Denotational semantics for stabiliser quantum programs

Robert I. Booth

SPLS, 3 December 2025

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

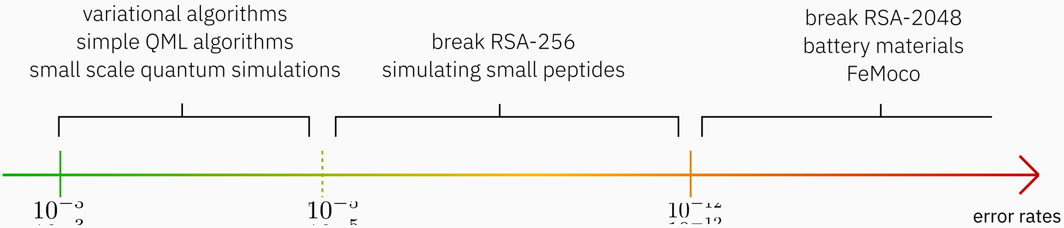
Abstract

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, classically-controlled Pauli operators, and affine classical operations, in which quantum error-correcting codes are first-class objects. The operations are interpreted as certain *affine relations* over finite fields. This offers a conceptually motivated and computationally-tractable alternative to the standard operator-algebraic semantics of quantum programs (whose time complexity grows exponentially as the state space increases in size).

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.

Errors in quantum computers

error probability:	
transistor gates	qubits
0.000000000000001	0.001



Credit: Tamas Noszko, Adithya Sireesh

Quantum error-correcting codes

Repetition code

$$\text{codewords:} \qquad 0 \mapsto 0000 \qquad\qquad 1 \mapsto 1111$$

Quantum error-correcting codes

Repetition code $=$ invariance under permutations

$$\text{codewords:} \qquad 0 \mapsto 0000 \qquad\qquad 1 \mapsto 1111$$

Quantum error-correcting codes

Quantum repetition code

codewords: $|0\rangle \mapsto |0000\rangle \qquad |1\rangle \mapsto |1111\rangle$

Quantum repetition code

codewords: $|0\rangle \mapsto |0000\rangle \qquad |1\rangle \mapsto |1111\rangle$

codevector: $\alpha\,|0\rangle + \beta\,|1\rangle \mapsto \alpha\,|0000\rangle + \beta\,|1111\rangle$

Quantum error-correcting codes

Quantum repetition code $=$ vectors invariant under a group representation

$$\text{codewords:} \qquad |0\rangle \mapsto |0000\rangle \qquad\qquad |1\rangle \mapsto |1111\rangle$$

$$\text{codevector:} \qquad \alpha\,|0\rangle + \beta\,|1\rangle \mapsto \alpha\,|0000\rangle + \beta\,|1111\rangle$$

Quantum error-correcting codes

Quantum repetition code $=$ vectors invariant under a group representation

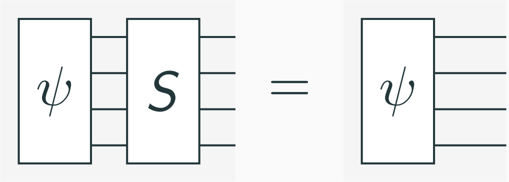
$$\text{codewords:} \qquad |0\rangle \mapsto |0000\rangle \qquad\qquad |1\rangle \mapsto |1111\rangle$$

$$\text{codevector:} \qquad \alpha\,|0\rangle + \beta\,|1\rangle \mapsto \alpha\,|0000\rangle + \beta\,|1111\rangle$$

For QEC, more complicated error types $\implies$ more complicated codes $\implies$ more complicated symmetry groups

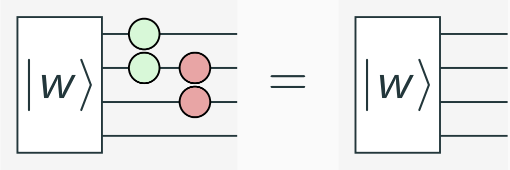
Stabiliser codes

Stabiliser codes are special QEC codes whose symmetries are *locally generated*:

