Denotational semantics for stabiliser quantum programs

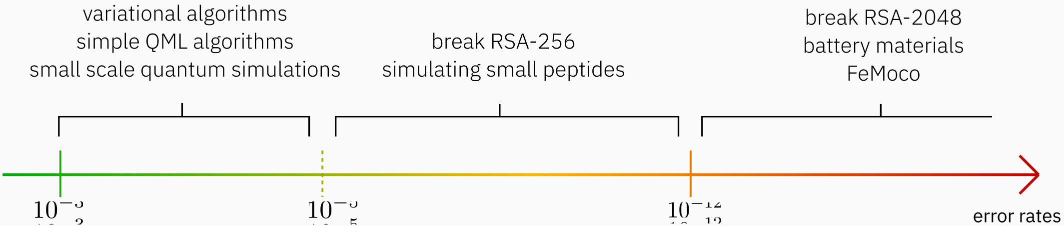
Robert I. Booth

Joint with: Cole Comfort

LIQCS, 17 June 2026

Noise in quantum hardware

error probability:	
transistor gates	qubits
0.000000000000001	0.001



Credit: Tamas Noszko, Adithya Sireesh

The fault-tolerant compilation problem

At the very least, we need to be able to implement the following operations:

- code initialisation
- syndrome measurements
- classically-controlled corrections
- logical operations

...and we need to do all of this fault-tolerantly!

Stabiliser Programming Language

More pragmatically, we want to be able to reason about the *fault-tolerance* properties of a simple programming language for stabiliser quantum mechanics:

$$c, d \ ::= c \,;\, d \mid \overbrace{\mathbf{init}\ \underline{\mathrm{x}} \mid \underline{\mathbf{y}} = A * \underline{\mathbf{x}} \mid \mathbf{disc}\ \underline{\mathrm{x}}}^{\textit{classical}} \mid \overbrace{\mathbf{qinit}\ \underline{\mathrm{x}} \mid \underline{\mathbf{x}} \mathrel{*}= U}^{\textit{quantum}}$$
$$\mid \underbrace{\mathbf{meas}\ \underline{\mathrm{x}}}_{\textit{measurement}} \mid \underbrace{\mathbf{ctrl}_P\ \underline{\mathrm{x}}\ \underline{\mathrm{y}}}_{\textit{classical control}}$$

Stabiliser Programming Language

More pragmatically, we want to be able to reason about the *fault-tolerance* properties of a simple programming language for stabiliser quantum mechanics:

$$
\begin{aligned}
c, d \;::=\; & c \,⨾\, d \mid \overbrace{\textbf{init}\ \underline{\mathrm{x}} \mid \underline{\mathbf{y}} = A * \underline{\mathbf{x}} \mid \textbf{disc}\ \underline{\mathrm{x}}}^{\textit{classical}} \mid \overbrace{\textbf{qinit}\ \underline{\mathrm{x}} \mid \underline{\mathbf{x}} \mathrel{*{=}} U}^{\textit{quantum}} \\
& \mid \underbrace{\textbf{meas}\ \underline{\mathrm{x}}}_{\textit{measurement}} \mid \underbrace{\textbf{ctrl}_P\ \underline{\mathrm{x}}\ \underline{\mathrm{y}}}_{\textit{classical control}}
\end{aligned}
$$

We give denotational semantics tailored to stabiliser fault-tolerance, with the long-term goal of supporting the development of

- invariants of fault-tolerant programs
- program synthesis and optimisation under fault-tolerance
- formal methods for fault-tolerance

The stabiliser theory

Stabiliser codes

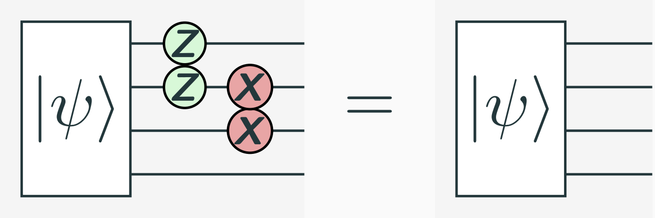
Subgroups of the Pauli group on n qubits that do not contain -1 are in bijection with subspaces of $\mathbb{C}^{2^n}$ called *stabiliser codes*.

The stabiliser theory

Stabiliser codes

Subgroups of the Pauli group on n qubits that do not contain -1 are in bijection with subspaces of $\mathbb{C}^{2^n}$ called *stabiliser codes*.

The code is given by the subspace of states such that



for all elements of the stabiliser group.

Stabiliser states ($\sim$codewords)

Stabiliser states

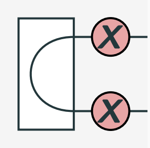
When the stabiliser group is *maximal*, the associated code is 1-dimensional, i.e. it is a *stabiliser state*.

