{Algebraic, Operational , Denotational} Semantics for Classical Controlled Quantum Communication

Theo Wang, Sam Staton
University of Oxford

What?

- … Quantum Communication

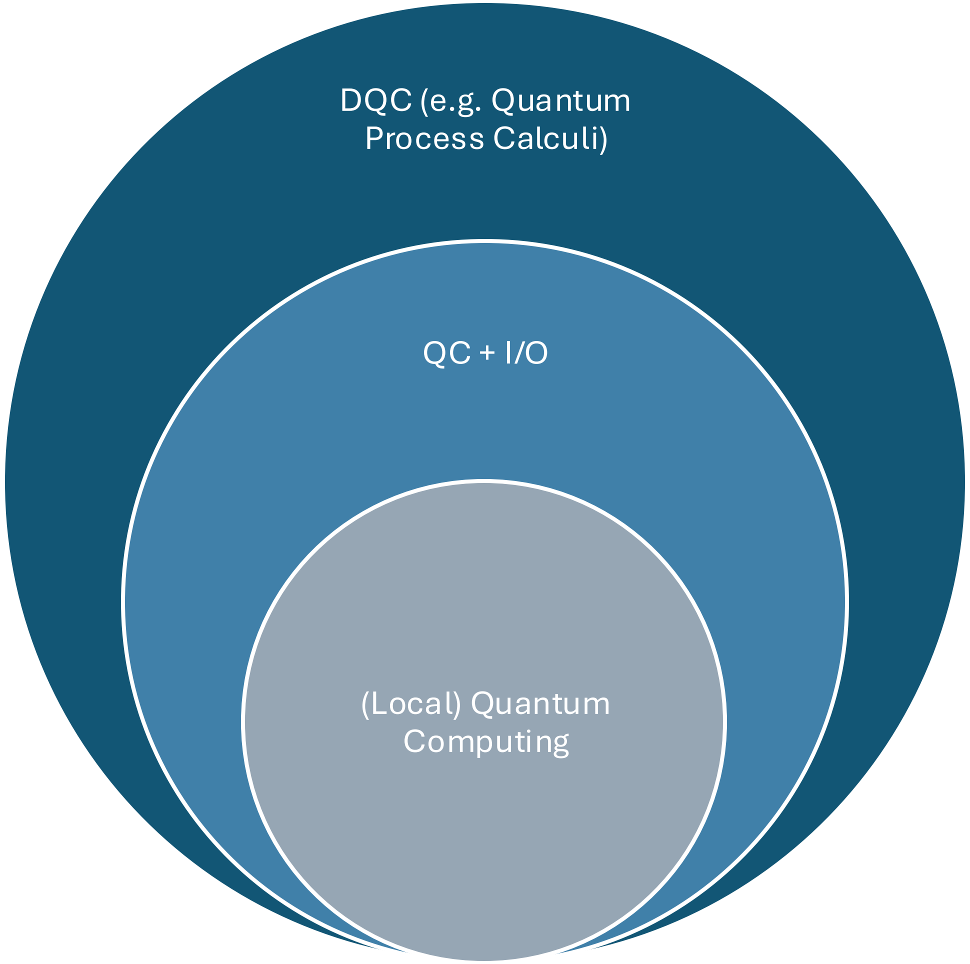
```
r = in(); q = apply_U(r); out(q)
```

- Classically-Controlled …

```
r = in(); q = new(|0>);
if(measure(r) == |0>) then out(q) else skip
```

Why?

- DQC
- Quantum Protocols

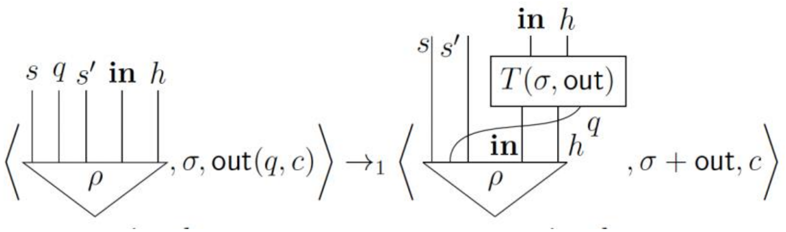


How?

$$\mathsf{new}(a.\mathsf{in}(b.x(a,b))) = \mathsf{in}(b.\mathsf{new}(a.x(a,b)))$$
$$\mathsf{apply}_X(a, a.\mathsf{measure}(a,x,y)) = \mathsf{measure}(a,y,x)$$

Algebraic Theory

Sound



Operational Semantics

Sound

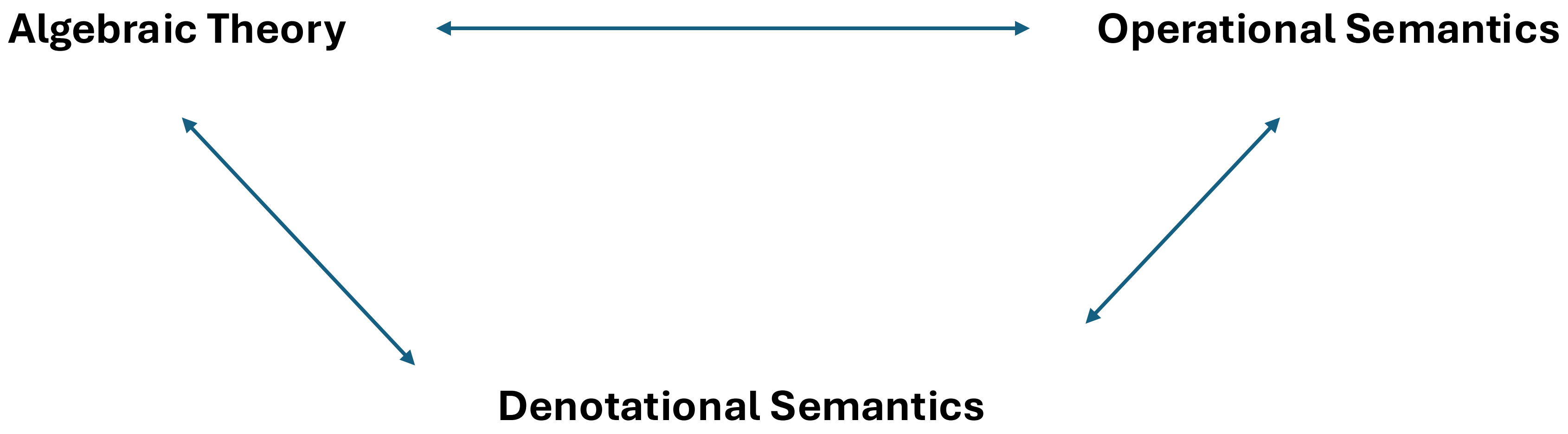
$$T(X) = \bigoplus_{\sigma \in \{\mathsf{in},\mathsf{out}\}^*} \mathsf{Aux}(\sigma) \otimes X.$$

Denotational Semantics

Adequate &
Fully Abstract

Qubit Quantum Computing

Starting Point



Algebraic Theory QUANTUM (POPL '15)

Operations $\mathbf{op} : (p \mid m_1 ... m_k)$

$\mathsf{new} : (0 \mid 1)$ $\quad$ $\mathsf{measure} : (1 \mid 0, 0)$

$\mathsf{apply}_U : (n \mid n)$

Terms $x_1 : m_1 ... x_k : m_k \mid q_1 ... q_p \vdash c$

$\mathsf{new}(q.c)$ $\quad$ $\mathsf{measure}(q, c_0, c_1)$

$\mathsf{apply}_U(\vec{q}, \vec{q}'.c)$

e.g.