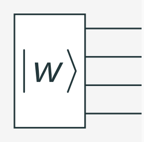


Stabiliser codes

Stabiliser codes are special QEC codes whose symmetries are *locally generated*:

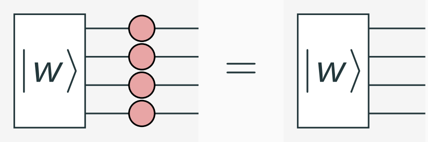


Stabiliser codes



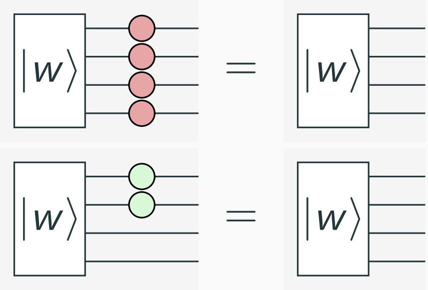
$$\begin{array}{cccc} |0000\rangle & |0001\rangle & |0010\rangle & |0011\rangle \\ |0100\rangle & |0101\rangle & |0110\rangle & |0111\rangle \\ |1000\rangle & |1001\rangle & |1010\rangle & |1011\rangle \\ |1100\rangle & |1101\rangle & |1110\rangle & |1111\rangle \end{array}$$

Stabiliser codes



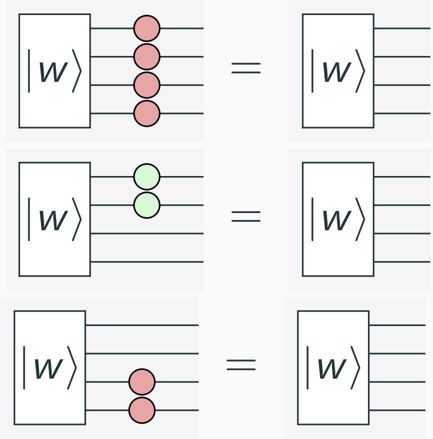
$\lvert 0000\rangle$	$\cancel{\lvert 0001\rangle}$	$\cancel{\lvert 0010\rangle}$	$\lvert 0011\rangle$
$\cancel{\lvert 0100\rangle}$	$\lvert 0101\rangle$	$\lvert 0110\rangle$	$\cancel{\lvert 0111\rangle}$
$\cancel{\lvert 1000\rangle}$	$\lvert 1001\rangle$	$\lvert 1010\rangle$	$\cancel{\lvert 1011\rangle}$
$\lvert 1100\rangle$	$\cancel{\lvert 1101\rangle}$	$\cancel{\lvert 1110\rangle}$	$\lvert 1111\rangle$

Stabiliser codes



$\lvert 0000\rangle$	$\cancel{\lvert 0001\rangle}$	$\cancel{\lvert 0010\rangle}$	$\lvert 0011\rangle$
$\cancel{\lvert 0100\rangle}$	$\lvert 0101\rangle$	$\lvert 0110\rangle$	$\cancel{\lvert 0111\rangle}$
$\cancel{\lvert 1000\rangle}$	$\lvert 1001\rangle$	$\lvert 1010\rangle$	$\cancel{\lvert 1011\rangle}$
$\lvert 1100\rangle$	$\cancel{\lvert 1101\rangle}$	$\cancel{\lvert 1110\rangle}$	$\lvert 1111\rangle$

