An Algebra of Pure Quantum Programming

based on joint work with:
Jonathan Grattage, Amr Sabry, Juliana Vizzotto

Thorsten Altenkirch
University of Nottingham
QML

Functional core language for quantum programming

Finitary, rst order

Main goal: Investigate high level structures for quantum programming in a simple setting.

Reversible and irreversible programs

Operational semantics:
quantum circuits with heap and garbage

Denotational semantics:
superoperators on finite dimensional spaces
QML

- Functional core language for quantum programming
QML

- Functional core language for quantum programming
- Finitary, first order

\[ \sigma = Q_1 \mid Q_2 \mid \sigma \otimes \tau \]
QML

- Functional core language for quantum programming
- Finitary, first order

\[ \sigma = Q_1 \mid Q_2 \mid \sigma \otimes \tau \]

- Main goal: Investigate high level structures for quantum programming in a simple setting.
QML

- Functional core language for quantum programming
- Finitary, first order

\[ \sigma = Q_1 | Q_2 | \sigma \otimes \tau \]

- Main goal: Investigate high level structures for quantum programming in a simple setting.
- Reversible and irreversible programs
QML

- Functional core language for quantum programming
- Finitary, first order

\[ \sigma = Q_1 \mid Q_2 \mid \sigma \otimes \tau \]

- Main goal: Investigate high level structures for quantum programming in a simple setting.
- Reversible and irreversible programs
- Operational semantics: quantum circuits with heap and garbage
QML

- Functional core language for quantum programming
- Finitary, first order

\[ \sigma = Q_1 \mid Q_2 \mid \sigma \otimes \tau \]

- Main goal: Investigate high level structures for quantum programming in a simple setting.
- Reversible and irreversible programs
- Operational semantics: quantum circuits with heap and garbage
- Denotational semantics: superoperators on finite dimensional spaces
Main design ideas
Main design ideas

- Contraction by sharing
Contraction by sharing

\[ \delta \in Q_2 \rightarrow Q_2 \otimes Q_2 \]
\[ \delta \, x = (x, x) \]
Contraction by sharing

\[ \delta \in Q_2 \rightarrow Q_2 \otimes Q_2 \]

\[ \delta \ x = (x, x) \]

\[ \delta \ (\text{false} +_Q \text{true}) \not\equiv (\text{false} +_Q \text{true}, \text{false} +_Q \text{true}) \]
Contraction by sharing

\[ \delta \in Q_2 \rightarrow Q_2 \otimes Q_2 \]
\[ \delta x = (x, x) \]
\[ \delta (\text{false} +_Q \text{true}) \not\equiv (\text{false} +_Q \text{true}, \text{false} +_Q \text{true}) \]
\[ \delta (\text{false} +_Q \text{true}) \equiv (\text{false}, \text{false}) +_Q (\text{true}, \text{true}) \]
Main design ideas

- Contraction by sharing
Main design ideas

- Contraction by sharing
- Explicit weakening by measurement
Explicit weakening by measurement
Explicit weakening by measurement

\[ \pi_1 \in Q_2 \otimes Q_2 \rightarrow Q_2 \]

\[ \pi_1 (x, y) = x \uparrow \{ y \} \]
Explicit weakening by measurement

\[ \pi_1 \in Q_2 \otimes Q_2 \rightarrow Q_2 \]
\[ \pi_1 \ xy = \text{let} \ (x, y) = xy \text{ in } x \uparrow \{ y \} \]
Explicit weakening by measurement

\[ \pi_1 \in Q_2 \otimes Q_2 \rightarrow Q_2 \]

\[ \pi_1 \ xy = \textbf{let} \ (x, y) = xy \ \textbf{in} \ x \uparrow \{y\} \]

\[ \pi_1 (\delta \ x) \equiv x? \]
Explicit weakening by measurement

\[ \pi_1 \in Q_2 \otimes Q_2 \rightarrow Q_2 \]

\[ \pi_1 \ xy = \textbf{let} \ (x, y) = xy \ \textbf{in} \ x \uparrow \{ y \} \]

\[ \pi_1 \ (\delta \ x) \equiv x? \]

\[ \pi_1 \ (\delta \ (\text{false} +_Q \text{true} )) \equiv \text{false} +_P \text{true} \]
Main design ideas

