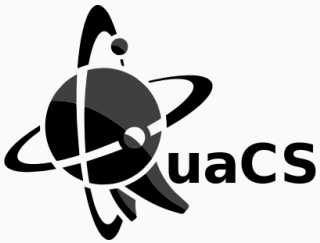
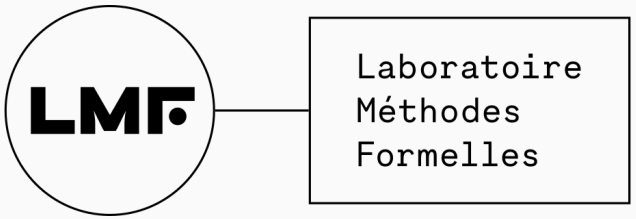
The Syntax and Semantics of Simply-typed Quantum Control

Louis Lemonnier

Acknowledgements to: Kostia Chardonnet, Robin Kaarsgaard, Benoît Valiron, Vladimir Zamdzhiev





Équipe MOCQUA. 30th May 2024

What is Quantum Control?

- Flow of the program controlled by pure quantum data.
- No use of measurement to control the flow.

```
input:  |xy>

if x:
        |xy>
else:
        |x(1−y)>
```

```
input:  |xy>

if meas(x):
        |y>
else:
        |(1−y)>
```

What is Quantum Control?

- Flow of the program controlled by pure quantum data.
- No use of measurement to control the flow.

```
input: |xy>

if x:
        |xy>
else:
        |x(Uy)>
```

```
input: |xy>

if meas(x):
        |y>
else:
        |Uy>
```

What is Quantum Control?

- Flow of the program controlled by pure quantum data.
- No use of measurement to control the flow.

```
input: |xy>                    input: |xy>

if x:                          if meas(x):
        |xy>                           |y>
else:                          else:
        |x(Uy)>                        |Uy>
```

(My) Quantum Control: work with reversible quantum operations.

They are more known as unitaries.

Is this approach new?

The syntax

Types:

$$\mathtt{I} \qquad A \oplus B \qquad A \otimes B \qquad \mathtt{Nat}$$

Terms and semantics as isometries:

The syntax

Types:

$$\boxed{\mathtt{I} \qquad A \oplus B} \qquad A \otimes B \qquad \mathtt{Nat}$$

Terms and semantics as isometries:

Syntax	$\mathbb{C}^n$	
$*\colon \mathtt{I}$	$\begin{pmatrix} 1 \end{pmatrix}$	1
$\mathtt{left}\ *\colon \mathtt{I} \oplus \mathtt{I}$	$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$	$\lvert 0 \rangle$
$\mathtt{right}\ *\colon \mathtt{I} \oplus \mathtt{I}$	$\begin{pmatrix} 0 \\ 1 \end{pmatrix}$	$\lvert 1 \rangle$
$(\alpha \cdot \mathtt{left}\ *) + (\beta \cdot \mathtt{right}\ *)\colon \mathtt{I} \oplus \mathtt{I}$	$\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$	$\alpha \lvert 0 \rangle + \beta \lvert 1 \rangle$

The syntax

Types:

$$\mathtt{I} \qquad A \oplus B \qquad \boxed{A \otimes B} \qquad \mathtt{Nat}$$

Terms and semantics as isometries:

	Syntax	$\mathbb{C}^n$	
t_i	$t_i \colon \mathtt{I} \oplus \mathtt{I}$	$\begin{pmatrix} \alpha_i \\ \beta_i \end{pmatrix}$	$\lvert\phi_i\rangle$
$t_1 \otimes t_2$	$t_1 \otimes t_2 \colon (\mathtt{I} \oplus \mathtt{I}) \otimes (\mathtt{I} \oplus \mathtt{I})$	$\begin{pmatrix} \alpha_1\alpha_2 \\ \beta_1\alpha_2 \\ \alpha_1\beta_2 \\ \beta_1\beta_2 \end{pmatrix}$	$\lvert\phi_1\rangle \otimes \lvert\phi_2\rangle$

The syntax

Types:

$$\mathtt{I} \qquad A \oplus B \qquad A \otimes B \qquad \boxed{\mathtt{Nat}}$$

Terms and semantics as isometries:

Syntax	$\ell^2(\mathbb{N})$	
$\mathtt{zero} : \mathtt{Nat}$	e_0	$\lvert\underline{0}\rangle$
$t : \mathtt{Nat}$	e_n	$\lvert\underline{n}\rangle$
$\mathtt{S}\ t : \mathtt{Nat}$	e_{n+1}	$\lvert\underline{n+1}\rangle$

The syntax

Types:

$$\mathtt{I} \qquad A \oplus B \qquad A \otimes B \qquad \mathtt{Nat} \qquad \boxed{\mu X.A}$$

Terms and semantics as isometries:

Syntax	$\bigoplus_{n=0}^{\infty} H^{\otimes n}$
$t\colon A[\mu X.A/X]$	$\star$
$\mathtt{fold}\ t\colon \mu X.A$	$\star$

Unitaries

Can be seen as λ-abstractions with pattern-matching.

λ-abstraction: and

Unitaries

Can be seen as λ-abstractions with pattern-matching.

λ-abstraction: $\lambda x.t \cong x \mapsto t$ and

Unitaries

Can be seen as λ-abstractions with pattern-matching.

λ-abstraction: $\lambda x.t \cong x \mapsto t$ and $(\lambda x.t)t' \to t[t'/x]$

Unitaries

Can be seen as λ-abstractions with pattern-matching.