$\mathsf{new}(q.\mathsf{apply}_H(q, q.\mathsf{measure}(q, x, y)))$

Equations $\Gamma \mid \Delta \vdash c = c'$

e.g.

$x : 0, y : 0 \mid \cdot \vdash \mathsf{new}(q.\mathsf{measure}(q, x, y)) = x$

Models $[\![c]\!] : [\![p]\!] \to \sum_i [\![m_i]\!]$

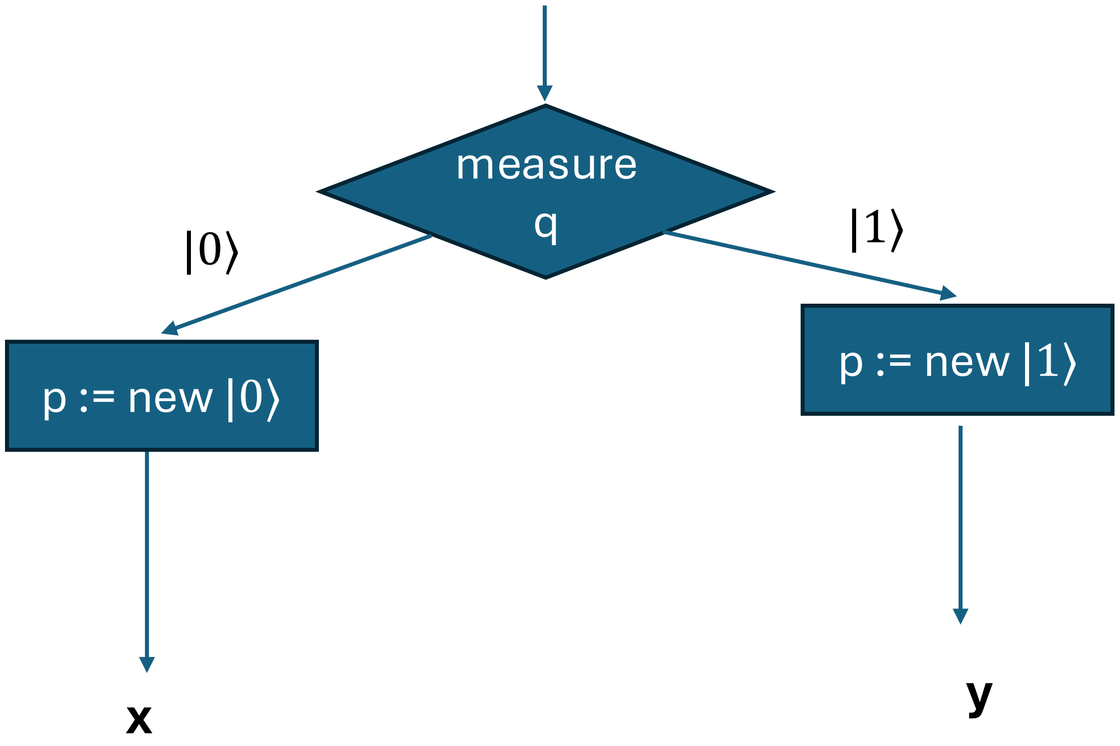
Fully Complete Denotational Model:

$[\![c]\!] : \mathbf{qbit}^{\otimes p} \to \bigoplus_i \mathbf{qbit}^{\otimes m_i}$

in the category of CP(TP) maps

Feature: Classical Branching

$x : 1, y : 1 \mid q \vdash \mathsf{measure}(q, \mathsf{new}_0(p.x(p)), \mathsf{new}_1(p.y(p)))$

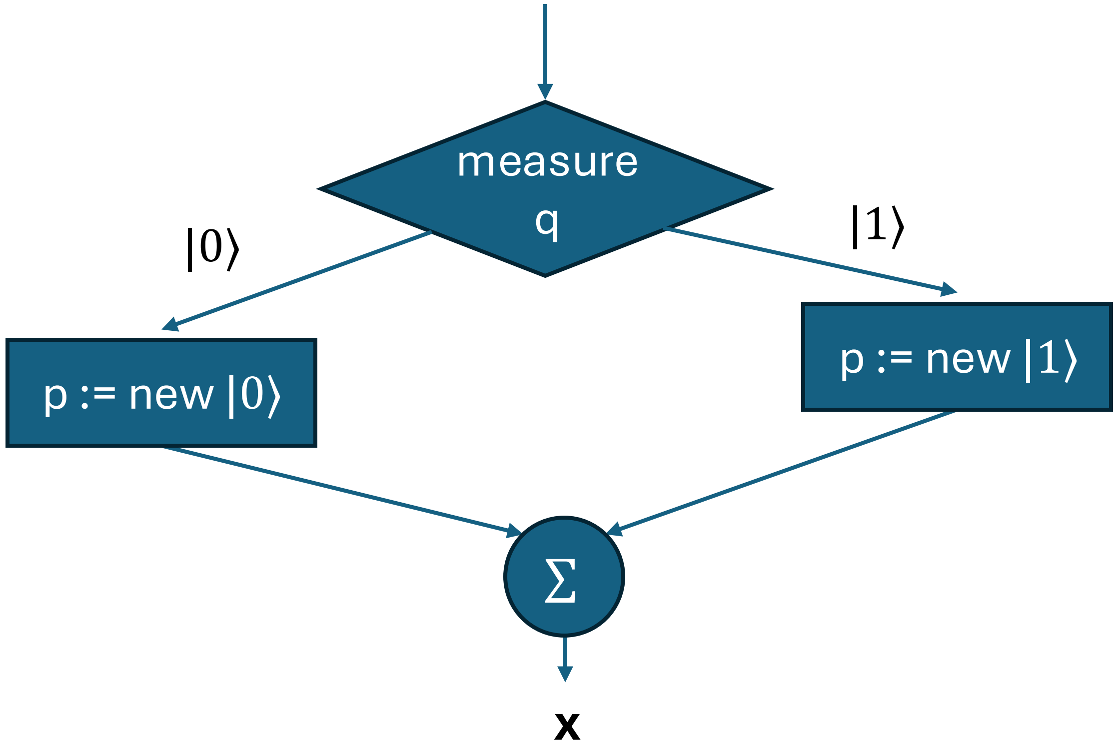


$[\![c]\!] : \mathbf{qbit} \to \mathbf{qbit} \oplus \mathbf{qbit}$

$\rho \mapsto (|0\rangle\langle 0|\rho|0\rangle\langle 0|, |1\rangle\langle 1|\rho|1\rangle\langle 1|)$