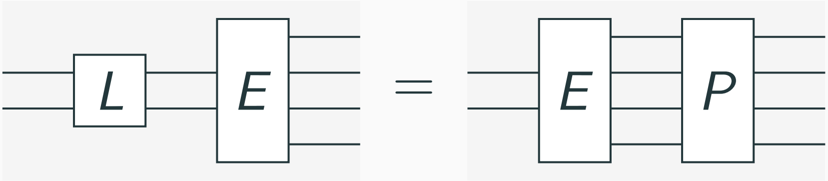
Stabiliser codes



$\vert 0000\rangle$	$\cancel{\vert 0001\rangle}$	$\cancel{\vert 0010\rangle}$	$\vert 0011\rangle$
$\cancel{\vert 0100\rangle}$	$\cancel{\vert 0101\rangle}$	$\cancel{\vert 0110\rangle}$	$\cancel{\vert 0111\rangle}$
$\cancel{\vert 1000\rangle}$	$\cancel{\vert 1001\rangle}$	$\cancel{\vert 1010\rangle}$	$\cancel{\vert 1011\rangle}$
$\vert 1100\rangle$	$\cancel{\vert 1101\rangle}$	$\cancel{\vert 1110\rangle}$	$\vert 1111\rangle$

Fault-tolerant quantum computation

Unlike classical error correction, quantum operations are very unreliable. We need to be able to act on *encoded* data:



This task is very non-trivial!

Syndrome extraction

How do we determine if the symmetry of the system *has* been broken i.e. an error has occured?

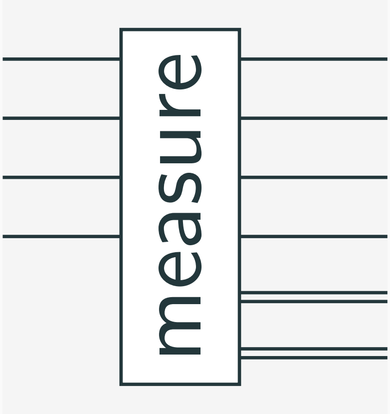
Syndrome extraction

How do we determine if the symmetry of the system *has* been broken i.e. an error has occured?

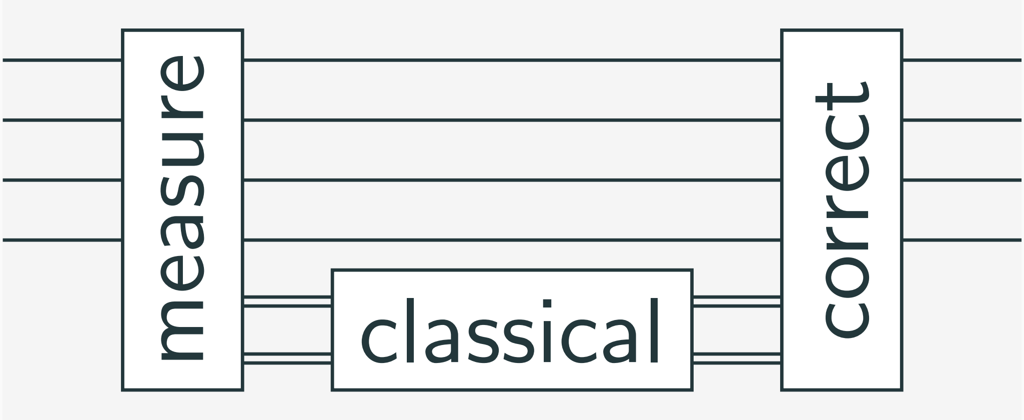
$$
\begin{tikzcd}
 & & |0\rangle \\
\alpha |0\rangle + \beta |1\rangle \arrow[r] & \textbf{meas} \arrow[ur, "\substack{\text{outcome } 0 \\ \text{prob } |\alpha|^2}"] \arrow[dr, "\substack{\text{outcome } 1 \\ \text{prob } |\beta|^2}"'] & \\
 & & |1\rangle
\end{tikzcd}
$$

Syndrome extraction

We need to to syndrome extraction:



Classical control and corrections



The fault-tolerant compilation problem

So (at the very least) we need to be able to implement the following operations:

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

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

Stabiliser Programming Language

$$c, d \ ::= c \fatsemi d \mid \overbrace{\textbf{init}\ \underline{\mathrm{x}} \mid \underline{\mathbf{y}} = A * \underline{\mathbf{x}} \mid \textbf{disc}\ \underline{\mathrm{x}}}^{\textit{classical}} \mid \underbrace{\textbf{qinit}\ \underline{\mathrm{x}} \mid \underline{\mathbf{x}} *\!\!= U}_{\textit{quantum}}$$

$$\mid \ \overbrace{\textbf{meas}\ \underline{\mathrm{x}}}^{\textit{measurement}} \ \mid \ \underbrace{\textbf{ctrl}_P\ \underline{\mathrm{x}}\ \underline{\mathrm{y}}}_{\textit{classical control}}$$

Semantics overview

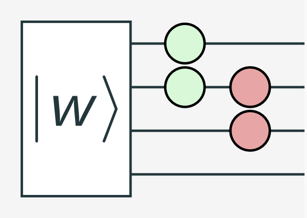
$\rightsquigarrow$ categorical semantics for Selinger's QPL (Quantum Programming Language)

$$\textit{pure}\ \mathrm{QM} \xrightarrow{\text{CPM construction}} \textit{mixed}\ \mathrm{QM} \xrightarrow{\text{split } \dagger\text{-idempotents}} \mathrm{QM\ with}\ \textit{measurements}$$

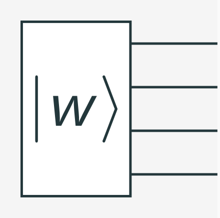
$$\mathrm{FdHilb} \longrightarrow \mathrm{MatAlg}_{CP} \longrightarrow \mathrm{FdC^*\text{-}Alg}_{CP}$$

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

Semantics of pure stabiliser processes (∼codewords)



$=$

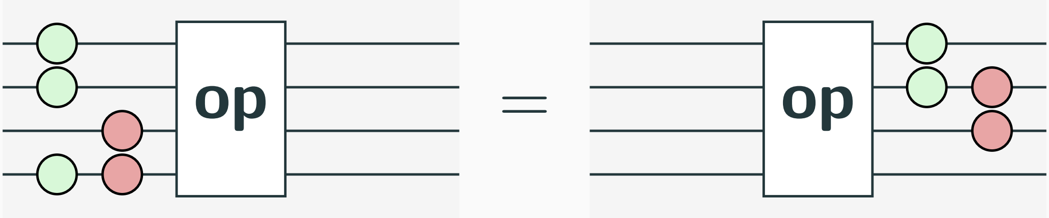


Cole Comfort and Aleks Kissinger. **"A Graphical Calculus for Lagrangian Relations".** In: (visited on 06/19/2023).

Semantics of pure stabiliser processes ($\sim$codewords)

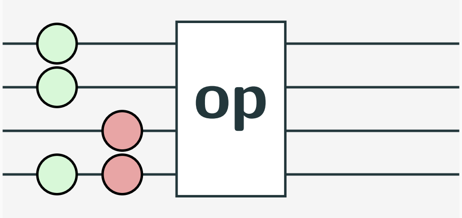
Cole Comfort and Aleks Kissinger. **"A Graphical Calculus for Lagrangian Relations".** In: (visited on 06/19/2023).

Semantics of pure stabiliser processes (∼codewords)

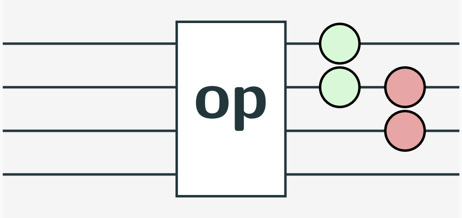


Cole Comfort and Aleks Kissinger. **"A Graphical Calculus for Lagrangian Relations".** In: (visited on 06/19/2023).