- Contraction by sharing
- Explicit weakening by measurement
Main design ideas

- Contraction by sharing
- Explicit weakening by measurement
- Reversible $i f^\circ$ and irreversible $i f$. 
Reversible \( \text{if}^o \) and irreversible \( \text{if} \)

\[
\neg \in Q_2 \rightarrow Q_2
\]

\[
\neg x = \text{if}^o x \text{ then false else true}
\]
Reversible $\text{if}^\circ$ and irreversible $\text{if}$

$$\neg \in Q_2 \rightarrow Q_2$$
$$\neg x = \text{if}^\circ x \text{ then } false \text{ else } true$$
$$\neg (\neg x) \equiv x$$
Reversible if° and irreversible if

\[ \neg \in Q_2 \rightarrow Q_2 \]
\[ \neg x = \text{if}^\circ x \text{ then } false \text{ else } true \]
\[ \neg (\neg x) \equiv x \]
\[ \neg^c \in Q_2 \rightarrow Q_2 \]
\[ \neg^c x = \text{if } x \text{ then } false \text{ else } true \]
\[ \neg^c (\neg^c (false +_Q true)) \equiv false +_P true \]
Why do we need `if`?
Why do we need if?

\[ c\text{swap} \in Q_2 \to Q_2 \otimes Q_2 \to Q_2 \otimes Q_2 \]

\[ c\text{swap} \, x \, (y, z) = \text{if}^o \, x \, \text{then} \, (z, y) \, \text{else} \, (y, z) \]
Why do we need if?

\[ c_{\text{swap}} \in Q_2 \rightarrow Q_2 \otimes Q_2 \rightarrow Q_2 \otimes Q_2 \]

\[ c_{\text{swap}} \ x \ (y, z) = \text{if}^\circ \ x \ \text{then} \ (z, y) \ \text{else} \ (y, z) \]

is not well-typed, because we cannot show \((z, y) \perp (y, z)\).
Why do we need if?

\[ cswap \in Q_2 \multimap Q_2 \otimes Q_2 \multimap Q_2 \otimes Q_2 \]

\[ cswap \; x \; (y, z) = \textbf{if}^* \; x \; \textbf{then} \; (z, y) \; \textbf{else} \; (y, z) \]

is not well-typed, because we cannot show \((z, y) \perp (y, z)\).

\[ cswap' \in Q_2 \multimap Q_2 \otimes Q_2 \multimap Q_2 \otimes Q_2 \]

\[ cswap' \; x \; (y, z) = \textbf{if} \; x \; \textbf{then} \; (z, y) \; \textbf{else} \; (y, z) \]

is well-typed, since \textbf{if} does not require orthogonality.
An algebra of quantum programs?
An algebra of quantum programs?

- QPL 2005: restricted to the pure fragment (no weakening, no if)
  Denotational semantics: isometries

- Extends the rules for the classical sublanguage (no superpositions, and behave the same)
  Denotational semantics: sets and (injective) functions
  Sound and complete.
  Completeness also gives rise to a normalisation algorithm (Normalisation by evaluation).
An algebra of quantum programs?

- QPL 2005: restricted to the pure fragment (no weakening, no if)
  Denotational semantics: isometries
- Extends the rules for the classical sublanguage (no superpositions, if and if° behave the same)
  Denotational semantics: sets and (injective) functions

Sound and complete. Completeness also gives rise to a normalisation algorithm (Normalisation by evaluation).
An algebra of quantum programs?

- QPL 2005: restricted to the pure fragment (no weakening, no if)
  Denotational semantics: isometries
- Extends the rules for the classical sublanguage (no superpositions, if and if° behave the same)
  Denotational semantics: sets and (injective) functions
- Sound and complete.
An algebra of quantum programs?