We keep the classical information "in the type"

$x : 1 \mid q \vdash \mathsf{measure}(q, \mathsf{new}_0(p.x(p)), \mathsf{new}_1(p.x(p)))$

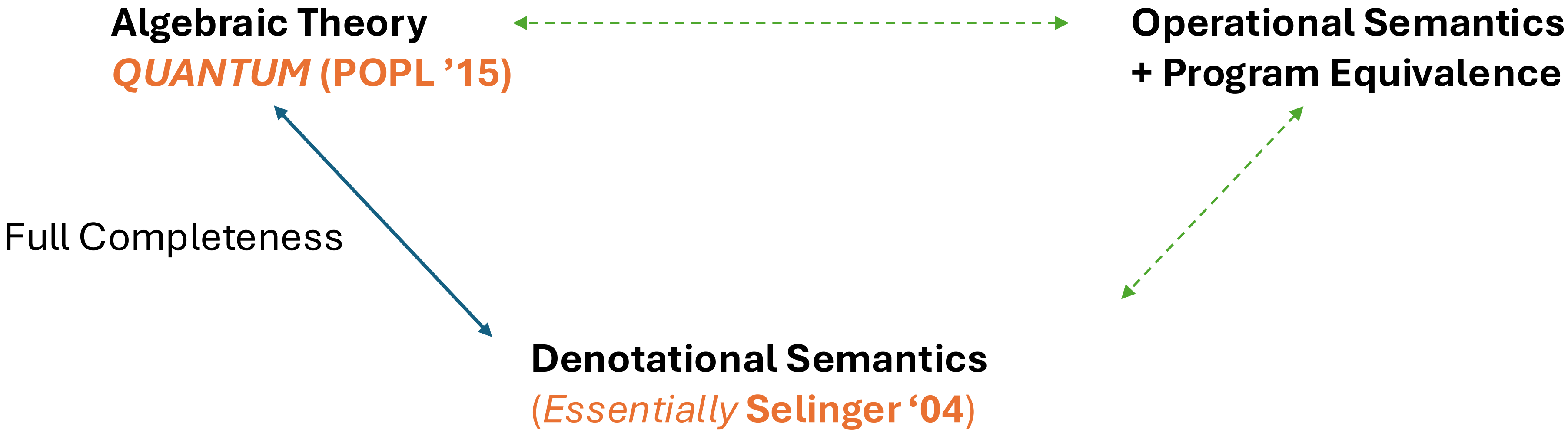


$[\![c']\!] : \mathbf{qbit} \to \mathbf{qbit}$

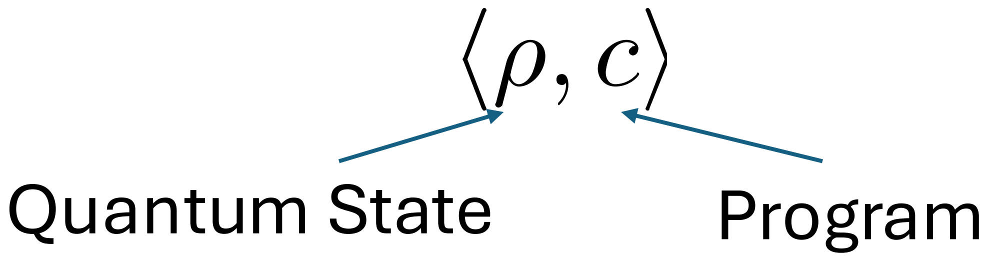
$\rho \mapsto |0\rangle\langle 0|\rho|0\rangle\langle 0| + |1\rangle\langle 1|\rho|1\rangle\langle 1|$

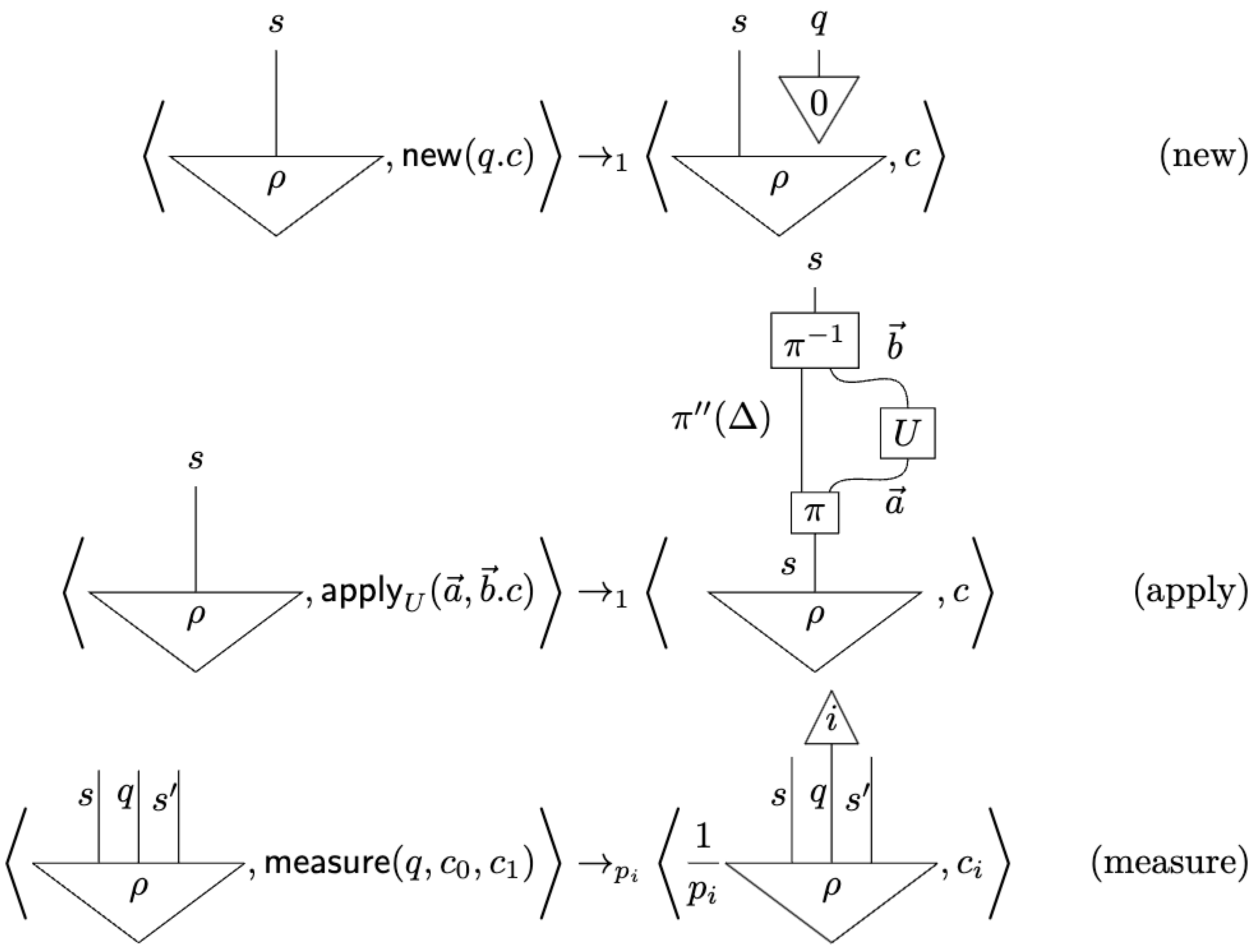
We discard any classical information not in the type

