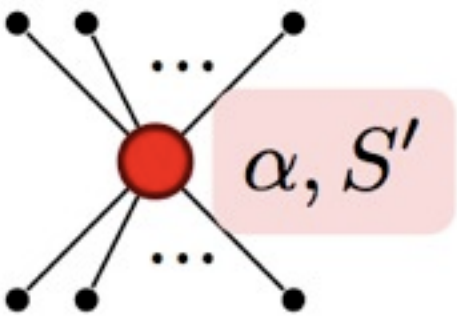


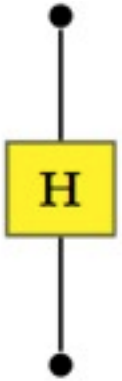
Boo!

Conditional Operations

Defn: Let S be a set of variables ("signals"); define $\mathbb{D}(S)$ to be the category of diagrams generated by:







where $S' \subseteq S$

Semantics

Defn: a *valuation* of S is a function $v : S \to \{0,1\}$. For each valuation we can define a map $\hat{v} : \mathbb{D}(S) \to \mathbb{D}$ by replacing the angle alpha with 0 if

$$\prod_{s \in S'} v(s) = 0$$

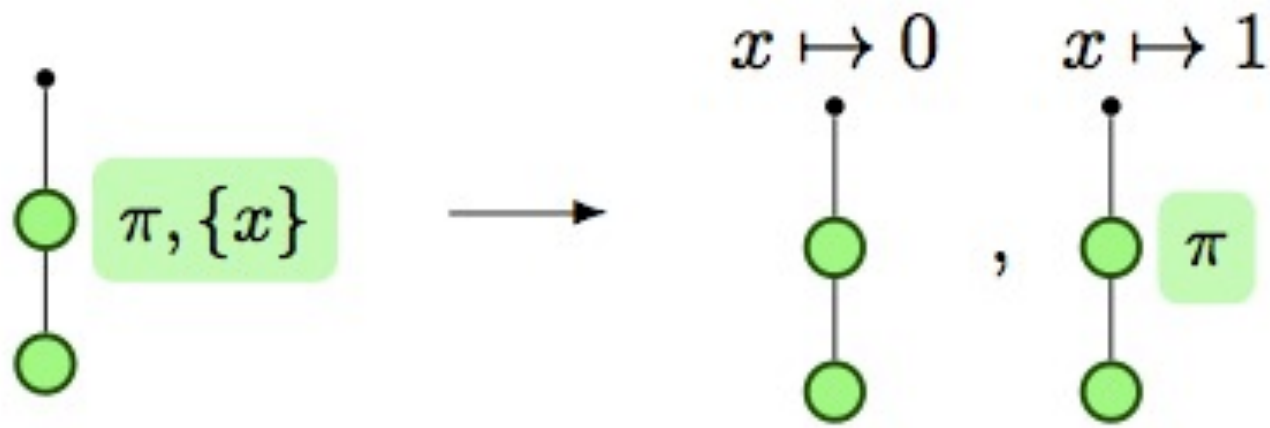
and leaving if unchanged otherwise.

Defn: The **denotation** of a diagram D in $\mathbb{D}(S)$ is a superoperator:

$$\rho \mapsto \sum_{v \in 2^S} [\![\hat{v}(D)]\!] \rho [\![\hat{v}(D)]\!]^\dagger$$

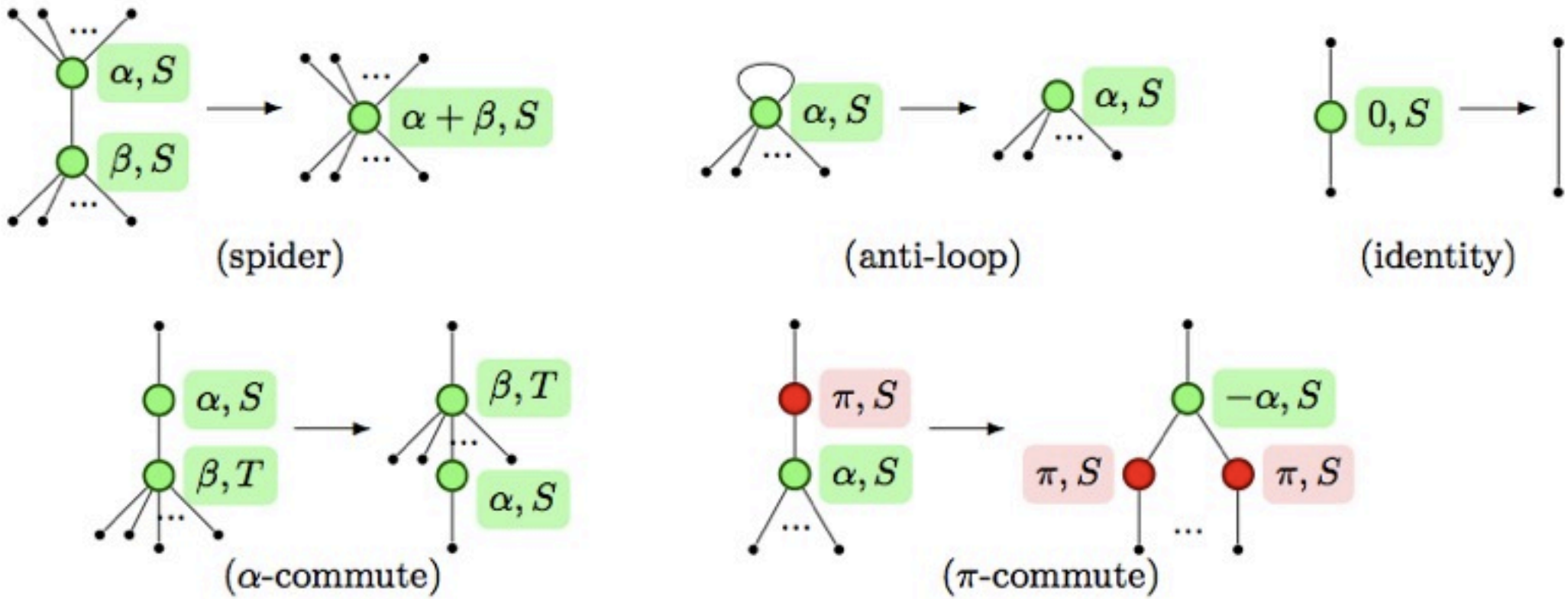
Example

A measurement in the $|+\rangle, |-\rangle$ basis:

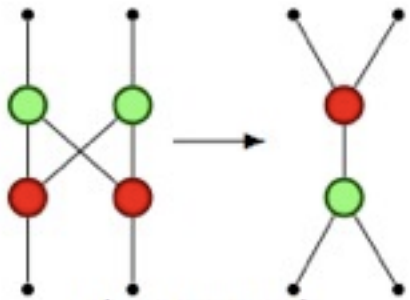


$$\rho \mapsto \langle +|\rho|+\rangle + \langle +|Z\rho Z|+\rangle$$
$$= \langle +|\rho|+\rangle + \langle -|\rho|-\rangle$$