- QPL 2005: restricted to the pure fragment (no weakening, no if)
  Denotational semantics: isometries
- Extends the rules for the classical sublanguage (no superpositions, if and if° behave the same)
  Denotational semantics: sets and (injective) functions
- Sound and complete.
- Completeness also gives rise to a normalisation algorithm (Normalisation by evaluation).
Example

\[ H \in Q_2 \rightarrow Q_2 \]

\[ H \ x = \text{if}^\circ \ x \]

then \( (false + (-1) * true) \)

else \( (false + true) \)
Example

\[ H \in \mathcal{Q}_2 \rightarrow \mathcal{Q}_2 \]
\[ H \; x = \text{if}^\circ \; x \]
\[ \quad \text{then} \; (false + (-1) \ast true) \]
\[ \quad \text{else} \; (false + true) \]

\[ \vdash H \; (H \; x) \equiv x \]
Derivation

\[ H (H \ x) = \text{if}^\circ (\text{if}^\circ x \text{then } (\text{false} + (-1) \times \text{true}) \text{else } (\text{false} + \text{true})) \text{then } (\text{false} + (-1) \times \text{true}) \text{else } (\text{false} + \text{true}) \]

-- by commuting conversion for \( \text{if}^\circ \)

\[ = \text{if}^\circ x \text{then } \text{if}^\circ (\text{false} + (-1) \times \text{true}) \text{then } (\text{false} + (-1) \times \text{true}) \text{else } (\text{false} + \text{true}) \text{else } \text{if}^\circ (\text{false} + \text{true}) \text{then } (\text{false} + (-1) \times \text{true}) \text{else } (\text{false} + \text{true}) \]
Derivation

\[ = \text{if}^\circ x \]
\[ \text{then if}^\circ (false + (-1) \times true) \]
\[ \text{then (false + (-1) \times true)} \]
\[ \text{else (false + true)} \]
\[ \text{else if}^\circ (false + true) \]
\[ \text{then (false + (-1) \times true)} \]
\[ \text{else (false + true)} \]

-- by if^\circ

\[ = \text{if}^\circ x \]
\[ \text{then (false - false + true + true)} \]
\[ \text{else (false + false + true - true)} \]
if° x
then (false − false + true + true)
else (false + false + true − true)

-- by simplification and normalisation

= if° x then true else false

-- by \(\eta\)-rule for if°

= x
Classical vs. quantum semantics
Classical vs. quantum semantics

Classical

\[
\Gamma \vdash^C t : \sigma \\
[t]^C \in [\Gamma] \rightarrow [\sigma]
\]
Classical vs. quantum semantics

**Classical**

\[
\Gamma \vdash^C t : \sigma \\
[t]^C \in [\Gamma] \rightarrow [\sigma]
\]

**Quantum**

\[
\Gamma \vdash^Q t : \sigma \\
[t]^Q \in Q^\circ \left[\Gamma\right] \left[\sigma\right]
\]
Classical vs. quantum semantics

Classical

\[
\Gamma \vdash_C t : \sigma \\
[t]^C \in [\Gamma] \rightarrow [\sigma]
\]

Quantum

\[
\Gamma \vdash_Q t : \sigma \\
[t]^Q \in Q^C [\Gamma] [\sigma]
\]

where \(Q^C A B\) is the set of linear, isometric \((\langle v|w \rangle = \langle f v|f w \rangle)\)
functions on the spaces with base \(A\) and \(B\) (finite).
Sound and complete (classical)
Soundness

\[
\Gamma \vdash^C t \equiv u : \sigma \\
\Leftrightarrow \quad [t]^C = [u]^C
\]
Sound and complete (classical)

Soundness \[ \Gamma \vdash^C t \equiv u : \sigma \]
\[ \llbracket t \rrbracket^C = \llbracket u \rrbracket^C \]

Completeness \[ \Gamma \vdash^C t, u : \sigma \]
\[ \llbracket t \rrbracket^C = \llbracket u \rrbracket^C \]
\[ \Gamma \vdash^C t \equiv u : \sigma \]
Sound and complete (quantum)

**Soundness**

\[ \Gamma \vdash^Q t \equiv u : \sigma \]

\[ [t]^Q = [u]^Q \]