So far…


Algebraic Theory
***QUANTUM* (POPL '15)**
Operational Semantics
+ Program Equivalence
Full Completeness
Denotational Semantics
(*Essentially* **Selinger '04**)

Operational Semantics

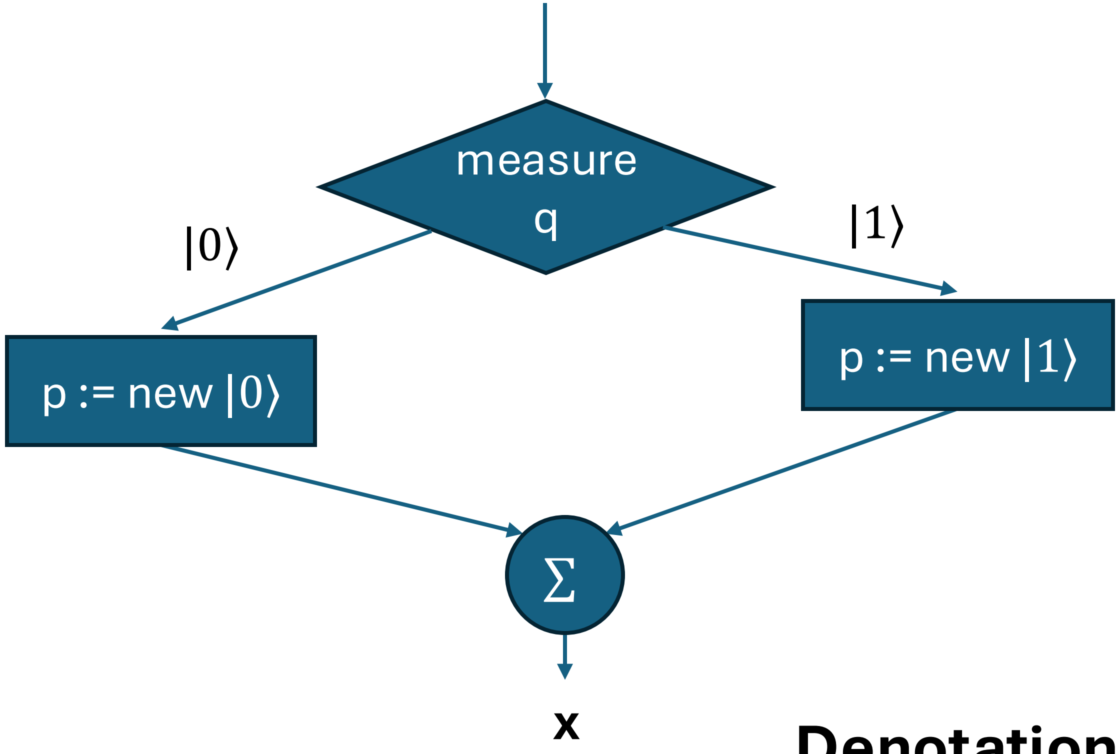


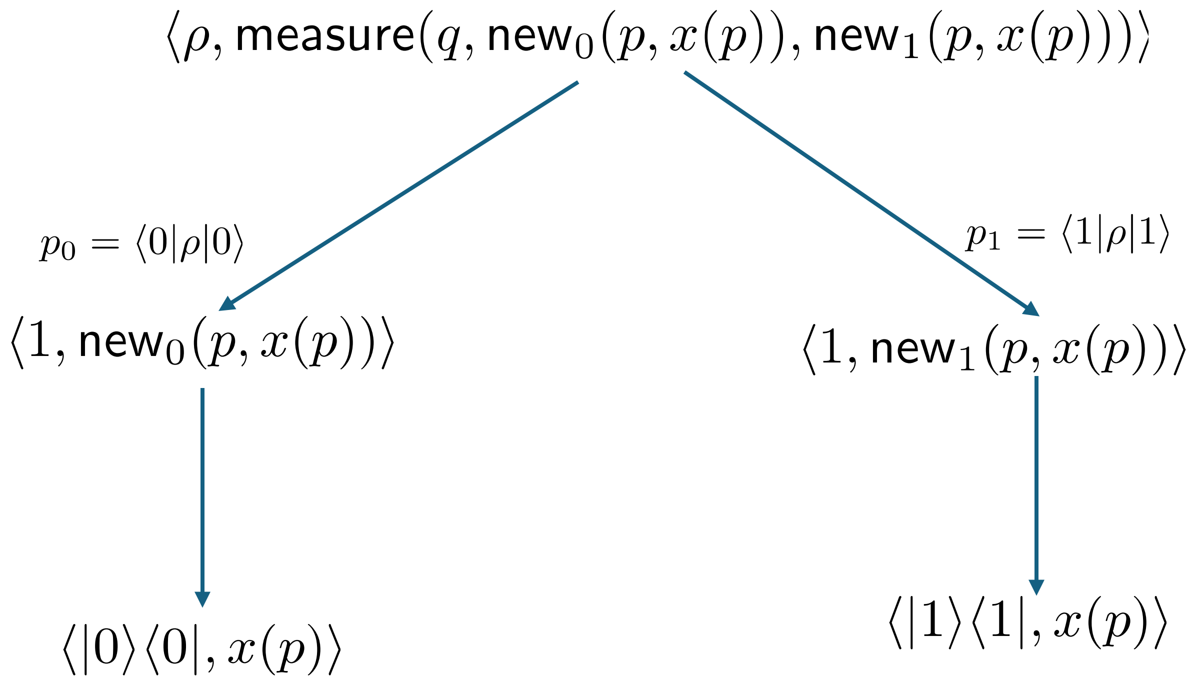


Defining a Program Equivalence

Strategy: Construct Denotation from the operational semantics

$x : 1 \mid q \vdash \mathsf{measure}(q, \mathsf{new}_0(p.x(p)), \mathsf{new}_1(p.x(p)))$