λ-abstraction: $\lambda x.t \cong x \mapsto t$ and $(\lambda x.t)t' \to t[t'/x]$

$$\left\{ x \mapsto x \right\} \qquad \left\{ x \otimes y \mapsto y \otimes x \right\}$$

Unitaries

Can be seen as λ-abstractions with pattern-matching.

λ-abstraction: $\lambda x.t \cong x \mapsto t$ and $(\lambda x.t)t' \to t[t'/x]$

$$\left\{ x \mapsto x \right\} \qquad \left\{ x \otimes y \mapsto y \otimes x \right\}$$

$$\left\{ \begin{array}{ll} \texttt{left}\ * & \mapsto \texttt{right}\ * \\ \texttt{right}\ * & \mapsto \texttt{left}\ * \end{array} \right\}$$

Unitaries

Can be seen as λ-abstractions with pattern-matching.

λ-abstraction: $\lambda x.t \cong x \mapsto t$ and $(\lambda x.t)t' \to t[t'/x]$

$$\left\{ x \mapsto x \right\} \qquad \left\{ x \otimes y \mapsto y \otimes x \right\}$$

$$\left\{ \begin{array}{ll} \texttt{left}\ * & \mapsto \texttt{right}\ * \\ \texttt{right}\ * & \mapsto \texttt{left}\ * \end{array} \right\}$$

$$\left\{ \begin{array}{ll} \texttt{left}\ * & \mapsto \frac{1}{\sqrt{2}}\texttt{left}\ * + \frac{1}{\sqrt{2}}\texttt{right}\ * \\ \texttt{right}\ * & \mapsto \frac{1}{\sqrt{2}}\texttt{left}\ * - \frac{1}{\sqrt{2}}\texttt{right}\ * \end{array} \right\}$$

Unitaries

Can be seen as λ-abstractions with pattern-matching.

λ-abstraction: $\lambda x.t \cong x \mapsto t$ and $(\lambda x.t)t' \to t[t'/x]$

$$\left\{ x \mapsto x \right\} \qquad \left\{ x \otimes y \mapsto y \otimes x \right\}$$

$$\left\{ \begin{array}{ll} \texttt{left}\ * & \mapsto \texttt{right}\ * \\ \texttt{right}\ * & \mapsto \texttt{left}\ * \end{array} \right\}$$

$$\left\{ \begin{array}{ll} \texttt{left}\ * & \mapsto \frac{1}{\sqrt{2}}\texttt{left}\ * + \frac{1}{\sqrt{2}}\texttt{right}\ * \\ \texttt{right}\ * & \mapsto \frac{1}{\sqrt{2}}\texttt{left}\ * - \frac{1}{\sqrt{2}}\texttt{right}\ * \end{array} \right\}$$

$$\left\{ \begin{array}{lll} (\texttt{left}\ *) \otimes x & \mapsto & (\texttt{left}\ *) \otimes x \\ (\texttt{right}\ *) \otimes (\texttt{left}\ *) & \mapsto & (\texttt{right}\ *) \otimes (\texttt{right}\ *) \\ (\texttt{right}\ *) \otimes (\texttt{right}\ *) & \mapsto & (\texttt{right}\ *) \otimes (\texttt{left}\ *) \end{array} \right\}$$

Semantics of unitaries

Operationally: $\mathit{CNOT} = \left\{ \begin{array}{lcl} (\texttt{left}\ *) \otimes x & \mapsto & (\texttt{left}\ *) \otimes x \\ (\texttt{right}\ *) \otimes (\texttt{left}\ *) & \mapsto & (\texttt{right}\ *) \otimes (\texttt{right}\ *) \\ (\texttt{right}\ *) \otimes (\texttt{right}\ *) & \mapsto & (\texttt{right}\ *) \otimes (\texttt{left}\ *) \end{array} \right\}$

$$\mathit{CNOT}((\tfrac{1}{\sqrt{2}}\texttt{left}\ * + \tfrac{1}{\sqrt{2}}\texttt{right}\ *) \otimes (\texttt{left}\ *))$$

Semantics of unitaries

Operationally: $\mathit{CNOT} = \left\{ \begin{array}{lcl} (\texttt{left}\ *) \otimes x & \mapsto & (\texttt{left}\ *) \otimes x \\ (\texttt{right}\ *) \otimes (\texttt{left}\ *) & \mapsto & (\texttt{right}\ *) \otimes (\texttt{right}\ *) \\ (\texttt{right}\ *) \otimes (\texttt{right}\ *) & \mapsto & (\texttt{right}\ *) \otimes (\texttt{left}\ *) \end{array} \right\}$

$$\begin{array}{rl} & \mathit{CNOT}((\frac{1}{\sqrt{2}}\texttt{left}\ * + \frac{1}{\sqrt{2}}\texttt{right}\ *) \otimes (\texttt{left}\ *)) \\ \rightarrow & \mathit{CNOT}(\frac{1}{\sqrt{2}}(\texttt{left}\ *) \otimes (\texttt{left}\ *) + \frac{1}{\sqrt{2}}(\texttt{right}\ *) \otimes (\texttt{left}\ *)) \end{array}$$