**Completeness**

\[ \Gamma \vdash^Q t, u : \sigma \]

\[ [t]^Q = [u]^Q \]

\[ \Gamma \vdash^Q t \equiv u : \sigma \]
Classical equations for $\mathbf{if}^\circ$
Classical equations for $\text{if}^\circ$

\[ \beta \]

\[ \text{if}^\circ \text{ false then } t \text{ else } u \equiv u \]

\[ \text{if}^\circ \text{ true then } t \text{ else } u \equiv t \]
Classical equations for $\text{if}^\circ$

$$\beta$$

\[
\begin{align*}
\text{if}^\circ \text{false then } t \text{ else } u & \equiv u \\
\text{if}^\circ \text{true then } t \text{ else } u & \equiv t
\end{align*}
\]

$$\eta$$

\[
\begin{align*}
\text{if}^\circ \ t \text{ then } \text{true else } \text{false} & \equiv t
\end{align*}
\]
Classical equations for $\text{if}^\circ$

\[ \beta \]
\[ \text{if}^\circ \ false \ \text{then} \ t \ \text{else} \ u \equiv u \]
\[ \text{if}^\circ \ true \ \text{then} \ t \ \text{else} \ u \equiv t \]

\[ \eta \]
\[ \text{if}^\circ \ t \ \text{then} \ true \ \text{else} \ false \equiv t \]

Commuting conversion
\[ \text{let} \ p = \text{if}^\circ \ t \equiv \ \text{if}^\circ \ t \]
\[ \text{then} \ u_0 \ \text{then} \ \text{let} \ p = u_0 \ \text{in} \ e \]
\[ \text{else} \ u_1 \ \text{else} \ \text{let} \ p = u_1 \ \text{in} \ e \]
\[ \text{in} \ e \]
Classical equations for let
Classical equations for let

$$Val^C ::= x \mid () \mid false \mid true \mid (val_1, val_2)$$
Classical equations for let

\[ \text{Val}^C ::= x \mid () \mid \text{false} \mid \text{true} \mid (\text{val}_1, \text{val}_2) \]

\[ \beta \]

\[ \text{let } p = \text{val} \text{ in } u \equiv u[\text{val} / p] \]
Classical equations for \textit{let}

\[
\text{Val}^C ::= x \mid () \mid \text{false} \mid \text{true} \mid (\text{val}_1, \text{val}_2)
\]

\[
\beta
\]

\[
\text{let } p = \text{val} \text{ in } u \equiv u [\text{val} / p]
\]

\[
\eta
\]

\[
\text{let } x = t \text{ in } x \equiv t
\]
Classical equations for let

\[
\text{Val}^C ::= x \mid () \mid \text{false} \mid \text{true} \mid (\text{val}_1, \text{val}_2)
\]

\[\beta\]

let \( p = \text{val} \) in \( u \equiv u [\text{val} / p] \)

\[\eta\]

let \( x = t \) in \( x \equiv t \)

Commuting conversion

let \( p = t \) in let \( q = u \) in \( e \equiv \) let \( q = u \) in let \( p = t \) in \( e \)
Lemma (Adequacy)

The equation $q^\sigma \in [\sigma] \rightarrow \text{Val}^C \sigma$
Lemma (Adequacy)
The equation $\vdash^C q^\sigma([\vdash t : \sigma]^C) \equiv t : \sigma$ is derivable.
Lemma (Adequacy)
The equation $\vdash^C q^\sigma([\vdash t : \sigma]^C) \equiv t : \sigma$ is derivable.