Denotation is:

$$\rho \mapsto p_0|0\rangle\langle 0| + p_1|1\rangle\langle 1|$$

Theorem:

$$[\![c]\!]_x(\rho) = \sum_{\psi} \Pr(\langle \rho, c\rangle \to^* \langle \psi, x(\Delta_\psi)\rangle) \cdot \psi$$

Corollary:

Two programs $\Gamma|\Delta \vdash c, c'$ are denotationally equal if and only if for all $x \in \Gamma$

$$\sum_{\psi} \Pr(\langle \rho, c\rangle \to^* \langle \psi, x(\Delta_\psi)\rangle) \cdot \psi = \sum_{\psi} \Pr(\langle \rho, c'\rangle \to^* \langle \psi, x(\Delta_\psi)\rangle) \cdot \psi$$

Recap

Algebraic Theory

QUANTUM (POPL '15)

Operational Semantics
+ Program Equivalence
Essentially the same as
Denotational Semantics

Denotational Semantics
In Category of CP maps

Quantum Computing + Quantum Communication

Goal

Algebraic Theory

QUANTUM extended with I/O operations ?

Operational Semantics
+ Program Equivalence
Extended with ?

Denotational Semantics
Construction on top of the **category of CP maps**?

I/O ⊗ QUANTUM: An Algebraic Theory for Quantum Communication

Operations $\mathbf{op} : (p \mid m_1 ... m_k)$

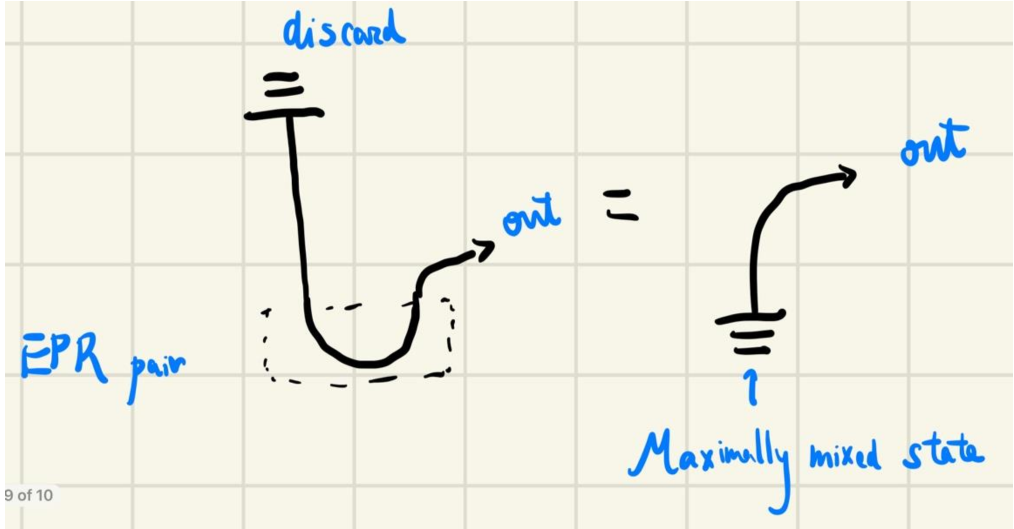
new : $(0 \mid 1)$ measure : $(1 \mid 0, 0)$ out : $(1 \mid 0)$

apply$_U$: $(n \mid n)$ in : $(0 \mid 1)$

Equations $\Gamma \mid \Delta \vdash c = c'$

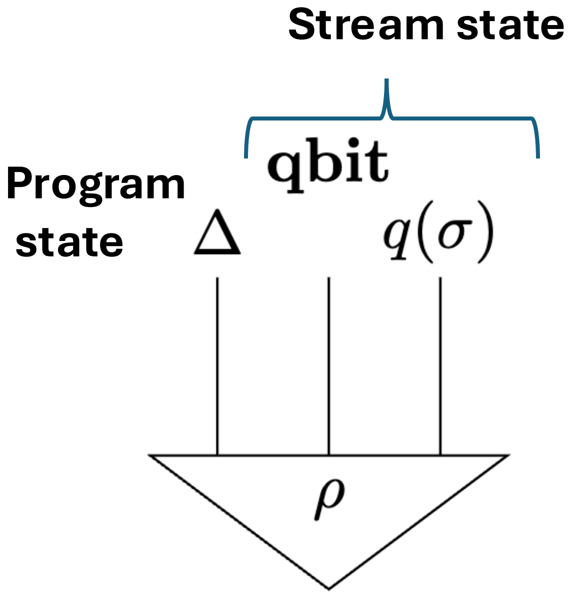
1- Input/Output do not commute with each other

2- Input/Output commute with "local" operations



$$\text{measure}(a, \text{out}(b, x), \text{out}(b, y)) = \text{out}(b, \text{measure}(a, x, y))$$