Semantics of unitaries

Operationally: $CNOT = \left\{ \begin{array}{lcl} (\texttt{left}\ *) \otimes x & \mapsto & (\texttt{left}\ *) \otimes x \\ (\texttt{right}\ *) \otimes (\texttt{left}\ *) & \mapsto & (\texttt{right}\ *) \otimes (\texttt{right}\ *) \\ (\texttt{right}\ *) \otimes (\texttt{right}\ *) & \mapsto & (\texttt{right}\ *) \otimes (\texttt{left}\ *) \end{array} \right\}$

$$
\begin{array}{rl}
 & CNOT((\frac{1}{\sqrt{2}}\texttt{left}\ * + \frac{1}{\sqrt{2}}\texttt{right}\ *) \otimes (\texttt{left}\ *)) \\
\to & CNOT(\frac{1}{\sqrt{2}}(\texttt{left}\ *) \otimes (\texttt{left}\ *) + \frac{1}{\sqrt{2}}(\texttt{right}\ *) \otimes (\texttt{left}\ *)) \\
\to & \frac{1}{\sqrt{2}} CNOT((\texttt{left}\ *) \otimes (\texttt{left}\ *)) + \frac{1}{\sqrt{2}} CNOT((\texttt{right}\ *) \otimes (\texttt{left}\ *)))
\end{array}
$$

Semantics of unitaries

Operationally: $\mathit{CNOT} = \left\{ \begin{array}{lcl} (\texttt{left}\ *) \otimes x & \mapsto & (\texttt{left}\ *) \otimes x \\ (\texttt{right}\ *) \otimes (\texttt{left}\ *) & \mapsto & (\texttt{right}\ *) \otimes (\texttt{right}\ *) \\ (\texttt{right}\ *) \otimes (\texttt{right}\ *) & \mapsto & (\texttt{right}\ *) \otimes (\texttt{left}\ *) \end{array} \right\}$

$$
\begin{array}{rl}
 & \mathit{CNOT}((\frac{1}{\sqrt{2}}\texttt{left}\ * + \frac{1}{\sqrt{2}}\texttt{right}\ *) \otimes (\texttt{left}\ *)) \\
\to & \mathit{CNOT}(\frac{1}{\sqrt{2}}(\texttt{left}\ *) \otimes (\texttt{left}\ *) + \frac{1}{\sqrt{2}}(\texttt{right}\ *) \otimes (\texttt{left}\ *)) \\
\to & \frac{1}{\sqrt{2}}\mathit{CNOT}((\texttt{left}\ *) \otimes (\texttt{left}\ *)) + \frac{1}{\sqrt{2}}\mathit{CNOT}((\texttt{right}\ *) \otimes (\texttt{left}\ *))) \\
\to & \frac{1}{\sqrt{2}}(\texttt{left}\ *) \otimes (\texttt{left}\ *) +
\end{array}
$$

Semantics of unitaries

Operationally: $\mathit{CNOT} = \left\{ \begin{array}{lcl} (\texttt{left}\ *) \otimes x & \mapsto & (\texttt{left}\ *) \otimes x \\ (\texttt{right}\ *) \otimes (\texttt{left}\ *) & \mapsto & (\texttt{right}\ *) \otimes (\texttt{right}\ *) \\ (\texttt{right}\ *) \otimes (\texttt{right}\ *) & \mapsto & (\texttt{right}\ *) \otimes (\texttt{left}\ *) \end{array} \right\}$

$$
\begin{array}{rl}
 & \mathit{CNOT}((\frac{1}{\sqrt{2}}\texttt{left}\ * + \frac{1}{\sqrt{2}}\texttt{right}\ *) \otimes (\texttt{left}\ *)) \\
\to & \mathit{CNOT}(\frac{1}{\sqrt{2}}(\texttt{left}\ *) \otimes (\texttt{left}\ *) + \frac{1}{\sqrt{2}}(\texttt{right}\ *) \otimes (\texttt{left}\ *)) \\
\to & \frac{1}{\sqrt{2}}\mathit{CNOT}((\texttt{left}\ *) \otimes (\texttt{left}\ *)) + \frac{1}{\sqrt{2}}\mathit{CNOT}((\texttt{right}\ *) \otimes (\texttt{left}\ *))) \\
\to & \frac{1}{\sqrt{2}}(\texttt{left}\ *) \otimes (\texttt{left}\ *) + \frac{1}{\sqrt{2}}(\texttt{right}\ *) \otimes (\texttt{right}\ *)
\end{array}
$$

Semantics of unitaries

Operationally: $\mathit{CNOT} = \left\{ \begin{array}{lcl} (\texttt{left}\ *) \otimes x & \mapsto & (\texttt{left}\ *) \otimes x \\ (\texttt{right}\ *) \otimes (\texttt{left}\ *) & \mapsto & (\texttt{right}\ *) \otimes (\texttt{right}\ *) \\ (\texttt{right}\ *) \otimes (\texttt{right}\ *) & \mapsto & (\texttt{right}\ *) \otimes (\texttt{left}\ *) \end{array} \right\}$

$$
\begin{array}{ll}
 & \mathit{CNOT}((\frac{1}{\sqrt{2}}\texttt{left}\ * + \frac{1}{\sqrt{2}}\texttt{right}\ *) \otimes (\texttt{left}\ *)) \\
\to & \mathit{CNOT}(\frac{1}{\sqrt{2}}(\texttt{left}\ *) \otimes (\texttt{left}\ *) + \frac{1}{\sqrt{2}}(\texttt{right}\ *) \otimes (\texttt{left}\ *)) \\
\to & \frac{1}{\sqrt{2}}\mathit{CNOT}((\texttt{left}\ *) \otimes (\texttt{left}\ *)) + \frac{1}{\sqrt{2}}\mathit{CNOT}((\texttt{right}\ *) \otimes (\texttt{left}\ *))) \\
\to & \frac{1}{\sqrt{2}}(\texttt{left}\ *) \otimes (\texttt{left}\ *) + \frac{1}{\sqrt{2}}(\texttt{right}\ *) \otimes (\texttt{right}\ *)
\end{array}
$$

Mathematically:

$$
\left[\!\!\left[\vdash \left\{ \begin{array}{ccc} t_1 & \mapsto & t'_1 \\ & \vdots & \\ t_m & \mapsto & t'_m \end{array} \right\} : A \leftrightarrow B \right]\!\!\right] = \sum_{i=1}^{m} [\![\Delta \vdash t'_i : B]\!] \circ [\![\Delta \vdash t_i : A]\!]^\dagger
$$

How do we ensure unitarity?

