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 $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$

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

$$\alpha |0\rangle + \beta |1\rangle$$

$p = \alpha^2 \quad p = \beta^2 \quad p = (\alpha + \beta)^2 / 2^2 \quad p = (\alpha - \beta)^2 / 2^2$

$|0\rangle \quad |1\rangle \quad |+\rangle \quad |-\rangle$
Quantum Measurements

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

$$Z = \{|0\rangle, |1\rangle\} \quad \text{and} \quad X = \left\{ \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \frac{|0\rangle + |1\rangle}{\sqrt{2}} \right\}$$

$$\alpha |0\rangle + \beta |1\rangle$$

$p=\alpha^2$  $p=\beta^2$  $p=(\alpha+\beta)/2^2$  $p=(\alpha-\beta)/2^2$

$|0\rangle$  $|1\rangle$  $|+\rangle$  $|\rangle$
The One-Way Model

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

$$CZ = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$
The One-Way Model

A graph state, coupled to some input qubits:

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

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

\[
|-\alpha\rangle := \frac{1}{\sqrt{2}} (|0\rangle - e^{i\alpha}|1\rangle)
\]
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
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.
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.
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.
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
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:

$$\Psi = \hat{X}_4^3 \hat{Z}_4^2 \hat{Z}_1^2 \hat{M}_3^0 \hat{M}_2^0 \hat{E}_{13} \hat{E}_{23} \hat{E}_{34} \hat{N}_3 \hat{N}_4$$

Each pattern implicitly defines a geometry \((G, I, O)\)
Geometry of a pattern

The following pattern computes the CNOT gate:

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

Each pattern implicitly defines a geometry \((G, I, O)\)
Geometry of a pattern

The following pattern computes the CNOT gate:

\[ \psi = X_4^3 Z_4^2 Z_1^2 M_3^0 M_2^0 E_{13} E_{23} E_{34} N_3 N_4 \]

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

Input vertices

Entanglement graph

1

2

3

4
Geometry of a pattern

The following pattern computes the CNOT gate:

\[ \Psi = \hat{X}_4^3 Z_4^2 Z_1^2 M_3^0 M_2^0 E_{13} E_{23} E_{34} N_3 N_4 \]

Each pattern implicitly defines a geometry \((G, I, O)\)
Flow and determinism

**Defn:** let $(G, I, O)$ be a geometry; a *flow* on $G$ is a pair $(f, <)$ where $f : O^C \rightarrow I^C$ 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:

|0⟩  |1⟩

Yay!  Boo!
How to Measure

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

\[
\begin{align*}
|0\rangle & \quad \text{Yay!} \\
|1\rangle & \quad \text{Yay!}
\end{align*}
\]
Flow

Input vertices

Output vertices
Flow

Input vertices

Output vertices

Boo!
Flow

Input vertices

Output vertices

Boo!

$v$

$f(v)$
Flow

Input vertices

Output vertices

Boo!

\( f(v) \)
Flow

Input vertices

Output vertices

\[ f(v) \]
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 \(g : O^C \rightarrow \mathcal{P}(I^C)\) is a function and \(<\) is a partial order s.t.

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

**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

- Measurement Pattern
  - "low-level program"
- Geometry
  - "entangled resource"
  - "graph state"
- Flow and GFlow
  - "correction strategy"

implicitly defines

can possess

jointly determine

Uniformly Deterministic Pattern
Determinism in MBQC

- Measurement Pattern
  - "low-level program"
- Geometry
  - "entangled resource"
  - "graph state"
- Flow and GFlow
  - "correction strategy"

Implicitly defines:

- can possess

Jointly determine:

- Uniformly Deterministic Pattern

Not necessarily the same!
Determinism in MBQC

- Measurement Pattern
  - "low-level program"
  - implicitly defines
    - Geometry
      - "entangled resource"
        - "graph state"
      - can possess
        - Flow and GFlow
          - "correction strategy"

- not necessarily the same!