Operational Semantics

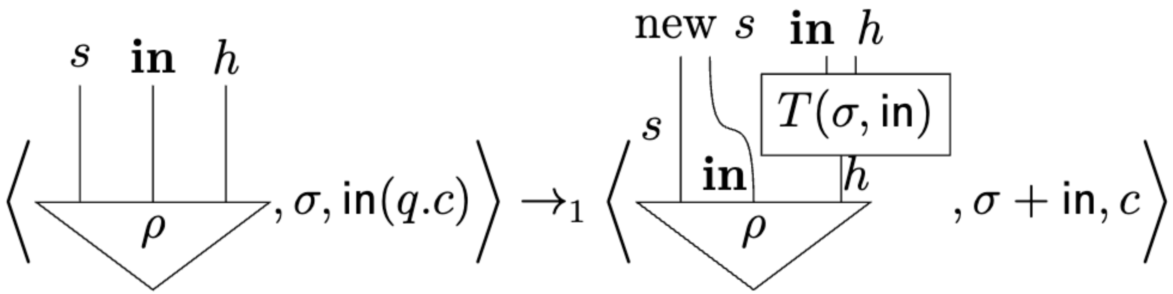


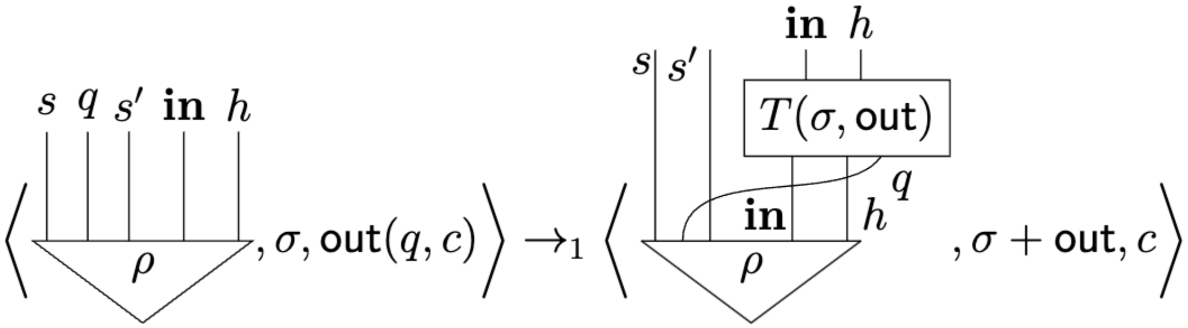
A "**quantum stream**" to handle inputs and outputs

State $\sigma \in \{\text{in}, \text{out}\}^*$
Stream $(q, T) \in \text{Stream } X_0$ where

- $q(\sigma) \in \text{Ob(CP)}$ is **the shape of the hidden state**
- $T(\sigma, \text{in}) \colon \Delta \to \text{qbit} \otimes \Delta$
  and $T(\sigma, \text{out}) \colon \text{qbit} \otimes \Delta \otimes \text{qbit} \to \text{qbit} \otimes \Delta$
  are **state evolution maps**

Configuration $\langle \rho, \sigma, c \rangle$ and Operational Semantics





Program Equivalence $\Gamma \mid \Delta \vdash c \simeq_{qu} c'$

Without communication: merge all branches with the same continuation

$$\underbrace{\forall \rho.}_{\text{Initial State}}\ \underbrace{\forall x.}_{\text{Classical Information}} \sum_{\psi} \Pr(\langle \rho, c\rangle \to^{*} \langle \psi, x(\Delta_{\psi})\rangle) \cdot \psi = \sum_{\psi} \Pr(\langle \rho, c'\rangle \to^{*} \langle \psi, x(\Delta_{\psi})\rangle) \cdot \psi$$

With communication: merge all branches with the same continuation *and the same I/O trace*

$$\underbrace{\forall \rho, (q, T), \sigma.}_{\text{Stream and Initial State}}\ \underbrace{\forall x, \sigma'.}_{\text{Classical Information}} \sum_{\psi} \Pr(\langle \rho, \sigma, c\rangle \to^{*} \langle \psi, \sigma + \sigma', x(\Delta_{\psi})\rangle) \cdot \psi$$

$$= \sum_{\psi} \Pr(\langle \rho, \sigma, c'\rangle \to^{*} \langle \psi, \sigma + \sigma', x(\Delta_{\psi})\rangle) \cdot \psi$$

Theorem: Operational semantics + Program Equivalence form a model of I/O ⊗ QUANTUM

Algebraic Theory

I/O ⊗ QUANTUM

Soundness →

Operational Semantics
+ Program Equivalence
With a **Quantum Stream**

$\Gamma \mid \Delta \vdash c = c'$ *implies* $\Gamma \mid \Delta \vdash c \simeq_{qu} c'$

Next Step

Algebraic Theory

I/O $\otimes$ QUANTUM

Soundness →

Operational Semantics
+ Program Equivalence
With a **Quantum Stream**

Denotational Semantics
Construction on top of the **category of CP maps**?

A Monadic Denotational Semantics

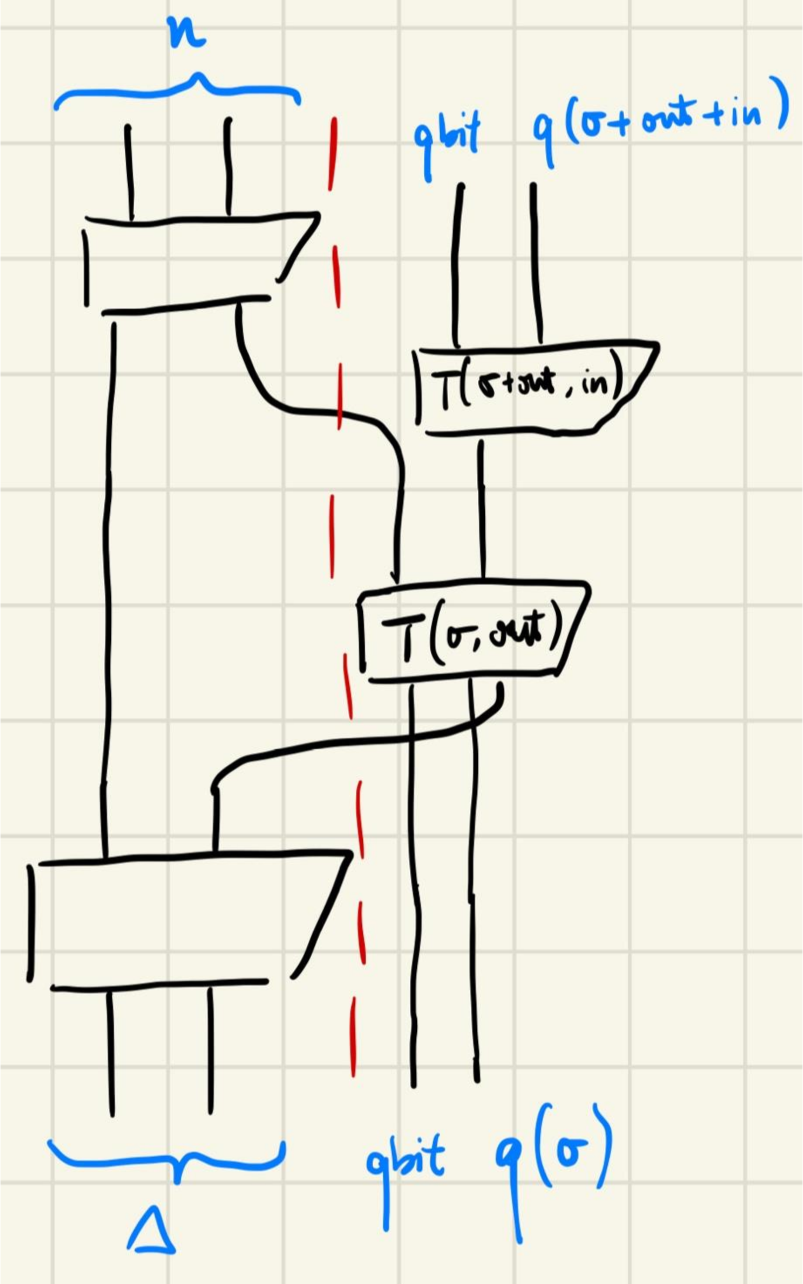
Our program equivalence

$$\forall \rho, (q, T), \sigma.\ \forall x, \sigma'.\ \sum_{\psi} \Pr(\langle \rho, \sigma, c \rangle \to^* \langle \psi, \sigma + \sigma', x(\Delta_\psi) \rangle) \cdot \psi$$