A syntactic notion of orthonormal basis (ONB).

In linear algebra, a unitary transforms an ONB into an ONB.

$$\left\{ \begin{array}{ccc} t_1 & \mapsto & t'_1 \\ & \vdots & \\ t_m & \mapsto & t'_m \end{array} \right\}$$

This is a unitary if both $\{t_1\}_i$ and $\{t'_j\}_j$ are ONBs.

How do we ensure unitarity?

A syntactic notion of orthonormal basis (ONB).

In linear algebra, a unitary transforms an ONB into an ONB.

$$\left\{ \begin{array}{ccc} t_1 & \mapsto & t'_1 \\ & \vdots & \\ t_m & \mapsto & t'_m \end{array} \right\}$$

This is a unitary if both $\{t_1\}_i$ and $\{t'_j\}_j$ are ONBs.

$$ONB_A\{x\} \qquad ONB_{A\otimes B}\{x \otimes y\}$$

$$ONB_{\mathrm{I}\oplus\mathrm{I}}\{\texttt{left}\ *, \texttt{right}\ *\}$$

$$ONB_{\mathrm{I}\oplus\mathrm{I}}\{\tfrac{1}{\sqrt{2}}\texttt{left}\ x + \tfrac{1}{\sqrt{2}}\texttt{right}\ x,\quad \tfrac{1}{\sqrt{2}}\texttt{left}\ x - \tfrac{1}{\sqrt{2}}\texttt{right}\ x\}$$

How do we ensure unitarity?

A syntactic notion of orthonormal basis (ONB).

In linear algebra, a unitary transforms an ONB into an ONB.

$$\left\{ \begin{array}{ccc} t_1 & \mapsto & t'_1 \\ & \vdots & \\ t_m & \mapsto & t'_m \end{array} \right\}$$

This is a unitary if both $\{t_1\}_i$ and $\{t'_j\}_j$ are ONBs.

$$ONB_A\{x\} \qquad ONB_{A\otimes B}\{x \otimes y\}$$

$$ONB_{\mathtt{Nat}}\{\mathtt{zero}, \mathtt{S}\ \mathtt{zero}, \mathtt{S}\ \mathtt{S}\ x\}$$

$$ONB_{\mathtt{I}\oplus\mathtt{I}}\{\tfrac{1}{\sqrt{2}}\mathtt{zero} + \tfrac{1}{\sqrt{2}}\mathtt{S}\ \mathtt{zero}, \quad \tfrac{1}{\sqrt{2}}\mathtt{zero} - \tfrac{1}{\sqrt{2}}\mathtt{S}\ \mathtt{zero}, \quad \mathtt{S}\ \mathtt{S}\ x\}$$

Properties

With a suitable equational theory (cf. my thesis).

Theorem

If $\Delta \vdash t\colon A$ is well-typed and $ONB_A(\{t_i\})$ then $\Delta \vdash t = \sum_i \alpha_i t_i\colon A$.

It implies that there is a unique normal form. And thus:

Properties

With a suitable equational theory (cf. my thesis).

Theorem

If $\Delta \vdash t : A$ is well-typed and $ONB_A(\{t_i\})$ then $\Delta \vdash t = \sum_i \alpha_i t_i : A$.

It implies that there is a unique normal form. And thus:

Theorem (Completeness)

$$\cdot \vdash t_1 = t_2 : A \quad \text{iff} \quad [\![\cdot \vdash t_1 : A]\!] = [\![\cdot \vdash t_2 : A]\!].$$

With a denotational semantics in Hilbert spaces and isometries.

Properties

With a suitable equational theory (cf. my thesis).

Theorem

If $\Delta \vdash t \colon A$ is well-typed and $ONB_A(\{t_i\})$ then $\Delta \vdash t = \sum_i \alpha_i t_i \colon A$.

It implies that there is a unique normal form. And thus:

Theorem (Completeness)

$$\cdot \vdash t_1 = t_2 \colon A \quad \text{iff} \quad [\![\cdot \vdash t_1 \colon A]\!] = [\![\cdot \vdash t_2 \colon A]\!].$$

With a denotational semantics in Hilbert spaces and isometries.

Other cool result: our syntactic ONBs are realisations of the identity.

If $ONB_A(S)$, then $\sum_{t \in S} [\![t]\!] \circ [\![t]\!]^\dagger = \mathrm{id}$.

What about Higher-order?

Comparison with the λ-calculus is limited.

In a unitary $\{t_i \mapsto t'_i\}$, the components t_i and t'_i are ground terms.

Cannot take a unitary as an input and cannot output a unitary.

Need to add a new form of (higher-order) functions.

What about Higher-order?

Comparison with the λ-calculus is limited.

In a unitary $\{t_i \mapsto t'_i\}$, the components t_i and t'_i are ground terms.

Cannot take a unitary as an input and cannot output a unitary.

Need to add a new form of (higher-order) functions.

$$\left\{ \; x \;\; \mapsto \;\; \texttt{let}\, y = \omega\; x\; \texttt{in}\; y \; \right\}$$

What about Higher-order?