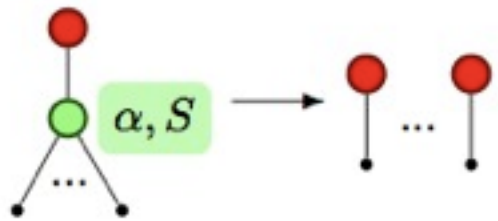
Equations for conditional diagrams



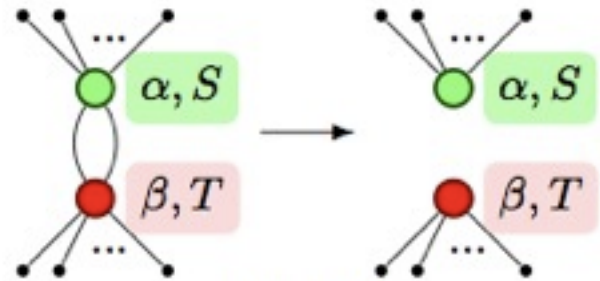
Equations for conditional diagrams



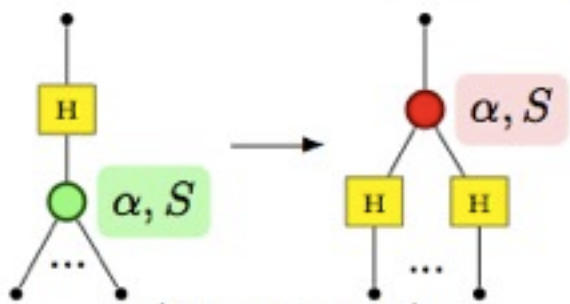
(bialgebra)



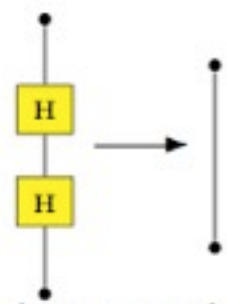
(copying)



(hopf)

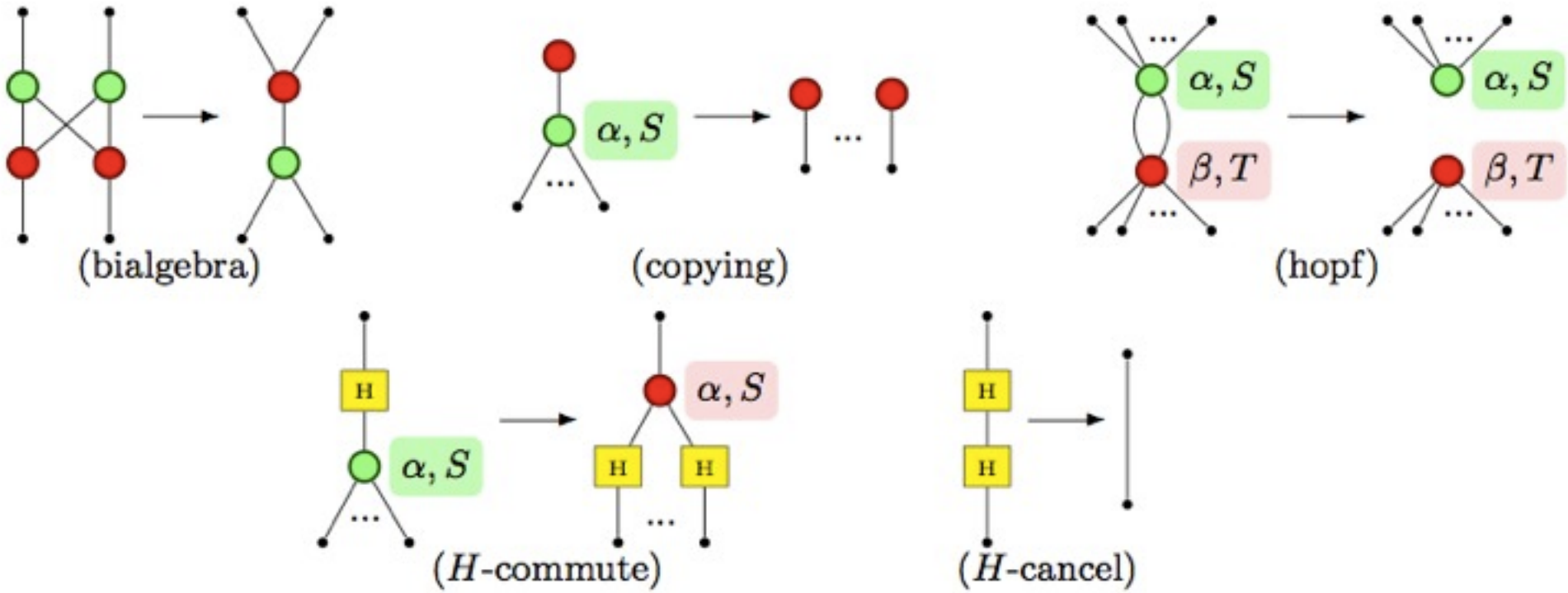


(H-commute)



(H-cancel)

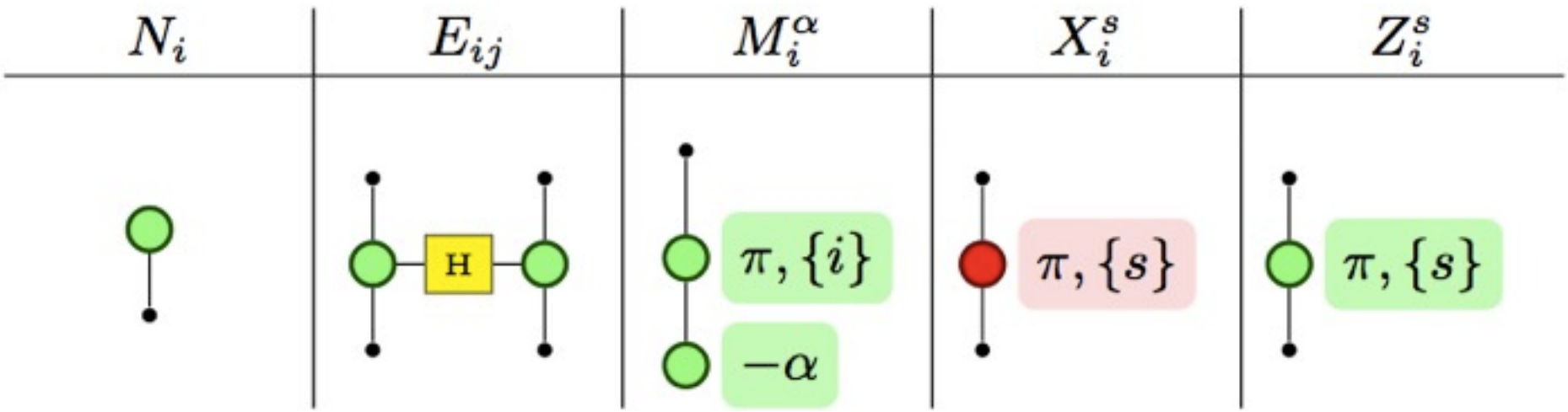
Equations for conditional diagrams



.... plus the same again with the colours exchanged

Translation from MC to diagram

We can translate any measurement pattern to diagram using the table below:



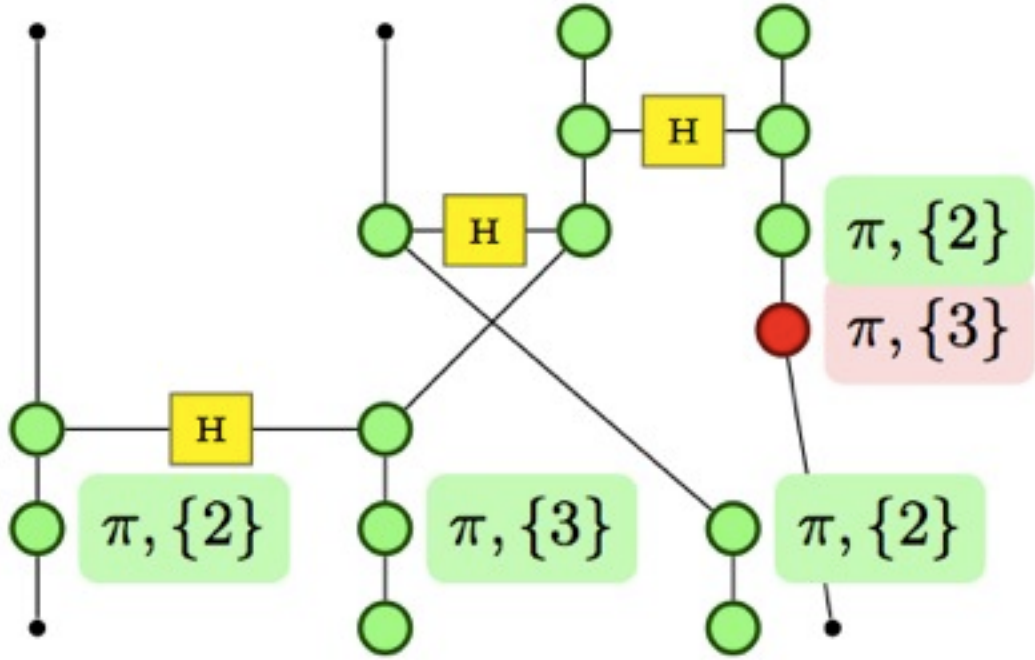
Example

The CNOT is computed by the pattern:

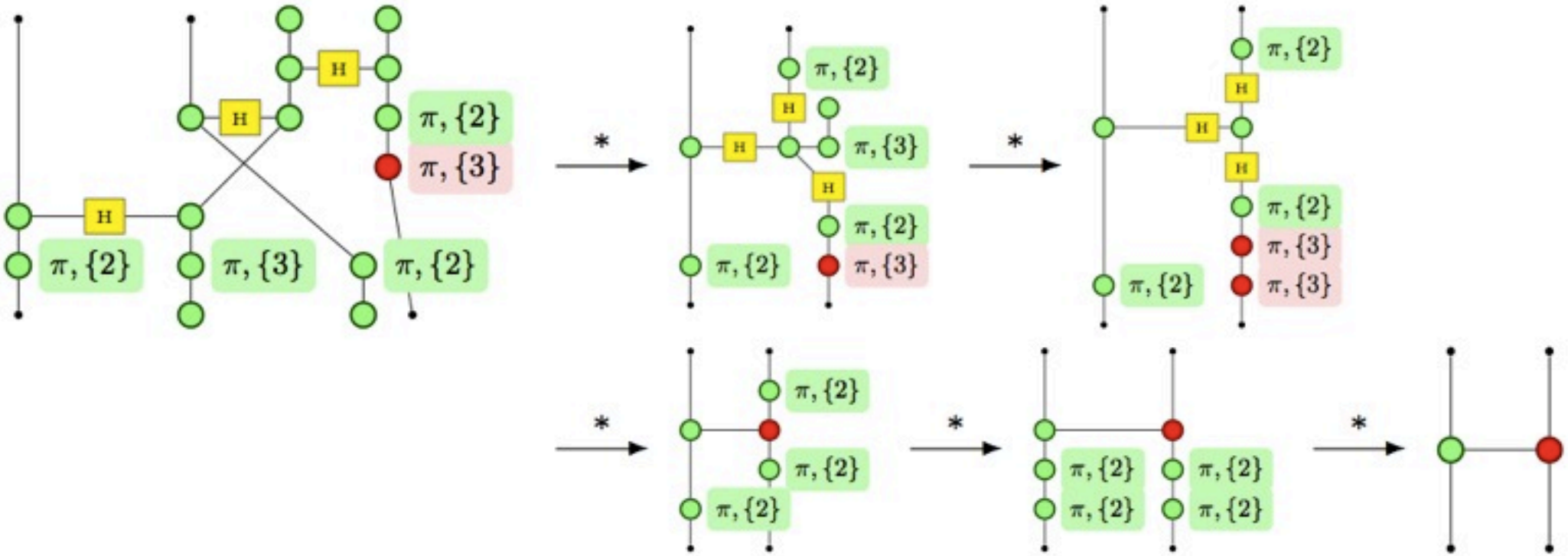
$$\mathfrak{P} = X_4^3 Z_4^2 Z_1^2 M_3^0 M_2^0 E_{13} E_{23} E_{34} N_3 N_4$$

Which yields the diagram:

$D_{\mathfrak{P}} =$



Example Rewrite

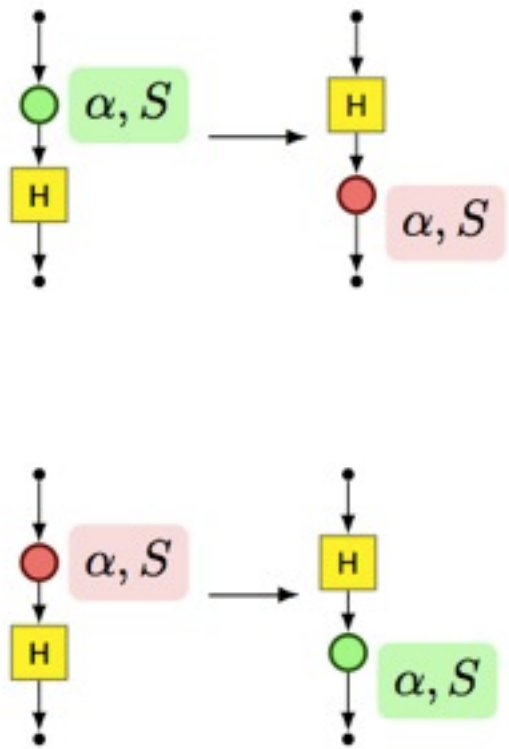


We removed all the conditional operations, therefore this **pattern** is deterministic.

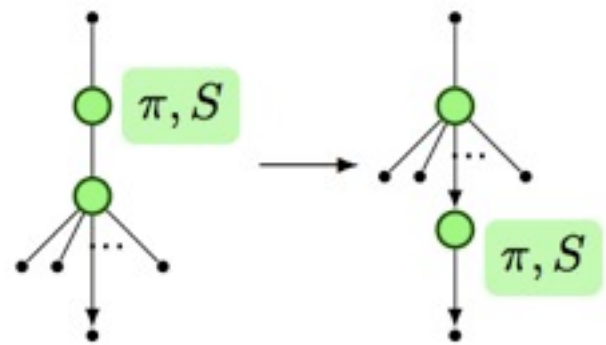
Can we do this in general?