- Uniformly Deterministic Pattern
  - jointly determine
  - translates to
    - Quantum Circuit

Ross Duncan • Flowincats • Oxford 2010
Determinism in MBQC

- Measurement Pattern
  - “low-level program”
  - implicitly defines

- Geometry
  - “entangled resource”
  - “graph state”
  - can possess

- Flow and GFlow
  - “correction strategy”

- Uniformly Deterministic Pattern
  - jointly determine

- Quantum Circuit with ancilla qubits
  - translates to

- not necessarily the same!
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:

- **Measurement Pattern**
  - "low-level program"

- **Geometry**
  - "entangled resource"
  - "graph state"

- **Flow and GFlow**
  - "correction strategy"

Translation:
- Graphical form
  - "direct translation"

Included in:
- Minimal Graphical Form
  - "annotated geometry"

Rewrites to:
Summary of this talk:

- **Quantum Circuit**
  - if original pattern is deterministic

- **Graphical form**
  - "direct translation"

- **Minimal Graphical Form**
  - "annotated geometry"

- **Quantum Circuit**
  - if original pattern is deterministic

- **Measurement Pattern**
  - "low-level program"

- **Geometry**
  - "entangled resource" "graph state"

- **Flow and GFlow**
  - "correction strategy"

- **"flow strategy"**

- **"direct translation"**

- **"annotated geometry"**

- **"graph state"**
Summary of this talk:

- **Measurement Pattern**
  - “low-level program”
- **Geometry**
  - “entangled resource”
  - “graph state”
- **Flow and GFlow**
  - “correction strategy”
- **Quantum Circuit**
  - “correction strategy”
  - if original pattern is deterministic
- **Graphical form**
  - “direct translation”
- **Minimal Graphical Form**
  - “annotated geometry”
- **Circuit-like form**
  - has flow

Additional terms:
- “graph state”
- “entangled resource”
Diagrams, Algebras and Circuits

Pessimistic Diagram Convention

PAST / HEAVEN

FUTURE / HELL
Generators for diagrams

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

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

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

(spider)

(anti-loop)

(identity)

(π-commute)
Equations
Equations

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

\[ m = \quad u = \]

Monoid Laws

\[ \]

Ros Duncan • Flowincats • Oxford 2010
Algebraic Properties

$$\delta = \begin{array}{c} \quad \\ \quad \end{array} \quad \epsilon = \begin{array}{c} \quad \\ \quad \end{array}$$

Comonoid Laws

$$\begin{array}{c} \quad \\ \quad \end{array} = \begin{array}{c} \quad \\ \quad \end{array} = \begin{array}{c} \quad \\ \quad \end{array} = \begin{array}{c} \quad \\ \quad \end{array}$$
Algebraic Properties

\[ \delta = \]

\[ \epsilon = \]

\[ m = \]

\[ u = \]

Bialgebra Laws

\[ = \]

\[ = \]

\[ = \]
Algebraic Properties

\[ \delta = \quad \epsilon = \quad m = \quad u = \]

Hopf Algebra Laws

(with trivial antipode)
Semantics for diagrams

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

\[
\begin{align*}
\begin{bmatrix}
\alpha
\end{bmatrix} &= \begin{cases} 
|0\rangle \otimes m \mapsto |0\rangle \otimes n \\
|1\rangle \otimes m \mapsto e^{i\alpha} |1\rangle \otimes n
\end{cases} \\
\begin{bmatrix}
\alpha
\end{bmatrix} &= \begin{cases} 
|+\rangle \otimes m \mapsto |+\rangle \otimes n \\
|\rangle \otimes m \mapsto e^{i\alpha} |-\rangle \otimes n
\end{cases} \\
\begin{bmatrix}
H
\end{bmatrix} &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}
\end{align*}
\]
Representing Qubits

\[ \begin{pmatrix} \bullet & \text{red} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle \]

\[ \begin{pmatrix} \bullet & \pi \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle \]