Comparison with the λ-calculus is limited.

In a unitary $\{t_i \mapsto t_i'\}$, the components t_i and t_i' are ground terms.

Cannot take a unitary as an input and cannot output a unitary.

Need to add a new form of (higher-order) functions.

$$\left\{ \; x \;\; \mapsto \;\; \texttt{let}\, y = \omega\; x\; \texttt{in}\; y \; \right\}$$

$$\lambda\phi. \left\{ \; x \;\; \mapsto \;\; \texttt{let}\, y = \phi\; x\; \texttt{in}\; y \; \right\}$$

What about Higher-order?

Comparison with the λ-calculus is limited.

In a unitary $\{t_i \mapsto t_i'\}$, the components t_i and t_i' are ground terms.

Cannot take a unitary as an input and cannot output a unitary.

Need to add a new form of (higher-order) functions.

$$\left\{ \; x \;\; \mapsto \;\; \texttt{let}\, y = \omega\; x\; \texttt{in}\; y \; \right\}$$

$$\lambda\phi. \left\{ \; x \;\; \mapsto \;\; \texttt{let}\, y = \phi\; x\; \texttt{in}\; y \; \right\}$$

We have a λ-calculus at the level of unitaries.

What does Higher-order achieve?

A general control operation.

$$\lambda\phi.\left\{\begin{array}{lcl} (\texttt{left}\ *) \otimes x & \mapsto & (\texttt{left}\ *) \otimes x \\ (\texttt{right}\ *) \otimes x & \mapsto & \texttt{let}\, y = \phi\ x\ \texttt{in} \\ & & (\texttt{right}\ *) \otimes y \end{array}\right\}$$

What does Higher-order achieve?

A general control operation.

$$\lambda\phi.\left\{\begin{array}{lcl} (\texttt{left}\ *)\otimes x & \mapsto & (\texttt{left}\ *)\otimes x \\ (\texttt{right}\ *)\otimes x & \mapsto & \texttt{let}\, y = \phi\ x\ \texttt{in} \\ & & (\texttt{right}\ *)\otimes y \end{array}\right\}$$

The quantum switch.

$$\lambda\phi.\lambda\psi.\left\{\begin{array}{lcl} (\texttt{left}\ *)\otimes x & \mapsto & \texttt{let}\, y = \phi\ x\ \texttt{in} \\ & & \texttt{let}\, z = \psi\ y\ \texttt{in} \\ & & (\texttt{left}\ *)\otimes z \\ (\texttt{right}\ *)\otimes x & \mapsto & \texttt{let}\, y = \psi\ x\ \texttt{in} \\ & & \texttt{let}\, z = \phi\ y\ \texttt{in} \\ & & (\texttt{right}\ *)\otimes z \end{array}\right\}$$

What does Higher-order achieve?

A general control operation.

$$\lambda\phi.\left\{\begin{array}{lcl}(\texttt{left}\ *)\otimes x & \mapsto & (\texttt{left}\ *)\otimes x \\ (\texttt{right}\ *)\otimes x & \mapsto & \texttt{let}\, y = \phi\ x\ \texttt{in} \\ & & (\texttt{right}\ *)\otimes y\end{array}\right\}$$

The quantum switch.

$$\lambda\phi.\lambda\psi.\left\{\begin{array}{lcl}(\texttt{left}\ *)\otimes x & \mapsto & \texttt{let}\, y = \phi\ x\ \texttt{in} \\ & & \texttt{let}\, z = \psi\ y\ \texttt{in} \\ & & (\texttt{left}\ *)\otimes z \\ (\texttt{right}\ *)\otimes x & \mapsto & \texttt{let}\, y = \psi\ x\ \texttt{in} \\ & & \texttt{let}\, z = \phi\ y\ \texttt{in} \\ & & (\texttt{right}\ *)\otimes z\end{array}\right\}$$

Conjecture

Completeness still holds with higher-order.

What is next?

We have:

- A programming language,
- with *inductive types*,
- with higher-order.

What is next?

We have:

- A programming language,
- with *inductive types*,
- with higher-order.

What is next?

What is next?

We have:

- A programming language,
- with *inductive types*,
- with higher-order.

What is next?

Recursion!

With a fixed point combinator.

$$\texttt{fix}\ \phi.\,\{\ldots\}$$

Quick stop: Recursion in Classical Reversibility

Suppose we have a syntax with:

- Only reversible functions,
- Inductive types,
- A fixpoint operator,
- A CbV operational semantics.

It is Turing-complete! (proven by Kostia Chardonnet)

Sound and adequate denotational semantics in partial injective functions.

Given two sets A and B, the set $[A \rightharpoonup B]$ has nice properties for fixpoints.

Read about it: [CLV23].

Quick stop: Recursion in Classical Reversibility

Suppose we have a syntax with:

- Only reversible functions,
- Inductive types,
- A fixpoint operator,
- A CbV operational semantics.

It is Turing-complete! (proven by Kostia Chardonnet)

Sound and adequate denotational semantics in partial injective functions.

Given two sets A and B, the set $[A \rightharpoonup B]$ has nice properties for fixpoints.

(The category of partial injective functions between sets is enriched in pointed dcpos.)

Read about it: [CLV23].

Until we meet again

Reasons to believe that *general* quantum recursion does not hold.

In Hilbert spaces: no enrichment (some details?), no trace (conjecture).

Many questions remain. Is quantum recursion not possible?

Until we meet again

Reasons to believe that *general* quantum recursion does not hold.

In Hilbert spaces: no enrichment (some details?), no trace (conjecture).

Many questions remain. Is quantum recursion not possible?