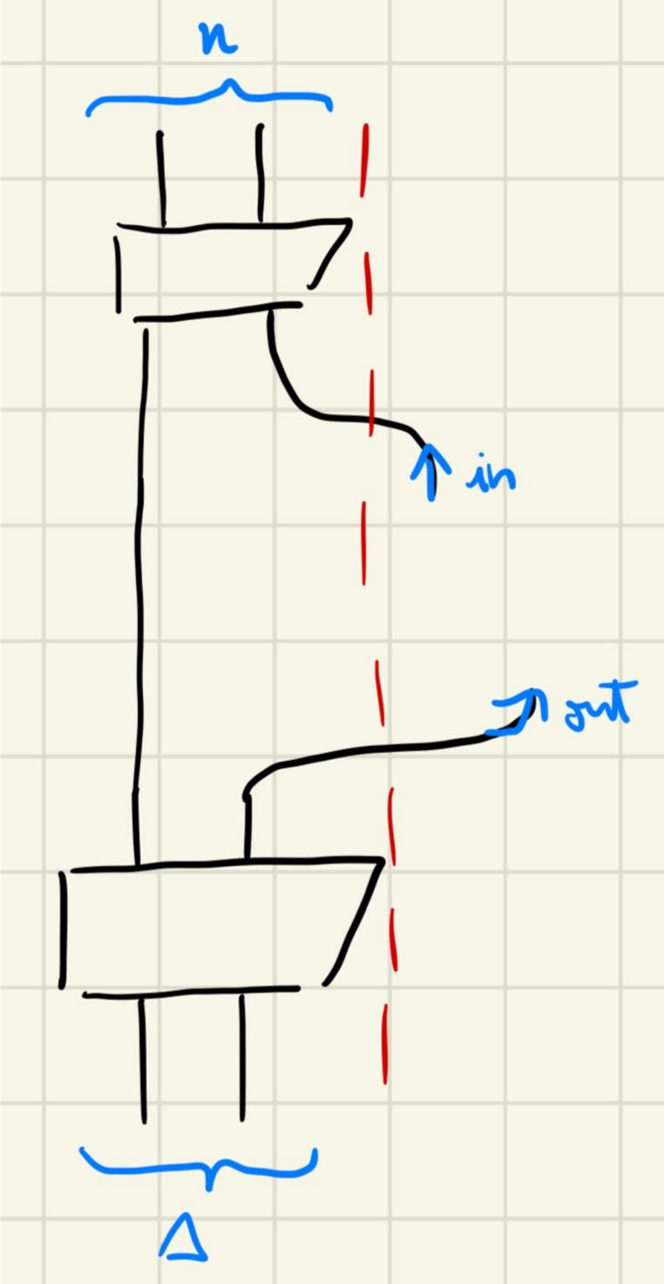
$$= \sum_{\psi} \Pr(\langle \rho, \sigma, c' \rangle \to^* \langle \psi, \sigma + \sigma', x(\Delta_\psi) \rangle) \cdot \psi$$

Consider

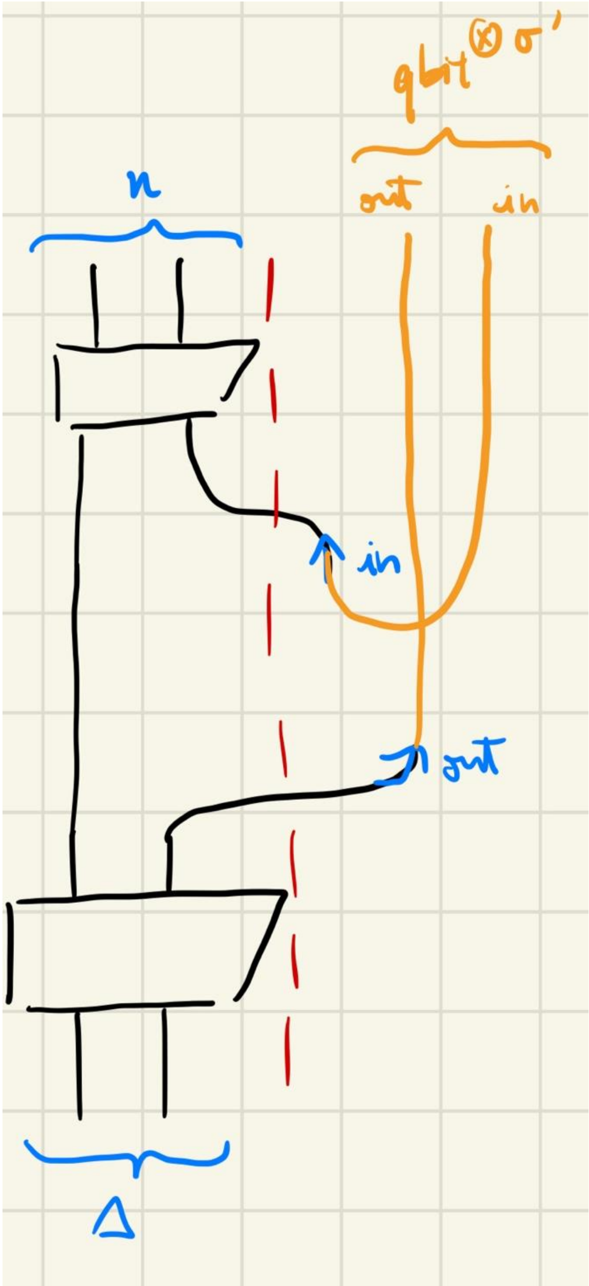
$$[\![c]\!]^{(q,T),\sigma}_{\sigma',(x:n)}(\rho) = \sum_{\psi} \Pr(\langle \rho, \sigma, c \rangle \to^* \langle \psi, \sigma + \sigma', x(\Delta_\psi) \rangle) \cdot \psi$$



$$[\![c]\!]^{(q,T),\sigma}_{\sigma',(x:n)}(\rho) = \sum_{\psi} \Pr(\langle \rho, \sigma, c\rangle \to^{*} \langle \psi, \sigma + \sigma', x(\Delta_{\psi})\rangle) \cdot \psi$$



$$[\![c]\!]_{\sigma',(x:n)} \ :?$$



$$[\![c]\!]_{\sigma',(x:n)} : \mathbf{qbit}^{\otimes \Delta} \to \mathbf{qbit}^{\otimes n} \otimes \mathbf{qbit}^{\otimes \sigma'}$$