\[ \begin{pmatrix} \bullet & \text{green} \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = |+\rangle \]

\[ \begin{pmatrix} \bullet & \pi \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = |-\rangle \]
Representing Logic Gates

\[
\begin{bmatrix}
\pi
\end{bmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}
\]

\[
\begin{bmatrix}
\pi
\end{bmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}
\]
Representing Logic Gates

\[ \land Z = [ \text{Diagram} ] = [ \text{Diagram} ] = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \]

\[ \land X = [ \text{Diagram} ] = [ \text{Diagram} ] = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \]
Reasoning via rewrites:
Reasoning via rewrites:
Reasoning via rewrites:

\[ \pi \quad \pi \]
Which diagrams are circuits?

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

- All of its \( \alpha \), \( \beta \), 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 :(

![Diagram](image-url)
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

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

where \( V(G) \) and \( E(G) \) are the sets of vertices and edges of \( G \).
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, corresponding to the CZ interaction
- one output wire at each qubit

\[ |G_{\text{triangle}} \rangle = \]
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
Deteterminism 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:

```
  H  H  H  H  H  H
  H  H  H  H  H  H
  H  H  H  H  H  H
  H  H  H  H  H  H
  H  H  H  H  H  H
  H  H  H  H  H  H
  H  H  H  H  H  H
```
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:
What about \textbf{GFlow}?

- If a geometry has a GFlow, there exists a strategy that will transform the diagram into a diagram with flow, \textit{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:
**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:

![Diagram showing the strategy for refining the GFlow.](image-url)
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⟩

Yay!  
|1⟩
Should be this:

Yay!

\[
\begin{align*}
|+\rangle \\
\end{align*}
\]

Yay!

\[
\begin{align*}
|\pi\rangle \\
\end{align*}
\]

\[
\begin{align*}
|\pi\rangle
\end{align*}
\]

\[
\begin{align*}
|\pi\rangle
\end{align*}
\]
Example: Hadamard

Prepared $|+\rangle$ qubit

$\wedge Z$-gate

$H$

Projective measurement
Example: Hadamard
Example: Hadamard

Yay!
Example: Hadamard

Yay!
Example: Hadamard
Example: Hadamard

![Hadamard Diagram]

Boo!
Example: Hadamard

Boo!
Example: Hadamard

毒性

Boo!
Example: Hadamard

\[ \pi \]

Boo!
Example: Hadamard

Add correction here:
Example: Hadamard

Add correction here: \[ \pi \pi \]
Example: Hadamard

Add correction here:

Boo!
Conditional Operations

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

where \( S' \subseteq S \)
Semantics

**Defn:** a valuation of $S$ is a function $v : S \rightarrow \{0, 1\}$. For each valuation we can define a map $\hat{v} : D(S) \rightarrow D$ 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 $D(S)$ 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
Equations for conditional diagrams

.... plus the same again with the colours exchanged
Translation from MC to diagram

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

<table>
<thead>
<tr>
<th>$N_i$</th>
<th>$E_{ij}$</th>
<th>$M_i^\alpha$</th>
<th>$X_i^s$</th>
<th>$Z_i^s$</th>
</tr>
</thead>
</table>

- $N_i$: Green circle
- $E_{ij}$: Green circle, black dot, yellow box labeled $H$
- $M_i^\alpha$: Green circle, black dot, red circle labeled $\pi, \{i\}$, green circle labeled $-\alpha$
- $X_i^s$: Red circle, black dot labeled $\pi, \{s\}$
- $Z_i^s$: Green circle, black dot labeled $\pi, \{s\}$
Example

The CNOT is computed by the pattern:

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

Which yields the diagram:
We removed all the conditional operations, therefore this \textbf{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:

'\pi\text{-green is attracted by the flow}':

'\pi\text{-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
**Thm:** This strategy terminates in one of two states:

- All conditional operations removed $\Rightarrow$ deterministic
- Not all removed, but their location reveals where extra corrections are needed.