Flow Strategy

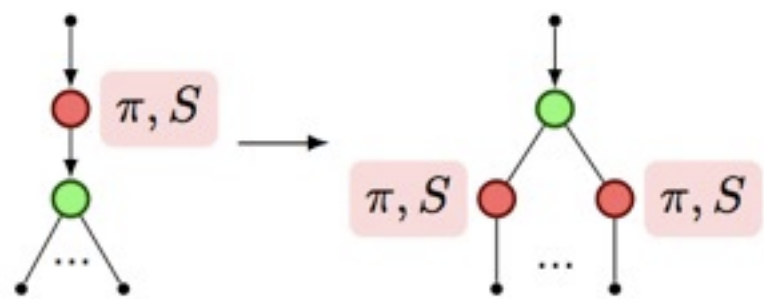
If the geometry has a flow we can define a rewrite strategy to propagate the "errors" forward:



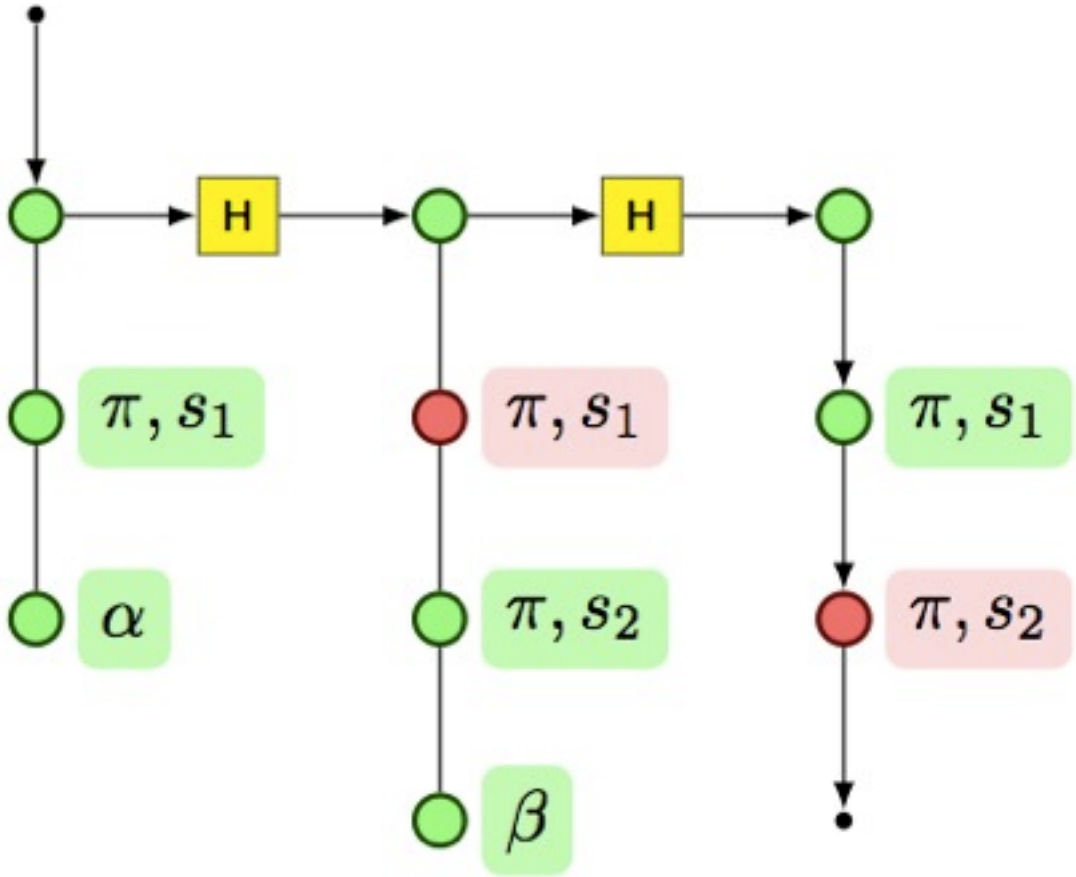
'π-green is attracted by the flow':



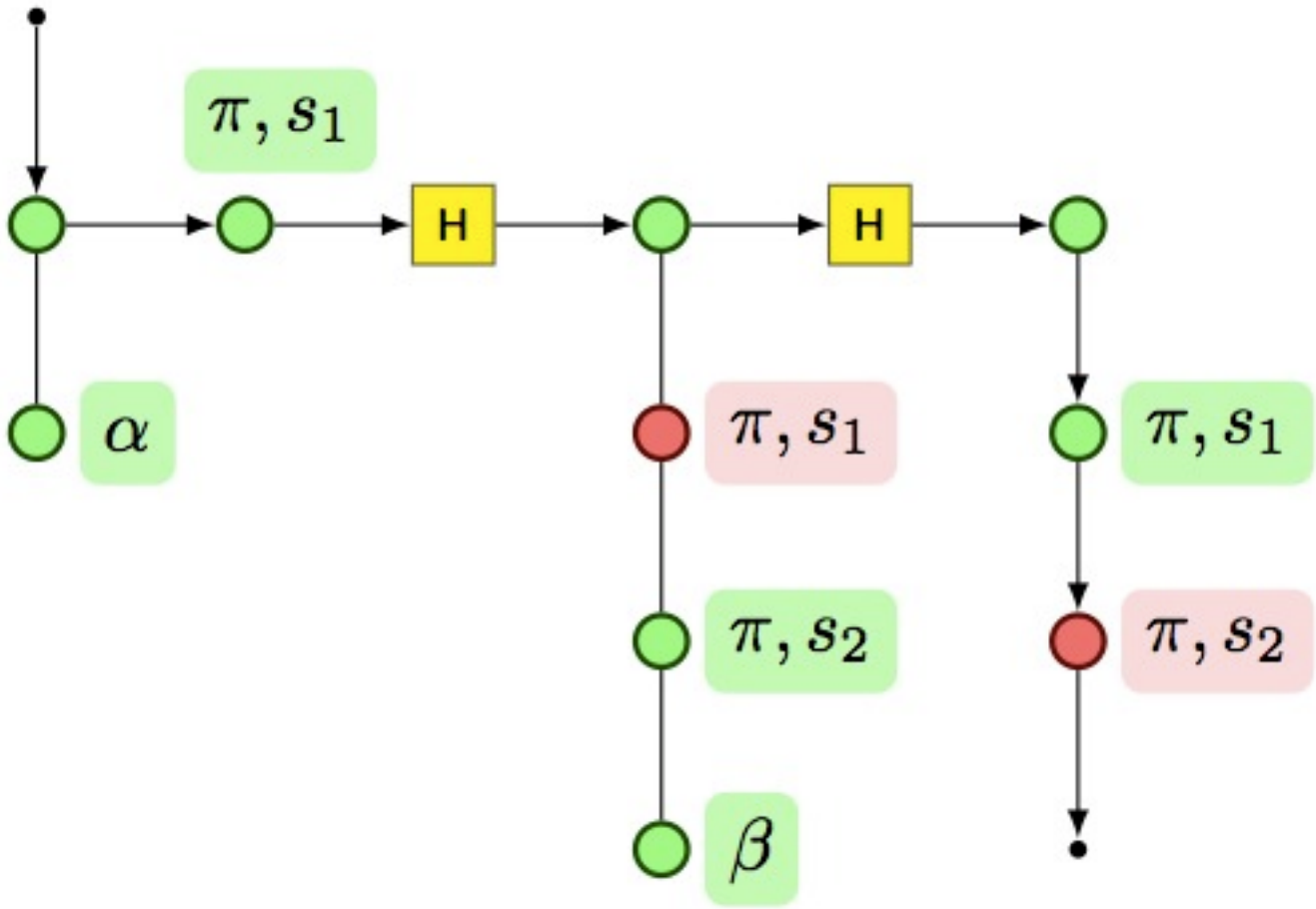
'π-red is pushed by the flow':