Semantics of pure stabiliser processes ($\sim$codewords)



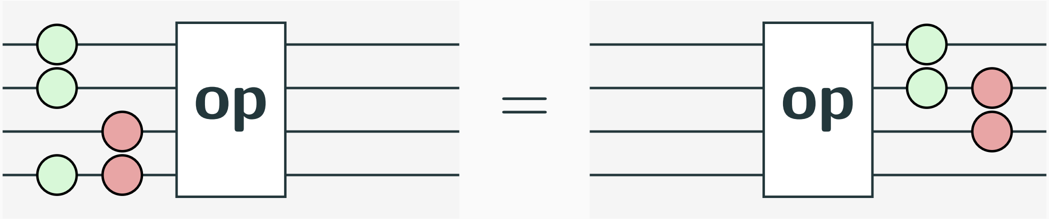
$=$



$\rightsquigarrow$ semantics in the category of relations (think *Petri nets* or *token-passing* in Geometry of Interaction)

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

Semantics of pure stabiliser processes ($\sim$codewords)



$\rightsquigarrow$ semantics in the category of relations (think *Petri nets* or *token-passing* in Geometry of Interaction)

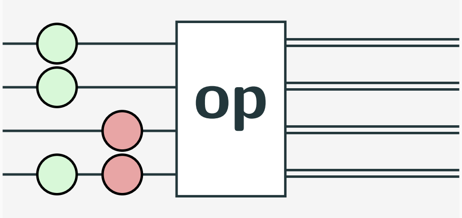
$$\text{pure stabiliser theory} \longleftrightarrow \text{affine Lagrangian relations}$$

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

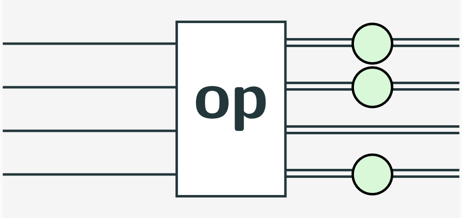
Semantics overview

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

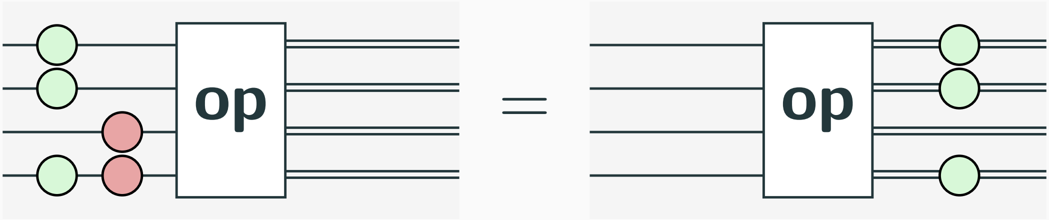
Semantics of measurements



$=$



Semantics of measurements



$\rightsquigarrow$ maps the 2-dimensional space of red/green tokens to the 1-dimensional space of green tokens (on each wire)

Some caveats / work in progress

- no qubits: for qubits, the symplectic representation on which we rely breaks down

Some caveats / work in progress

- no qubits: for qubits, the symplectic representation on which we rely breaks down
- only linear (or really, affine) classical control: this makes it impossible for now to reason about decoding, because the vast majority of decoders are nonlinear

Some caveats / work in progress

- no qubits: for qubits, the symplectic representation on which we rely breaks down
- only linear (or really, affine) classical control: this makes it impossible for now to reason about decoding, because the vast majority of decoders are nonlinear
- What is the compositional structure of fault-tolerance?

Some caveats / work in progress

- no qubits: for qubits, the symplectic representation on which we rely breaks down
- only linear (or really, affine) classical control: this makes it impossible for now to reason about decoding, because the vast majority of decoders are nonlinear
- What is the compositional structure of fault-tolerance?
- Are there program logics or type systems for verifying/enforcing fault-tolerance properties of quantum programs?