Stabiliser states ($\sim$codewords)

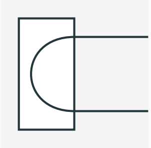
Stabiliser states

When the stabiliser group is *maximal*, the associated code is 1-dimensional, i.e. it is a *stabiliser state*.

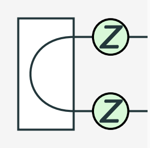
For example, the Bell states are stabiliser states:



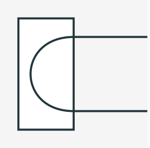
=



and



=

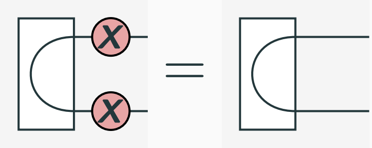


!

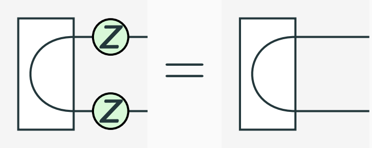
Stabiliser states

When the stabiliser group is *maximal*, the associated code is 1-dimensional, i.e. it is a *stabiliser state*.

For example, the Bell states are stabiliser states:



and



!

Stabiliser theory

The stabiliser theory has as processes those initialisations, transformations, and measurements that map stabiliser states to stabiliser states.

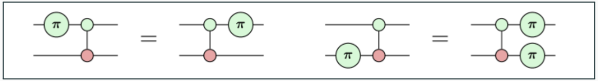
Semantics overview

$$\begin{tikzcd}
\textit{pure}\ \mathrm{QM} \arrow[r, "\text{CPM construction}"] & \textit{mixed}\ \mathrm{QM} \arrow[r, "\text{split } \dagger\text{-idempotents}"] & \mathrm{QM\ with}\ \textit{classical types} \\
\mathrm{FdHilb} \arrow[r] & \mathrm{MatAlg}_{CP} \arrow[r] & \mathrm{FdC^*\text{-}Alg}_{CP} \\
|\psi\rangle \arrow[r] & \rho \arrow[r] & \rho \otimes d
\end{tikzcd}$$

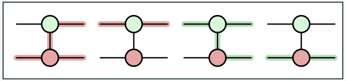
Peter Selinger. **"Dagger Compact Closed Categories and Completely Positive Maps: (Extended Abstract)".** In: ().
Peter Selinger. **"Idempotents in Dagger Categories".** In: ().

Pauli pushing in all its forms

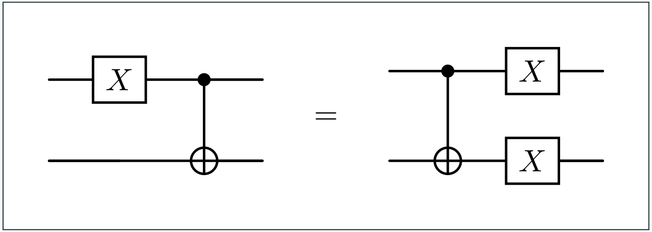
Pauli pushing



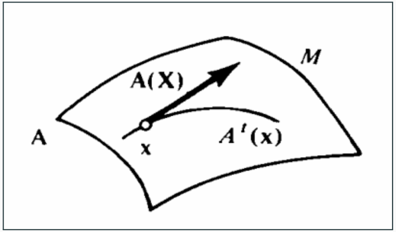
Pauli webs = stabiliser flows



Pauli propagation

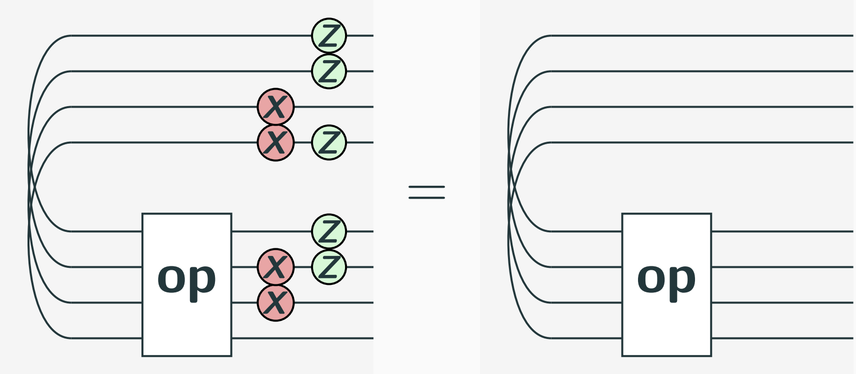


Lagrangian relations



Pure stabiliser processes

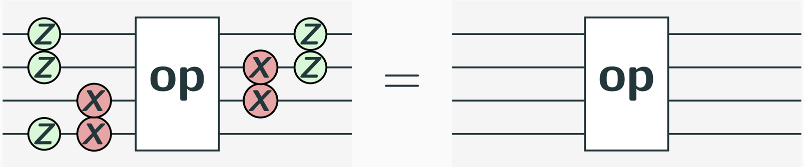
We can therefore define a compact-closed category of stabiliser processes—those whose name is a stabiliser state:



Cole Comfort and Aleks Kissinger. **"A Graphical Calculus for Lagrangian Relations".** In: (visited on 06/19/2023).