We show

$$\gamma \in [\Gamma]^C \quad u = q^\Gamma \gamma$$

$$q^\sigma([t]^C \gamma) \equiv \text{let}^* \Gamma = u \text{ in } t$$

by induction over derivations.
\[ q_{\Gamma}^\sigma \in ([\Gamma] \rightarrow [\sigma]) \rightarrow Tm \Gamma \sigma \]
\[ q^\sigma_\Gamma \in ([\Gamma] \to [\sigma]) \to Tm \Gamma \sigma \]

\[ \phi_{\Gamma, x:Q_2} \in (Tm (\Gamma, x : Q_2) \sigma) \to (Tm \Gamma \sigma) \times (Tm \Gamma \sigma) \]

\[ \phi_{x:Q_2} t = (\text{let } x = \text{false in } t, \text{let } x = \text{true in } t) \]

\[ \phi_{\Gamma, x:Q_2}^{-1} (t, u) = \text{if }^\circ x \text{ then } t \text{ else } u \]
\[ q_\Gamma^\sigma \in ([\Gamma] \rightarrow [\sigma]) \rightarrow \text{Tm}\ \Gamma\ \sigma \]

\[ \phi_{\Gamma,x:Q_2} \in (\text{Tm}\ (\Gamma, x : Q_2)\ \sigma) \rightarrow (\text{Tm}\ \Gamma\ \sigma) \times (\text{Tm}\ \Gamma\ \sigma) \]

\[ \phi_{x:Q_2} t = (\text{let}\ x = \text{false}\ \text{in}\ t, \text{let}\ x = \text{true}\ \text{in}\ t) \]

\[ \phi_{\Gamma,x:Q_2}^{-1}(t,u) = \text{if}\ x\ \text{then}\ t\ \text{else}\ u \]

\[ q_\bullet(f) = q^\sigma f \]

\[ q_\Gamma^\sigma(x:Q_2)(f) = \phi_{\Gamma,x:Q_2}^{-1} \circ (q_\Gamma^\sigma \times q_\Gamma^\sigma) \circ [\phi_{\Gamma,x:Q_2}] \]
Normalisation and completeness
Normalisation and completeness

Definition

\[
\begin{align*}
\Gamma \vdash^C t : \sigma \\
\Rightarrow \quad \text{nf}_\Gamma^\sigma t = q_\Gamma^\sigma(\mathbb{I} \vdash t : \sigma]^C) 
\end{align*}
\]
Normalisation and completeness

**Definition**

\[
\Gamma \vdash^C t : \sigma \\
\text{nf}_\Gamma t = q_\Gamma^\sigma ([\Gamma \vdash t : \sigma]^C)
\]

**Lemma: Inversion**

\[
\Gamma \vdash^C t : \sigma \\
\Gamma \vdash^C \text{nf}_\Gamma^\sigma (t) \equiv t : \sigma
\]
Normalisation and completeness

Definition

\[ \frac{\Gamma \vdash^C t : \sigma}{\text{nf}_\Gamma^\sigma t = q^\sigma_\Gamma ([\Gamma \vdash t : \sigma]^C)} \]

Lemma: Inversion

\[ \frac{\Gamma \vdash^C t : \sigma}{\Gamma \vdash^C \text{nf}_\Gamma^\sigma(t) \equiv t : \sigma} \]

Proposition: Completeness

\[ \frac{\Gamma \vdash^C t, u : \sigma \quad [t]^C = [u]^C}{\Gamma \vdash^C t \equiv u : \sigma} \]
Quantum equations

\[(\text{if}^\circ)\]

\[
\text{if}^\circ (t_0 + t_1) \text{ then } u_0 \text{ else } u_1
\]

\[\equiv (\text{if}^\circ t_0 \text{ then } u_0 \text{ else } u_1) + (\text{if}^\circ t_1 \text{ then } u_0 \text{ else } u_1)\]

\[
\text{if}^\circ (\lambda * t) \text{ then } u_0 \text{ else } u_1
\]

\[\equiv \lambda * (\text{if}^\circ (\lambda * t) \text{ then } u_0 \text{ else } u_1)\]
Quantum equations

(if°)

if° (t₀ + t₁) then u₀ else u₁
≡ (if° t₀ then u₀ else u₁) + (if° t₁ then u₀ else u₁)

if° (λ * t) then u₀ else u₁
≡ λ * (if° (λ * t) then u₀ else u₁)