If you ask people in the community:

- Classical recursion allows non termination.
- *"Non-terminating quantum-controlled program do not make sense."*
- But no mathematical account of this point yet.

Until we meet again

Reasons to believe that *general* quantum recursion does not hold.

In Hilbert spaces: no enrichment (some details?), no trace (conjecture).

Many questions remain. Is quantum recursion not possible?

If you ask people in the community:

- Classical recursion allows non termination.
- *"Non-terminating quantum-controlled program do not make sense."*
- But no mathematical account of this point yet.

Hilbert spaces might not be the right model.

Until we meet again

Reasons to believe that *general* quantum recursion does not hold.

In Hilbert spaces: no enrichment (some details?), no trace (conjecture).

Many questions remain. Is quantum recursion not possible?

If you ask people in the community:

- Classical recursion allows non termination.
- *"Non-terminating quantum-controlled program do not make sense."*
- But no mathematical account of this point yet.

Hilbert spaces might not be the right model.

Thank you!

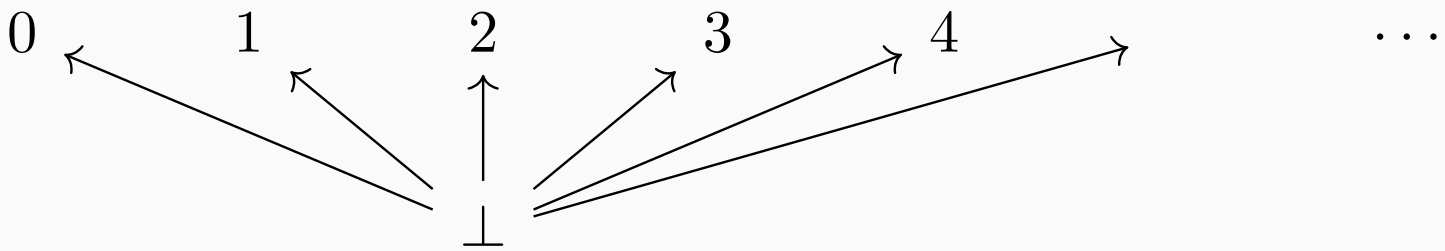
Come to my defence on 19th June, 2pm.

Domain theory (for babies)

Domain theory (for babies)

$\mathbb{N}_\bot \stackrel{\text{def}}{=} \mathbb{N} \cup \{\bot\}$ is a partial-ordered set with nice properties.

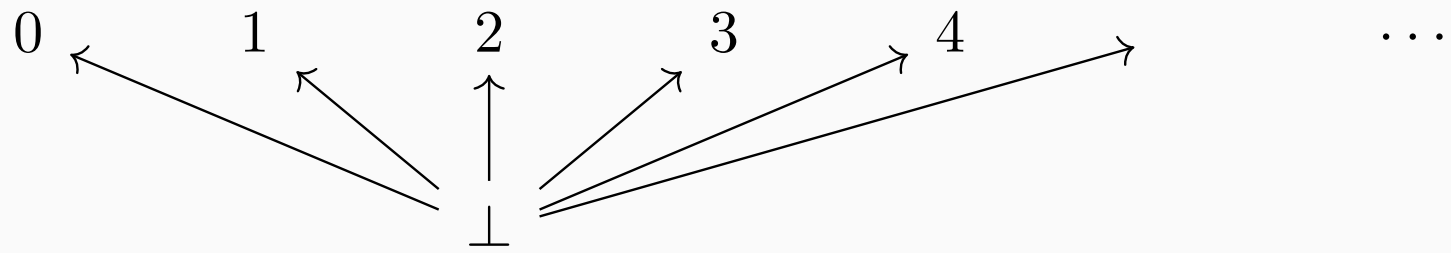
$\bot$ means undefined.



Domain theory (for babies)

$\mathbb{N}_\bot \stackrel{\text{def}}{=} \mathbb{N} \cup \{\bot\}$ is a partial-ordered set with nice properties.

$\bot$ means undefined.

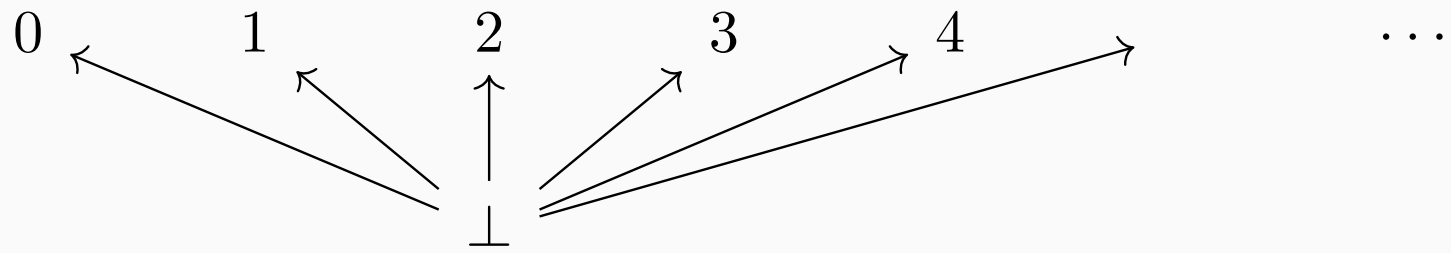


The set $[D \to D']$ of monotone functions also has nice properties.

Domain theory (for babies)

$\mathbb{N}_\bot \stackrel{\text{def}}{=} \mathbb{N} \cup \{\bot\}$ is a partial-ordered set with nice properties.

$\bot$ means undefined.