A Monadic Denotational Semantics

$$[\![c]\!] : \mathbf{qbit}^{\otimes \Delta} \rightarrow \bigoplus_{\sigma,(x:n)\in\Gamma} \mathbf{qbit}^{\otimes \sigma} \otimes \mathbf{qbit}^{\otimes n}$$

Equivalently: $[\![c]\!] : [\![\Delta]\!] \rightarrow T([\![\Gamma]\!])$

where

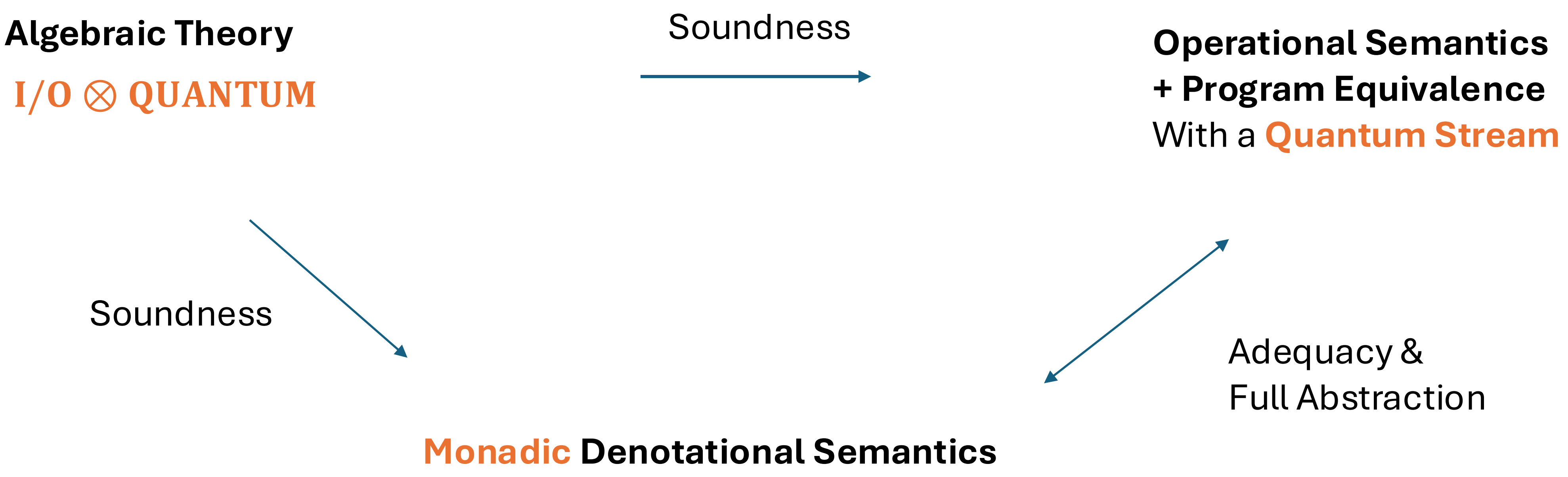
$$T : \mathbf{CP}^{\infty} \rightarrow \mathbf{CP}^{\infty}$$

$$X \mapsto \bigoplus_{\sigma} \mathbf{qbit}^{\otimes \sigma} \otimes X$$

Theorem (Adequacy and Full Abstraction):

$$\Gamma \mid \Delta \vdash c \simeq_{qu} c' \iff [\![c]\!] = [\![c']\!]$$

TADAAAAA



Algebraic Theory
I/O ⊗ QUANTUM
Soundness
Operational Semantics
+ Program Equivalence
With a Quantum Stream
Soundness
Adequacy &
Full Abstraction
Monadic Denotational Semantics

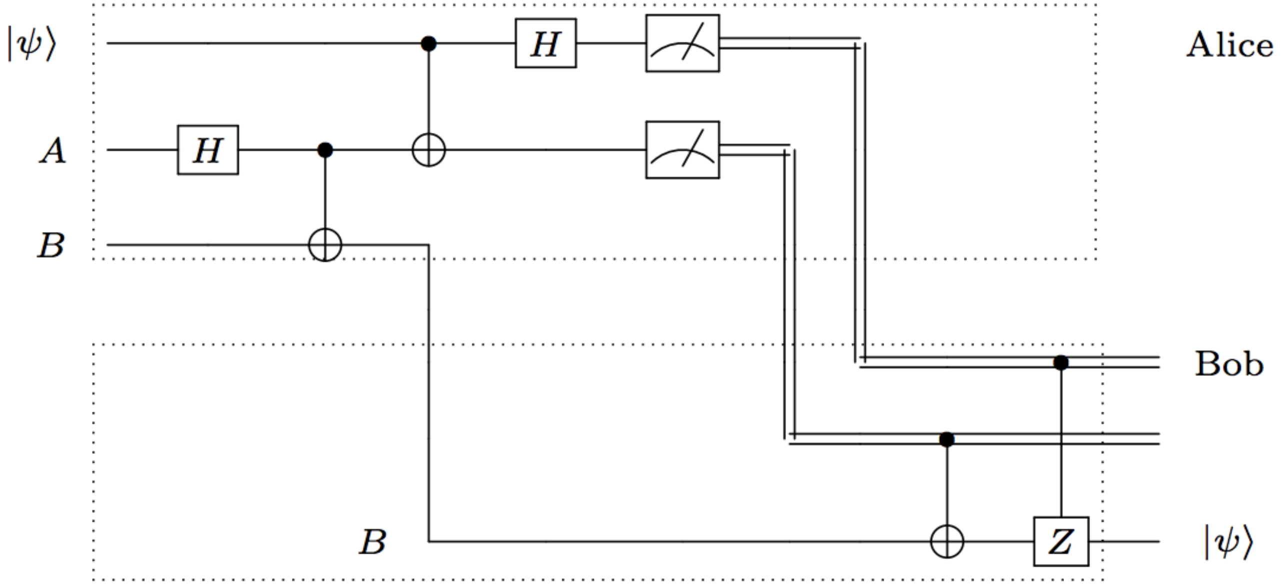
Future Work

- Complete model?
- Connections to quantum process calculi?
- Connections to Higher Order Quantum Computing?

Preprint link: https://theo.wang/works/2024-09-MSc-thesis

Garbage Slides

Why?



https://att-innovate.github.io/squanch/demos/quantum-teleportation.html

```
A: q = generate_epr(); out(q); [...]

B: q = in(); ...
```