Pure stabiliser processes

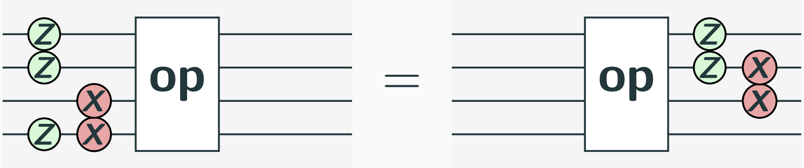
We can therefore define a compact-closed category of stabiliser processes—those whose name is a stabiliser state:



Cole Comfort and Aleks Kissinger. **"A Graphical Calculus for Lagrangian Relations"**. In: (visited on 06/19/2023).

Pure stabiliser processes

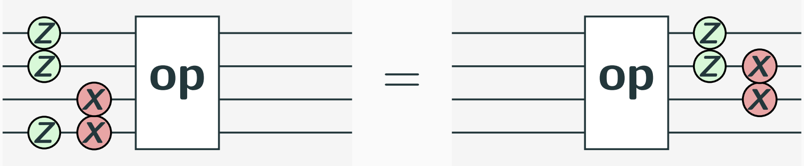
We can therefore define a compact-closed category of stabiliser processes—those whose name is a stabiliser state:

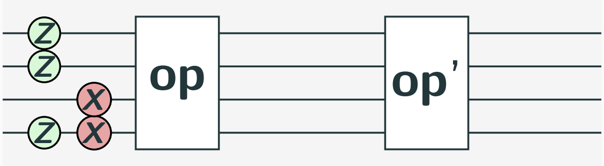


Cole Comfort and Aleks Kissinger. **"A Graphical Calculus for Lagrangian Relations"**. In: (visited on 06/19/2023).

Pure stabiliser processes

We can therefore define a compact-closed category of stabiliser processes—those whose name is a stabiliser state:

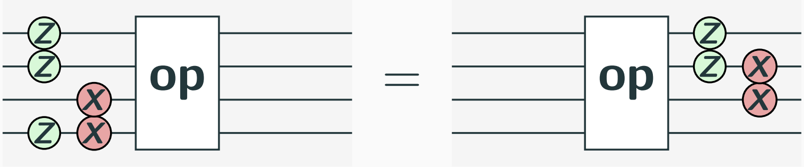


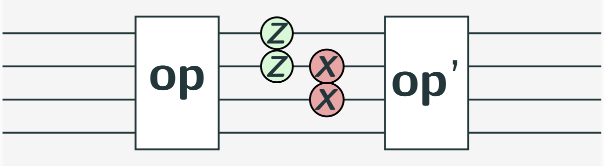


Cole Comfort and Aleks Kissinger. **"A Graphical Calculus for Lagrangian Relations".** In: (visited on 06/19/2023).

Pure stabiliser processes

We can therefore define a compact-closed category of stabiliser processes—those whose name is a stabiliser state:

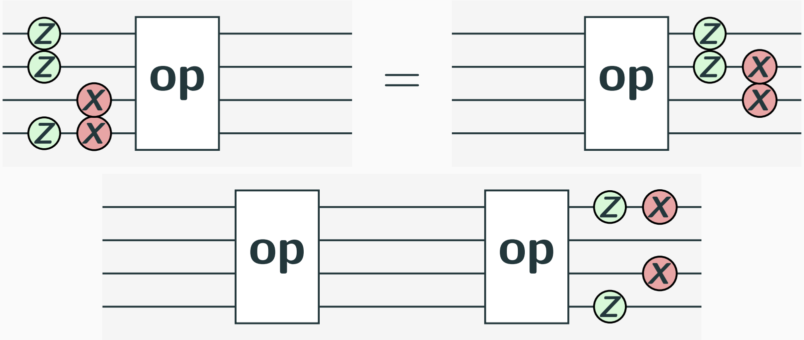




Cole Comfort and Aleks Kissinger. **"A Graphical Calculus for Lagrangian Relations".** In: (visited on 06/19/2023).

Pure stabiliser processes

We can therefore define a compact-closed category of stabiliser processes—those whose name is a stabiliser state:

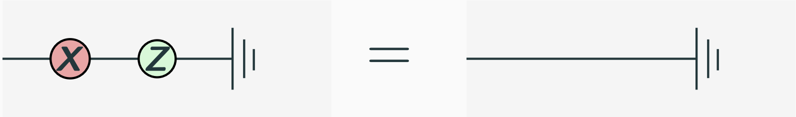


Cole Comfort and Aleks Kissinger. **"A Graphical Calculus for Lagrangian Relations".** In: (visited on 06/19/2023).