The set $[D \to D']$ of monotone functions also has nice properties.

How do we use it?

Interpretation of recursion

Define $F : [[\mathbb{N}_\perp \to \mathbb{N}_\perp] \to [\mathbb{N}_\perp \to \mathbb{N}_\perp]]$ as:

$$F(g)(n) = \begin{cases} 1 & \text{if } n = 0, \\ n \times g(n-1) & \text{else.} \end{cases}$$

Write $\perp : \mathbb{N}_\perp \to \mathbb{N}_\perp$ the constant function with output $\perp$.
$\perp : \mathbb{N}_\perp \to \mathbb{N}_\perp$ is:

$$\perp \quad \perp \quad \perp \quad \perp \quad \perp \quad \perp \quad \perp \quad \cdots$$

Interpretation of recursion

Define $F : [[\mathbb{N}_\perp \to \mathbb{N}_\perp] \to [\mathbb{N}_\perp \to \mathbb{N}_\perp]]$ as:

$$F(g)(n) = \begin{cases} 1 & \text{if } n = 0, \\ n \times g(n-1) & \text{else.} \end{cases}$$

Write $\perp : \mathbb{N}_\perp \to \mathbb{N}_\perp$ the constant function with output $\perp$.
$\perp : \mathbb{N}_\perp \to \mathbb{N}_\perp$ is:

$$\perp \quad \perp \quad \perp \quad \perp \quad \perp \quad \perp \quad \perp \quad \cdots$$

$F(\perp) : \mathbb{N}_\perp \to \mathbb{N}_\perp$ is:

$$1 \quad \perp \quad \perp \quad \perp \quad \perp \quad \perp \quad \perp \quad \cdots$$

Interpretation of recursion

Define $F : [[\mathbb{N}_\perp \to \mathbb{N}_\perp] \to [\mathbb{N}_\perp \to \mathbb{N}_\perp]]$ as:

$$F(g)(n) = \begin{cases} 1 & \text{if } n = 0, \\ n \times g(n-1) & \text{else.} \end{cases}$$

Write $\perp : \mathbb{N}_\perp \to \mathbb{N}_\perp$ the constant function with output $\perp$.
$F(\perp) : \mathbb{N}_\perp \to \mathbb{N}_\perp$ is:

$$1 \quad \perp \quad \perp \quad \perp \quad \perp \quad \perp \quad \perp \quad \cdots$$

$F(F(\perp)) : \mathbb{N}_\perp \to \mathbb{N}_\perp$ is:

$$1 \quad 1 \quad \perp \quad \perp \quad \perp \quad \perp \quad \perp \quad \cdots$$

Interpretation of recursion

Define $F : [[\mathbb{N}_\perp \to \mathbb{N}_\perp] \to [\mathbb{N}_\perp \to \mathbb{N}_\perp]]$ as:

$$F(g)(n) = \begin{cases} 1 & \text{if } n = 0, \\ n \times g(n-1) & \text{else.} \end{cases}$$

Write $\perp : \mathbb{N}_\perp \to \mathbb{N}_\perp$ the constant function with output $\perp$.
$F(F(\perp)) : \mathbb{N}_\perp \to \mathbb{N}_\perp$ is:

$$1 \quad 1 \quad \perp \quad \perp \quad \perp \quad \perp \quad \perp \quad \cdots$$

$F(F(F(\perp))) : \mathbb{N}_\perp \to \mathbb{N}_\perp$ is:

$$1 \quad 1 \quad 2 \quad \perp \quad \perp \quad \perp \quad \perp \quad \cdots$$

Interpretation of recursion

Define $F : [[\mathbb{N}_\perp \to \mathbb{N}_\perp] \to [\mathbb{N}_\perp \to \mathbb{N}_\perp]]$ as:

$$F(g)(n) = \begin{cases} 1 & \text{if } n = 0, \\ n \times g(n-1) & \text{else.} \end{cases}$$

Write $\perp : \mathbb{N}_\perp \to \mathbb{N}_\perp$ the constant function with output $\perp$.
$F(F(F(\perp))) : \mathbb{N}_\perp \to \mathbb{N}_\perp$ is:

$$1 \quad 1 \quad 2 \quad \perp \quad \perp \quad \perp \quad \perp \quad \cdots$$

$F(F(F(F(\perp)))) : \mathbb{N}_\perp \to \mathbb{N}_\perp$ is:

$$1 \quad 1 \quad 2 \quad 6 \quad \perp \quad \perp \quad \perp \quad \cdots$$

Interpretation of recursion

Define $F : [[\mathbb{N}_\perp \to \mathbb{N}_\perp] \to [\mathbb{N}_\perp \to \mathbb{N}_\perp]]$ as:

$$F(g)(n) = \begin{cases} 1 & \text{if } n = 0, \\ n \times g(n-1) & \text{else.} \end{cases}$$

Write $\perp : \mathbb{N}_\perp \to \mathbb{N}_\perp$ the constant function with output $\perp$.
$F(F(F(F(\perp)))) : \mathbb{N}_\perp \to \mathbb{N}_\perp$ is:

$$1 \quad 1 \quad 2 \quad 6 \quad \perp \quad \perp \quad \perp \quad \cdots$$

$F(F(F(F(F(\perp))))) : \mathbb{N}_\perp \to \mathbb{N}_\perp$ is:

$$1 \quad 1 \quad 2 \quad 6 \quad 24 \quad \perp \quad \perp \quad \cdots$$

Interpretation of recursion

Define $F : [[\mathbb{N}_\perp \to \mathbb{N}_\perp] \to [\mathbb{N}_\perp \to \mathbb{N}_\perp]]$ as:

$$F(g)(n) = \begin{cases} 1 & \text{if } n = 0, \\ n \times g(n-1) & \text{else.} \end{cases}$$