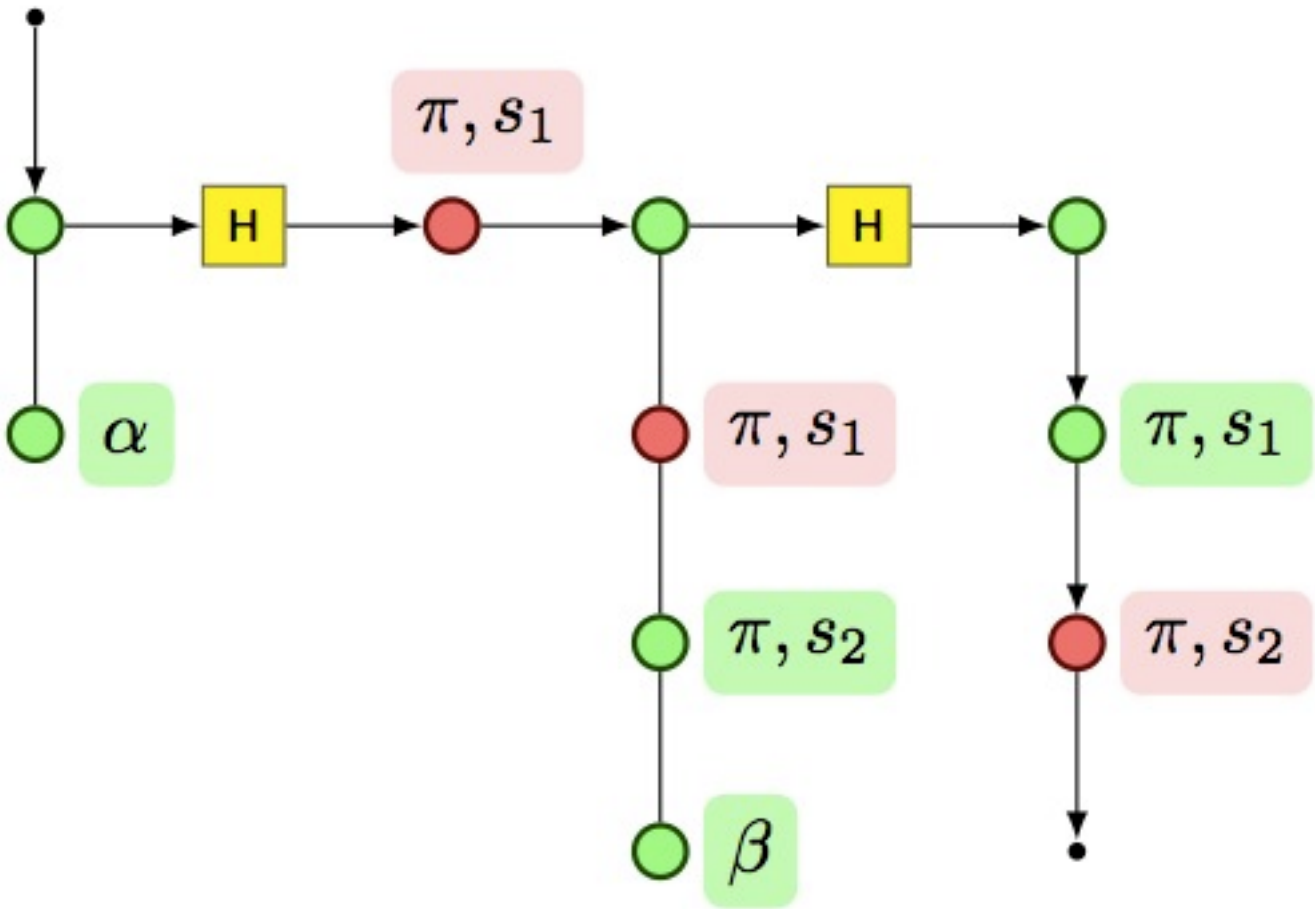
Flow Strategy



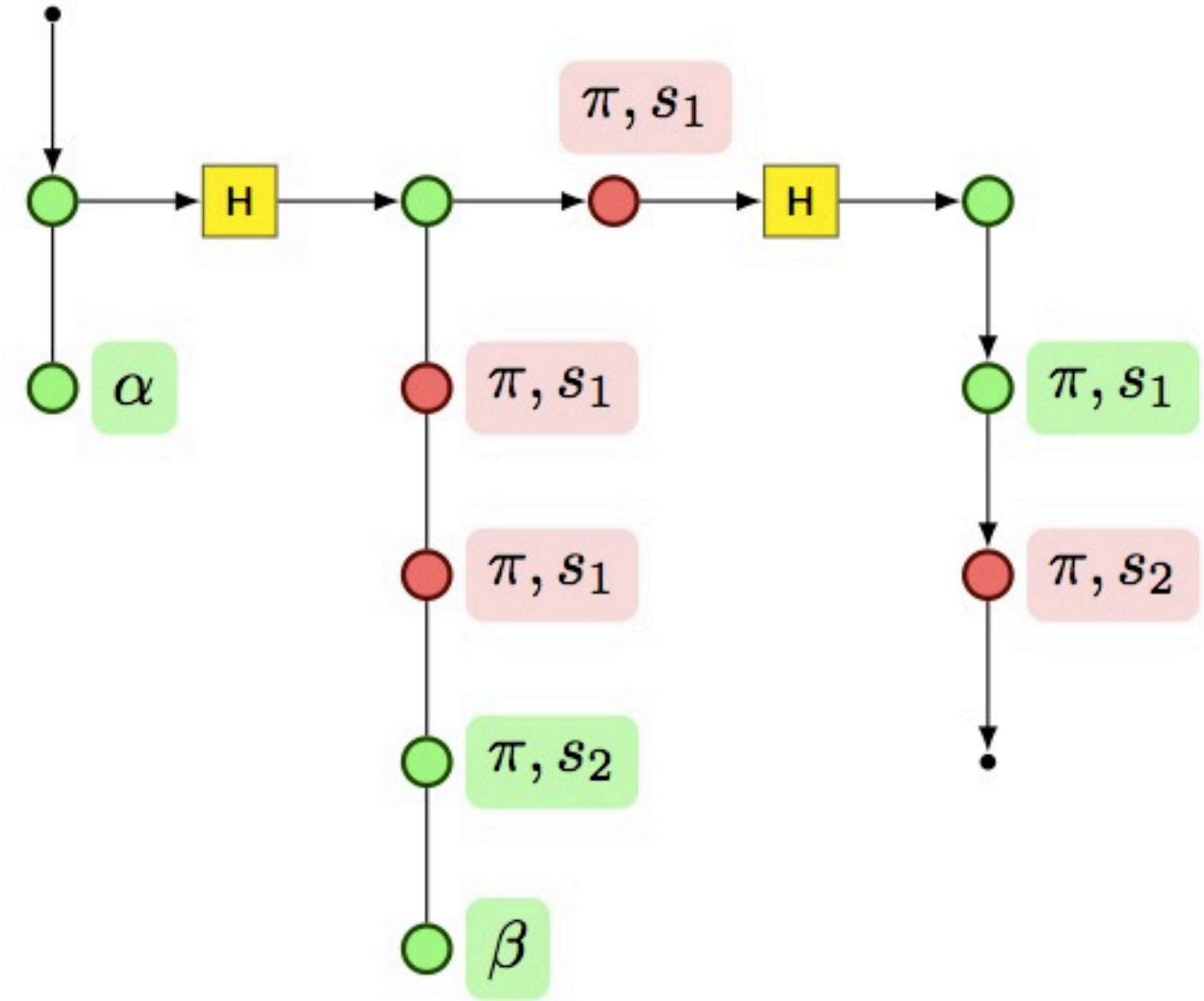
Flow Strategy



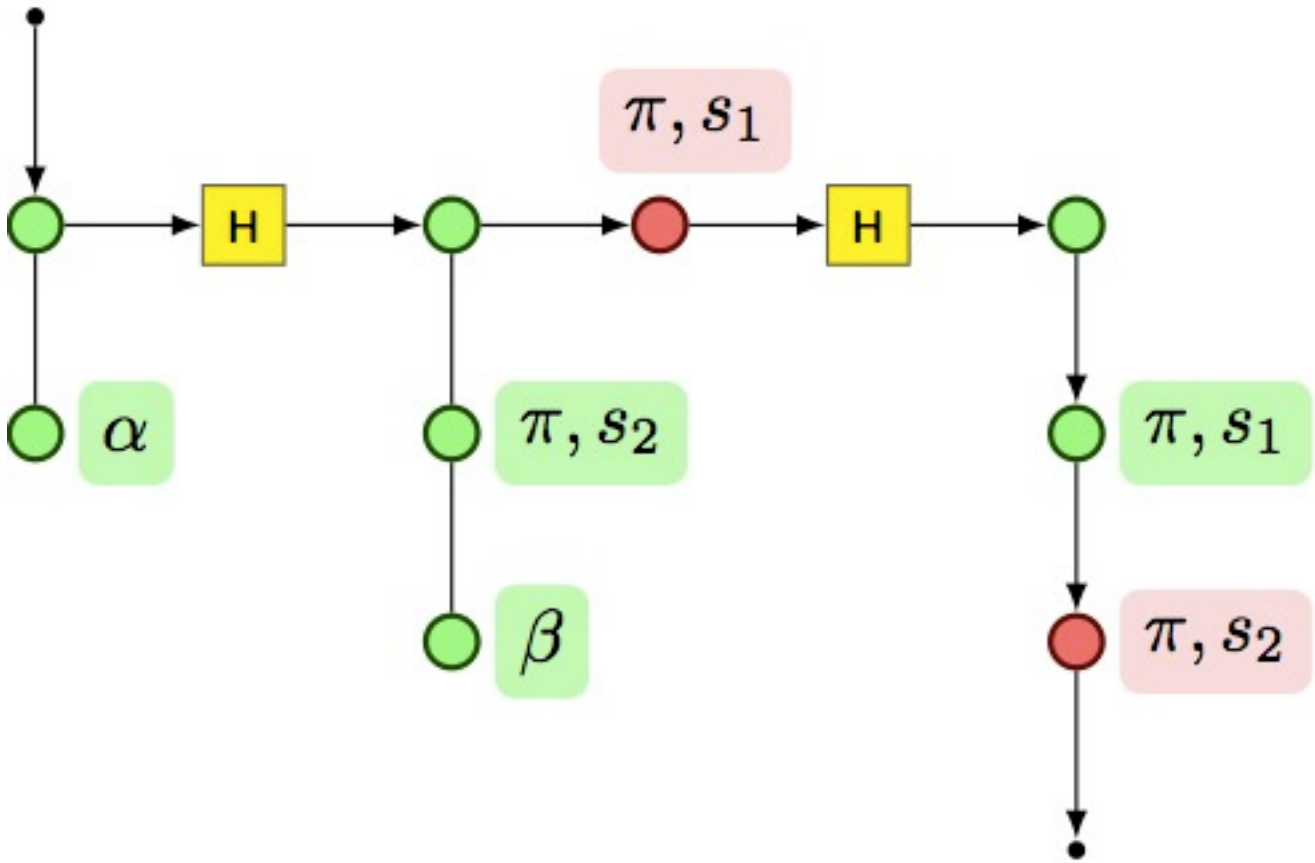
Flow Strategy



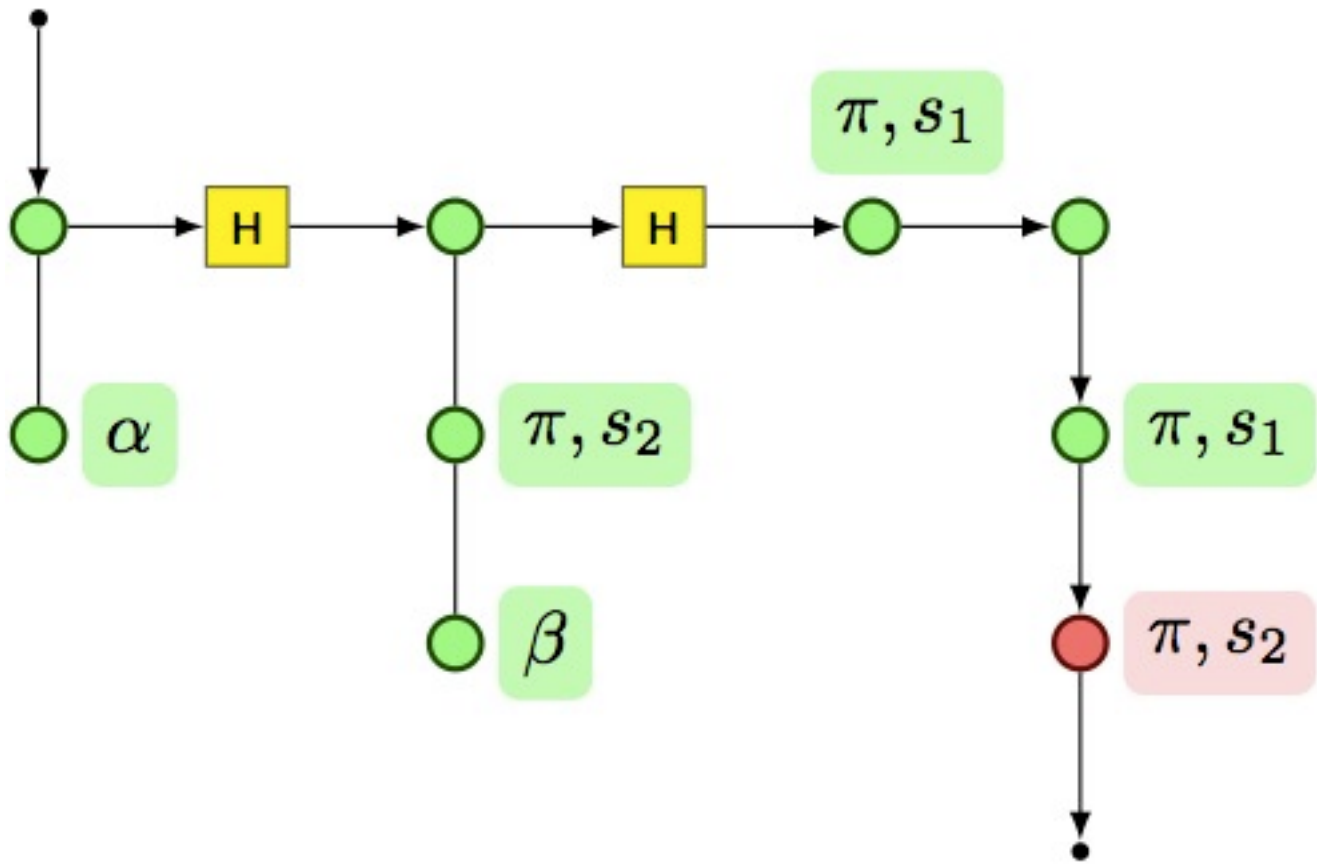
Flow Strategy