(superpositions)

\[ t + u \equiv u + t \]
\[ t + 0 \equiv t \]
\[ t + (u + v) \equiv (t + u) + v \]
\[ λ * (t + u) \equiv λ * t + λ * u \]
\[ λ * t + κ * t \equiv (λ + κ) * t \]
\[ 0 * t \equiv 0 \]
quote (quantum)
\( q^\sigma \in [\sigma]^Q \rightarrow \text{Val}^Q \sigma \)
\( q^\sigma \in [\sigma]^Q \rightarrow \text{Val}^Q \sigma \)

\[
q^{Q_2} \vec{v} = (\vec{v} 1) \ast \text{true} + (\vec{v} 0) \ast \text{false}
\]

\[
q^{\sigma \otimes \tau} \vec{v} = \ldots
\]
\[ q^\sigma \in [\sigma]^Q \rightarrow \text{Val}^Q \sigma \]

\[ q^{Q_2} \overrightarrow{\nu} = (\overrightarrow{\nu} 1) \ast \text{true} + (\overrightarrow{\nu} 0) \ast \text{false} \]

\[ q^{\sigma \otimes \tau} \overrightarrow{\nu} = \ldots \]

Diagram:

- \( (0,0) \) with \( \nu_{(0,0)} \) and \( \pi_1 \nu 0 \)
- \( (0,1) \) with \( \nu_{(0,1)} \) and \( \pi_1 \nu 0 \)
- \( (1,0) \) with \( \nu_{(1,0)} \) and \( \pi_1 \nu 1 \)
- \( (1,1) \) with \( \nu_{(1,1)} \) and \( \pi_1 \nu 1 \)
Adequacy (quantum)

Lemma (Adequacy)
The equation $\vdash^Q q^\sigma ([\vdash t : \sigma]^Q) \equiv t : \sigma$ is derivable.
Lemma (Adequacy)
The equation $\vdash^Q q^\sigma ([\vdash t : \sigma]^Q) \equiv t : \sigma$ is derivable.

We need

Lemma
$q^\sigma$ is linear and isometric and preserves $\otimes$, that is:

1. $q^\sigma (\kappa * \vec{v}) \equiv \kappa * (q^\sigma \vec{v})$
2. $q^\sigma (v + w) \equiv (q^\sigma v) + (q^\sigma w)$
3. $\langle v|w \rangle = \langle q^\sigma v|q^\sigma w \rangle$
4. $q^{\sigma \otimes \tau} v \otimes w \equiv (q^\sigma v, q^\tau w)$
Normalisation ... (quantum)
Normalisation ... (quantum)

Replace all Cs by Qs!
Normalisation . . . (quantum)

Replace all Cs by Qs!

Definition

\[
\Gamma \vdash^Q t : \sigma \\
\text{nf}_\Gamma^\sigma t = q_\Gamma^\sigma([\Gamma \vdash t : \sigma]^Q)
\]

Lemma: Inversion

\[
\Gamma \vdash^Q t : \sigma \\
\Gamma \vdash^Q \text{nf}_\Gamma^\sigma(t) \equiv t : \sigma
\]

Proposition: Qompleteness

\[
\Gamma \vdash^Q t, u : \sigma \\
[t]^Q = [u]^Q \\
\Gamma \vdash^Q t \equiv u : \sigma
\]
Conclusions and further work

The analogy with the classical construction proved very helpful.

- Extend to the full language (include and weakening).
- Higher order?
- Indexed types and programs?
- Useful to verify real programs?
Conclusions and further work

- The analogy with the classical construction proved very helpful.
Conclusions and further work

- The analogy with the classical construction proved very helpful.
- Extend to the full language (include `if` and weakening).
Conclusions and further work

- The analogy with the classical construction proved very helpful.
- Extend to the full language (include `if` and weakening).
- Higher order?
Conclusions and further work

- The analogy with the classical construction proved very helpful.
- Extend to the full language (include if and weakening).
- Higher order?
- Indexed types and programs?
Conclusions and further work

- The analogy with the classical construction proved very helpful.
- Extend to the full language (include `if` and weakening).
- Higher order?
- Indexed types and programs?
- Useful to verify real programs?