Semantics overview

$$
\begin{tikzcd}
\textbf{operational} & & \textbf{denotational} \\
\text{pure stabiliser theory} \arrow[rr, leftrightarrow] \arrow[d, "\text{CPM construction}"'] & & \text{affine Lagrangian relations} \arrow[d, "\text{CPM construction}"] \\
\text{mixed stabiliser theory} \arrow[rr, leftrightarrow] \arrow[d, "\text{split } \dagger\text{-idempotents}"'] & & \text{affine coisotropic relations} \arrow[d, "\text{split } \dagger\text{-idempotents}"] \\
\text{stabiliser theory with classical types} \arrow[rr, leftrightarrow] & & \text{affine relations with symplectic modality}
\end{tikzcd}
$$

Semantics of trace



Semantics of trace

$\rightsquigarrow$ the trace is just a discard!

Semantics of measurements

Semantics of measurements

$\rightsquigarrow$ maps the 2-dimensional space of red/green tokens to the 1-dimensional space of bit-flips

The physical operations are total

In quantum theory, the physical operations are the *trace-preserving* ones:

The physical operations are total

In quantum theory, the physical operations are the *trace-preserving* ones:

$\rightsquigarrow$ the physical stabiliser operations are the total relations!

Summary of results

Full definability

For all typed environments Γ and Δ and morphisms $f : [\![\Gamma]\!] \to [\![\Delta]\!]$ in $\mathrm{AR}_{\mathrm{qc}}$, there exists a well-formed judgement $\Gamma \vdash c \triangleright \Delta$ such that

Full abstraction

Well-formed judgements c and d are observationally equivalent if and only if their denotations are equal $[\![c]\!] = [\![d]\!]$.

Future work

- How do we recover this semantics for the qubit case? What about bosonic quantum computing?

Future work

- How do we recover this semantics for the qubit case? What about bosonic quantum computing?

- How do we go beyond affine classical control?

Future work

- How do we recover this semantics for the qubit case? What about bosonic quantum computing?
- How do we go beyond affine classical control?
- How can we use these semantics for fault-tolerant compilation?

Future work

- How do we recover this semantics for the qubit case? What about bosonic quantum computing?
- How do we go beyond affine classical control?
- How can we use these semantics for fault-tolerant compilation? What is the compositional / algebraic / combinatorial structure of fault-tolerance?

The paper

Denotational semantics for stabiliser quantum programs

Robert I. Booth
University of Oxford, United Kingdom

Cole Comfort
Université Paris-Saclay, CNRS, ENS Paris-Saclay, Inria, CentraleSupélec, Laboratoire Méthodes Formelles

Abstract

The stabiliser fragment of quantum theory is a foundational building block for quantum error correction and the fault-tolerant compilation of quantum programs. In this article, we develop a sound, universal and complete denotational semantics for stabiliser operations which include measurement and classical control and in which quantum error-correcting codes are first-class objects. The operations are interpreted as certain *affine relations*, offering a significantly simpler alternative to the standard operator-algebraic semantics of quantum programs.

We demonstrate the power of the resulting semantics by describing a small, proof-of-concept assembly language for stabiliser programs with fully-abstract denotational semantics. These results pave the way to formally verified fault-tolerant quantum compilers based on the stabiliser fragment.

2012 ACM Subject Classification Theory of computation → Denotational semantics; Theory of computation → Quantum computation theory; Theory of computation → Program reasoning

Keywords and phrases stabiliser theory, denotational semantics, quantum error-correction, symplectic linear algebra, categorical semantics, quantum programming languages

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23

The symplectic rosetta stone

$$\gamma : \mathbb{F}_2^n \oplus \mathbb{F}_2^n \longrightarrow \mathcal{P}_n$$

$$(x|z) \longmapsto \prod_{j=1}^{n} X_j^{x_j} Z_j^{z_j} \qquad \text{so that} \qquad \gamma(s|t)\gamma(x|z) = e^{i\pi s^T z}\gamma(s+x|t+z)$$

Stabiliser	**Symplectic**
Product: $\gamma(s\|t)\,\gamma(x\|z)$	Sum: $(s+x \mid t+z)$
Commuting Paulis	$s^T z + t^T x = 0$
Subgroup of $\mathcal{P}_n$	Subspace of $\mathbb{F}_2^n \oplus \mathbb{F}_2^n$
Abelian subgroup	Isotropic subspace
Maximal abelian subgroup	Lagrangian subspace
Signs	Subspace character: $\langle v, \cdot\rangle$ (affine part)