Write $\perp : \mathbb{N}_\perp \to \mathbb{N}_\perp$ the constant function with output $\perp$.
$F(F(F(F(F(\perp))))) : \mathbb{N}_\perp \to \mathbb{N}_\perp$ is:

$$1 \quad 1 \quad 2 \quad 6 \quad 24 \quad \perp \quad \perp \quad \cdots$$

$F(F(F(F(F(F(\perp)))))) : \mathbb{N}_\perp \to \mathbb{N}_\perp$ is:

$$1 \quad 1 \quad 2 \quad 6 \quad 24 \quad 120 \quad \perp \quad \cdots$$

Interpretation of recursion

Define $F : [[\mathbb{N}_\perp \to \mathbb{N}_\perp] \to [\mathbb{N}_\perp \to \mathbb{N}_\perp]]$ as:

$$F(g)(n) = \begin{cases} 1 & \text{if } n = 0, \\ n \times g(n-1) & \text{else.} \end{cases}$$

Write $\perp : \mathbb{N}_\perp \to \mathbb{N}_\perp$ the constant function with output $\perp$.
$F(F(F(F(F(\perp))))) : \mathbb{N}_\perp \to \mathbb{N}_\perp$ is:

$$1 \quad 1 \quad 2 \quad 6 \quad 24 \quad \perp \quad \perp \quad \cdots$$

$F(F(F(F(F(F(\perp)))))) : \mathbb{N}_\perp \to \mathbb{N}_\perp$ is:

$$1 \quad 1 \quad 2 \quad 6 \quad 24 \quad 120 \quad \perp \quad \cdots$$

The interpretation of the factorial is then the limit of $(F^n(\perp))_n$.

Interpretation of recursion

Define $F : [[\mathbb{N}_\perp \to \mathbb{N}_\perp] \to [\mathbb{N}_\perp \to \mathbb{N}_\perp]]$ as:

$$F(g)(n) = \begin{cases} 1 & \text{if } n = 0, \\ n \times g(n-1) & \text{else.} \end{cases}$$

Write $\perp : \mathbb{N}_\perp \to \mathbb{N}_\perp$ the constant function with output $\perp$.
$F(F(F(F(F(\perp))))) : \mathbb{N}_\perp \to \mathbb{N}_\perp$ is:

$$1 \quad 1 \quad 2 \quad 6 \quad 24 \quad \perp \quad \perp \quad \cdots$$

$F(F(F(F(F(F(\perp)))))) : \mathbb{N}_\perp \to \mathbb{N}_\perp$ is:

$$1 \quad 1 \quad 2 \quad 6 \quad 24 \quad 120 \quad \perp \quad \cdots$$

The interpretation of the factorial is then the limit of $(F^n(\perp))_n$.

A notion of limit linked to the order is necessary.

Quick stop: Recursion in Classical Reversibility

Read about it: [CLV23].

Quick stop: Recursion in Classical Reversibility

Suppose we have a syntax with:

- Only reversible functions,
- Inductive types,
- A fixpoint operator,
- A CbV operational semantics.

It is Turing-complete! (proven by Kostia Chardonnet)

Sound and adequate denotational semantics in partial injective functions.

Given two sets A and B, the set $[A \rightharpoonup B]$ has nice properties for fixpoints.

Read about it: [CLV23].

Quick stop: Recursion in Classical Reversibility

Suppose we have a syntax with:

- Only reversible functions,
- Inductive types,
- A fixpoint operator,
- A CbV operational semantics.

It is Turing-complete! (proven by Kostia Chardonnet)

Sound and adequate denotational semantics in partial injective functions.

Given two sets A and B, the set $[A \rightharpoonup B]$ has nice properties for fixpoints.

(The category of partial injective functions between sets is enriched in pointed dcpos.)

Read about it: [CLV23].

The Quantum Case

A unitary operator f is such that the inverse of f is its adjoint.

Dimension 2 for simple examples.

Reversible quantum operations are subunitaries such as:

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \quad \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & \frac{1}{\sqrt{2}} \\ 0 & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$$

The Quantum Case

A unitary operator f is such that the inverse of f is its adjoint.

Dimension 2 for simple examples.

Reversible quantum operations are subunitaries such as:

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \quad \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & \frac{1}{\sqrt{2}} \\ 0 & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$$

Arguably, undefined ($\perp$) in a Hilbert space is the zero vector $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$.

The Quantum Case

A unitary operator f is such that the inverse of f is its adjoint.

Dimension 2 for simple examples.

Reversible quantum operations are subunitaries such as:

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \quad \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & \frac{1}{\sqrt{2}} \\ 0 & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$$

Arguably, undefined ($\perp$) in a Hilbert space is the zero vector $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$.

$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$ is less defined than $\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$ which is less defined than $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$.

A disappointment

However, this order does not interact well with composition:

$$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \leq \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

thus $$\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} \circ \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \circ \begin{pmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{pmatrix} \leq \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} \circ \begin{pmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{pmatrix},$$

i.e. $$1/2 \leq 0.$$

A disappointment

However, this order does not interact well with composition:

$$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \leq \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

thus $$\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} \circ \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \circ \begin{pmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{pmatrix} \leq \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} \circ \begin{pmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{pmatrix},$$

i.e. $$1/2 \leq 0.$$

Theorem (No go theorem)

There is no such order on subunitary maps between Hilbert spaces.