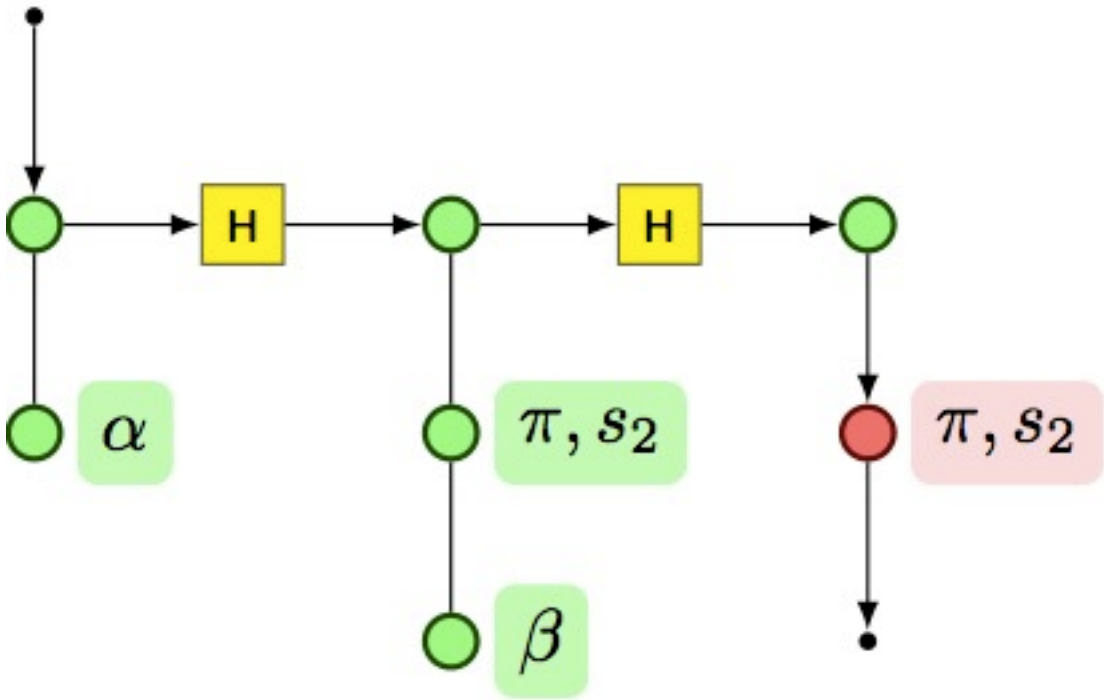
Flow Strategy



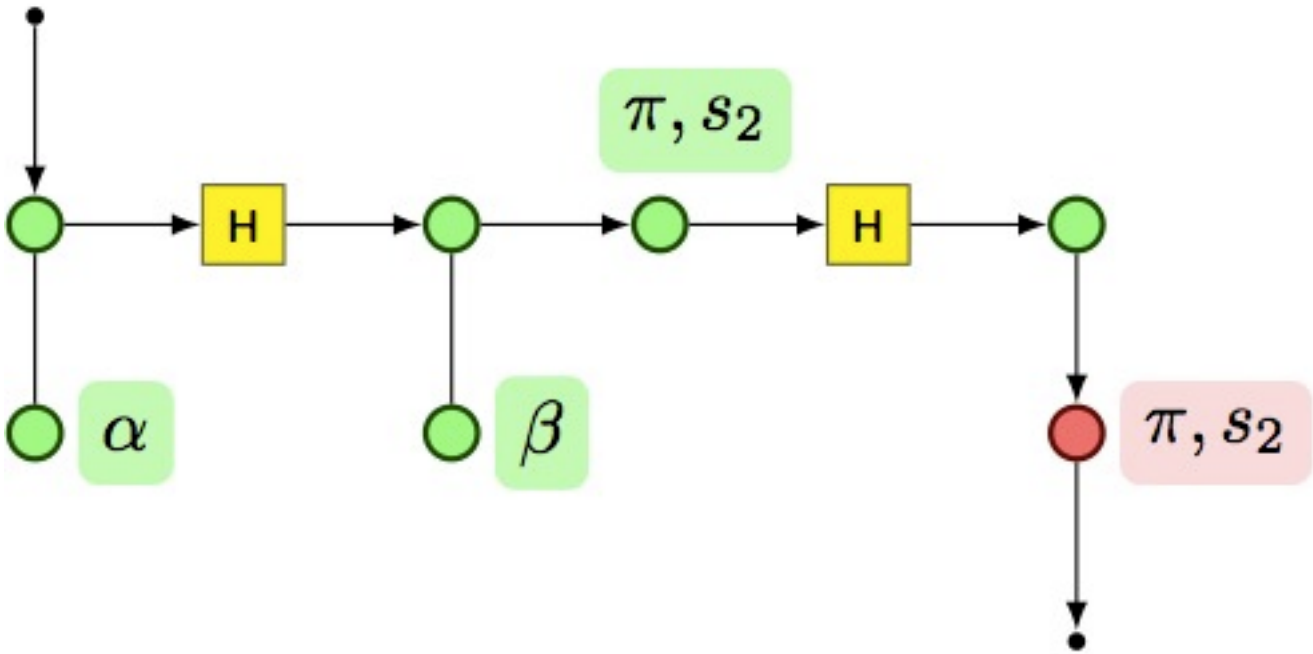
Flow Strategy



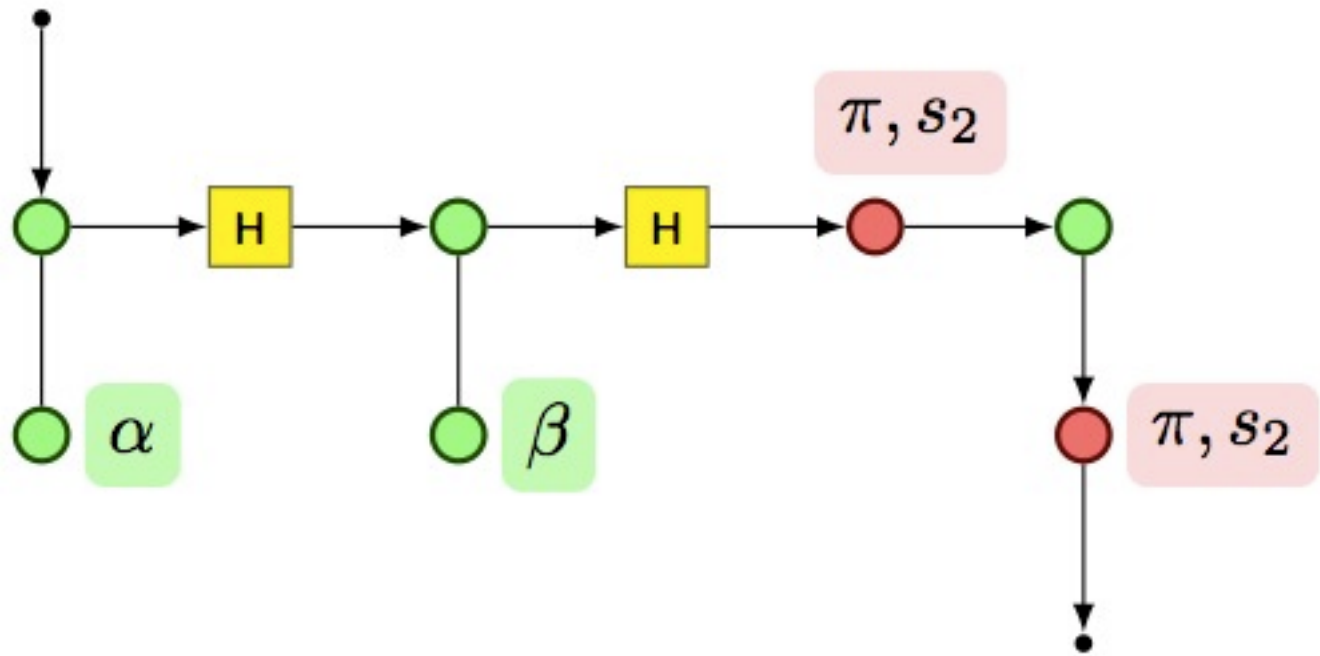
Flow Strategy



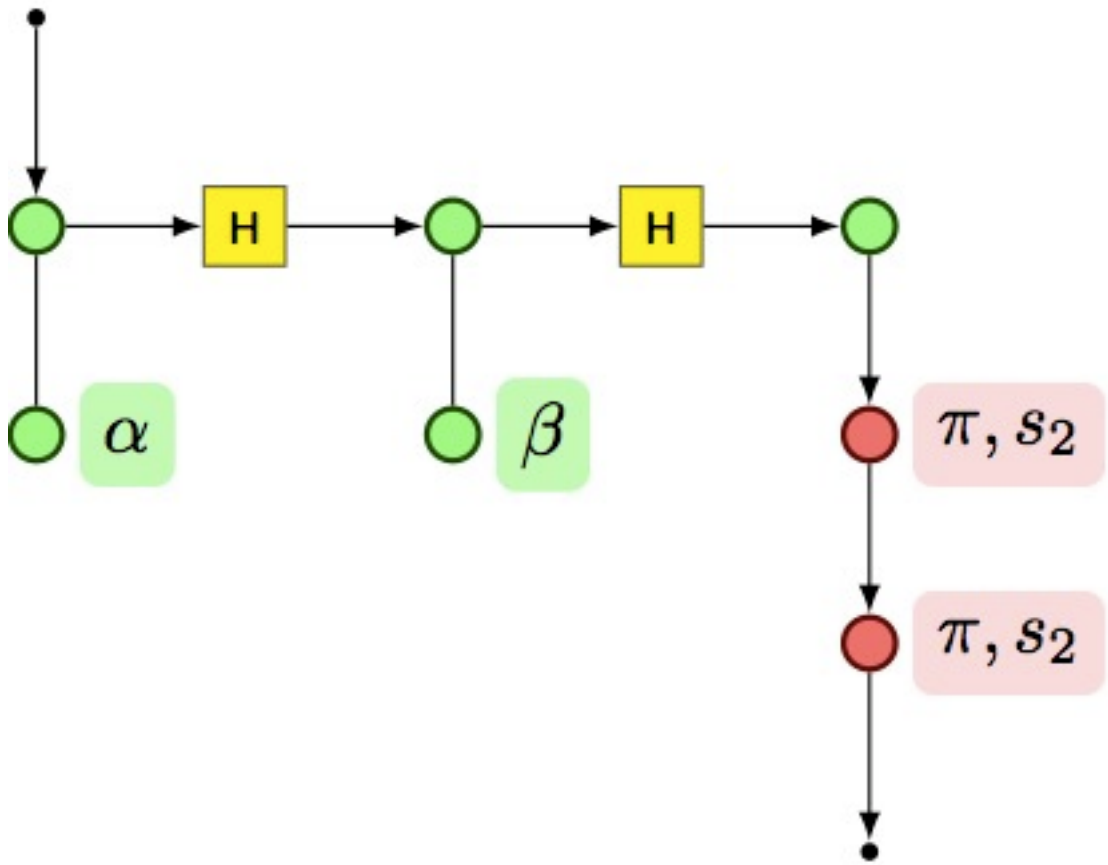
Flow Strategy



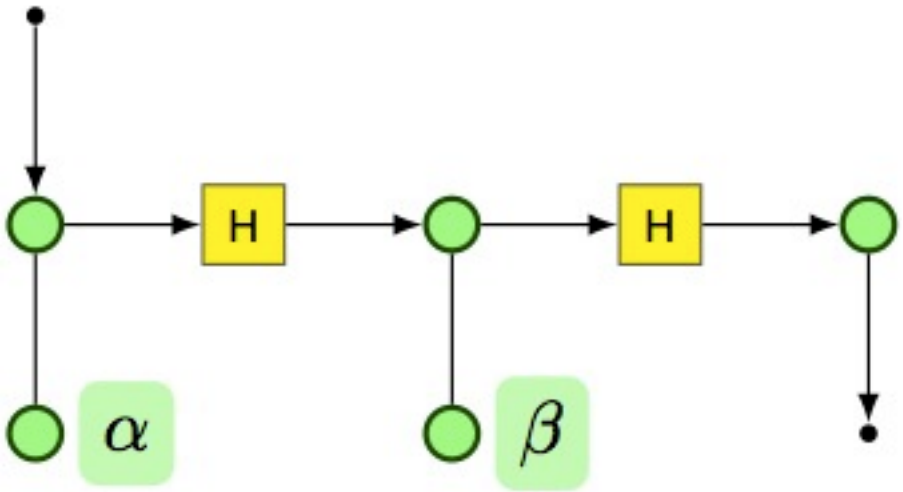
Flow Strategy



Flow Strategy



Flow Strategy



Flow Strategy

Thm: This strategy terminates in one of two states:

- All conditional operations removed => deterministic
- Not all removed, but their location reveals where extra corrections are needed.

THANKS

Duncan and Perdrix. "*Rewriting measurement-based quantum computations with generalised flow*".
ICALP 2010, Part II, LNCS 6199

This is the 75th slide.