An overview of QML

Jonathan Grattage

ENS de Lyon
Research conducted at the University of Nottingham

June 2010

QML

Overview

- A high-level quantum language with a structure similar to functional languages
- Simplify the design of quantum algorithms:
  Allow formal reasoning principles
  Provide a more intuitive understanding

QML

Overview

- A high-level quantum language with a structure similar to functional languages
- Simplify the design of quantum algorithms:
  Allow formal reasoning principles
  Provide a more intuitive understanding

Design

- First-order, functional, quantum language
- "Quantum data and control"
- Based on strict linear logic: controlled, explicit, weakening
- Design guided by categorical semantics
- Controlling measurement

Classical vs. Quantum

Classical Case (**FCC**)	Quantum Case (**FQC**)
Finite sets	Finite dimensional Hilbert spaces
Cartesian product ($\times$)	Tensor product ($\otimes$)
Bijections	Unitary operators
Functions	Superoperators
Injective functions	Isometries
Projections	Partial trace

QML Syntax

- Types

$$\sigma = \mathcal{Q}_{\mathbf{1}} \mid \mathcal{Q}_{\mathbf{2}} \mid \sigma \otimes \tau$$

QML Syntax

- Types

$$\sigma = \mathcal{Q}_\mathbf{1} \mid \mathcal{Q}_\mathbf{2} \mid \sigma \otimes \tau$$

- Expressions

$$\begin{array}{llcl}
(\mathit{Variables}) & x, y, \ldots & \in & \mathit{Vars} \\
(\mathit{Prob.\ ampl}) & \kappa, \iota, \ldots & \in & \mathbb{C} \\
(\mathit{Patterns}) & p, q & ::= & x \mid (x, y) \\
(\mathit{Terms}) & t, u & ::= & x \mid x^{\vec{y}} \mid () \mid (t, u) \\
 & & \mid & \mathbf{let}\ p = t\ \mathbf{in}\ u \\
 & & \mid & \mathbf{if}\ \ t\ \mathbf{then}\ u\ \mathbf{else}\ u' \\
 & & \mid & \mathbf{if}^{\circ}\ t\ \mathbf{then}\ u\ \mathbf{else}\ u' \\
 & & \mid & \mathbf{qfalse} \mid \mathbf{qtrue} \mid \kappa \times t \mid t + u
\end{array}$$

QML Syntax

- Types

$$\sigma = \mathcal{Q}_{\mathbf{1}} \mid \mathcal{Q}_{\mathbf{2}} \mid \sigma \otimes \tau$$

- Expressions

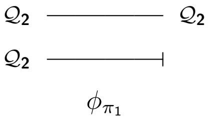
$$\begin{array}{lll}
(\mathit{Variables})\ x, y, \ldots & \in & \mathit{Vars} \\
(\mathit{Prob.\ ampl})\ \kappa, \iota, \ldots & \in & \mathbb{C} \\
(\mathit{Patterns})\ \ p, q & ::= & x \mid (x, y) \\
(\mathit{Terms})\ \ t, u & ::= & x \mid x^{\vec{y}} \mid () \mid (t, u) \\
 & \mid & \mathbf{let}\ p = t\ \mathbf{in}\ u \\
 & \mid & \mathbf{if}\ \ t\ \mathbf{then}\ u\ \mathbf{else}\ u' \\
 & \mid & \mathbf{if}^{\circ}\ t\ \mathbf{then}\ u\ \mathbf{else}\ u' \\
 & \mid & \mathbf{qfalse} \mid \mathbf{qtrue} \mid \kappa \times t \mid t + u
\end{array}$$

- EPR State $= (\mathbf{qfalse}, \mathbf{qfalse}) + (\mathbf{qtrue}, \mathbf{qtrue})$

Control of Weakening

- Projection Function

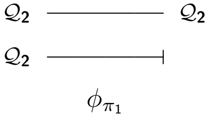
$$\pi_1 \in (\mathcal{Q}_\mathbf{2}, \mathcal{Q}_\mathbf{2}) \to \mathcal{Q}_\mathbf{2}$$
$$\pi_1\,(x, y) = x^y$$



Control of Weakening

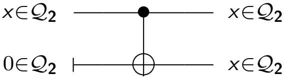
- Projection Function

$$\pi_1 \in (\mathcal{Q}_{\mathbf{2}}, \mathcal{Q}_{\mathbf{2}}) \to \mathcal{Q}_{\mathbf{2}}$$
$$\pi_1\ (x, y) = x^y$$



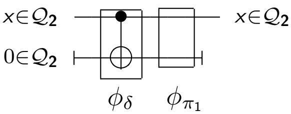
- Diagonal Function

$$\delta \in \mathcal{Q}_{\mathbf{2}} \to (\mathcal{Q}_{\mathbf{2}}, \mathcal{Q}_{\mathbf{2}})$$
$$\delta\ x = (x, x)$$



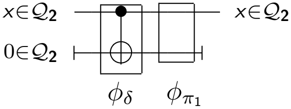
Control of Weakening

- $\pi_1 \circ \delta \in \mathcal{Q}_2 \to \mathcal{Q}_2$

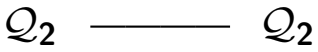


Control of Weakening

- $\pi_1 \circ \delta \in \mathcal{Q}_2 \to \mathcal{Q}_2$

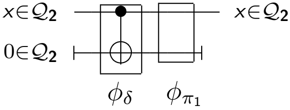


- Classical Case:



Control of Weakening

- $\pi_1 \circ \delta \in \mathcal{Q}_\mathbf{2} \to \mathcal{Q}_\mathbf{2}$



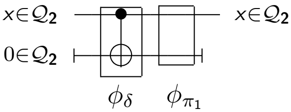
- Classical Case:

$$\mathcal{Q}_\mathbf{2} \quad \text{———} \quad \mathcal{Q}_\mathbf{2}$$

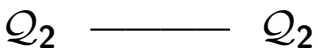
- Quantum Case:
  Input $= \frac{1}{\sqrt{2}} \times$ *false* $+ \frac{1}{\sqrt{2}} \times$ *true* (equal superposition)

Control of Weakening

- $\pi_1 \circ \delta \in \mathcal{Q}_2 \to \mathcal{Q}_2$



- Classical Case:



- Quantum Case:
  Input $= \frac{1}{\sqrt{2}} \times \mathit{false} + \frac{1}{\sqrt{2}} \times \mathit{true}$ (equal superposition)
  Output $= \{\frac{1}{2}\}\mathit{false} + \{\frac{1}{2}\}\mathit{true}$ (probability distribution)

More Weakening

- *forget* mentions x

$$\begin{aligned}&\mathit{forget} \in \mathcal{Q}_{\mathbf{2}} \multimap \mathcal{Q}_{\mathbf{2}}\\&\mathit{forget}\ x = \textbf{if}\ x\ \textbf{then qtrue else qtrue}\end{aligned}$$

More Weakening

- *forget* mentions x

$$\begin{array}{l} \mathit{forget} \in \mathcal{Q}_{\mathbf{2}} \multimap \mathcal{Q}_{\mathbf{2}} \\ \mathit{forget}\ x = \mathbf{if}\ x\ \mathbf{then\ qtrue\ else\ qtrue} \end{array}$$

- **if** always measures the conditional

More Weakening

- *forget* mentions x

$$
\begin{aligned}
&\mathit{forget} \in \mathcal{Q}_{\mathbf{2}} \multimap \mathcal{Q}_{\mathbf{2}} \\
&\mathit{forget}\ x = \mathbf{if}\ x\ \mathbf{then\ qtrue\ else\ qtrue}
\end{aligned}
$$

- **if** always measures the conditional
-

$$
\begin{aligned}
&\mathit{forget}' \in \mathcal{Q}_{\mathbf{2}} \multimap \mathcal{Q}_{\mathbf{2}} \\
&\mathit{forget}'\ x = \mathbf{if}^{\circ}\ x\ \mathbf{then\ qtrue\ else\ qtrue}
\end{aligned}
$$

More Weakening

- *forget* mentions x

$$\begin{array}{l} \mathit{forget} \in \mathcal{Q}_{\mathbf{2}} \multimap \mathcal{Q}_{\mathbf{2}} \\ \mathit{forget}\ x = \mathbf{if}\ x\ \mathbf{then\ qtrue\ else\ qtrue} \end{array}$$

- **if** always measures the conditional
-

$$\begin{array}{l} \mathit{forget}' \in \mathcal{Q}_{\mathbf{2}} \multimap \mathcal{Q}_{\mathbf{2}} \\ \mathit{forget}'\ x = \mathbf{if}^{\circ}\ x\ \mathbf{then\ qtrue\ else\ qtrue} \end{array}$$

- Type error: **true** $\not\perp$ **true**.

QML By Example: Conditionals

Not operations

$not_C, not_Q \in \mathcal{Q}_\mathbf{2} \multimap \mathcal{Q}_\mathbf{2}$

$not_C\ x = \textbf{if}\ \ x\ \textbf{then qfalse else qtrue}$ -- Classical

$not_Q\ x = \textbf{if}^\circ\ x\ \textbf{then qfalse else qtrue}$ -- Quantum

QML By Example: Conditionals

Not operations

$$not_C, not_Q \in \mathcal{Q}_\mathbf{2} \multimap \mathcal{Q}_\mathbf{2}$$
$$not_C\ x = \textbf{if}\ \ x\ \textbf{then qfalse else qtrue} \quad \text{-- Classical}$$
$$not_Q\ x = \textbf{if}^\circ\ x\ \textbf{then qfalse else qtrue} \quad \text{-- Quantum}$$

Quantum Controlled-Not

$$cnot \quad \in \mathcal{Q}_\mathbf{2} \multimap \mathcal{Q}_\mathbf{2} \otimes \mathcal{Q}_\mathbf{2}$$
$$cnot\ x = \textbf{if}^\circ\ x\ \textbf{then}\ (\textbf{qtrue}, not_Q\ x)\ \textbf{else}\ (\textbf{qfalse}, x)$$

QML By Example: Conditionals

Not operations

$$not_C, not_Q \in \mathcal{Q}_\mathbf{2} \multimap \mathcal{Q}_\mathbf{2}$$

$$not_C\ x = \textbf{if}\ \ x\ \textbf{then qfalse else qtrue} \quad \texttt{--}\ \text{Classical}$$

$$not_Q\ x = \textbf{if}^{\circ}\ x\ \textbf{then qfalse else qtrue} \quad \texttt{--}\ \text{Quantum}$$

Quantum Controlled-Not

$$cnot \in \mathcal{Q}_\mathbf{2} \multimap \mathcal{Q}_\mathbf{2} \otimes \mathcal{Q}_\mathbf{2}$$

$$cnot\ x = \textbf{if}^{\circ}\ x\ \textbf{then}\ (\textbf{qtrue}, not_Q\ x)\ \textbf{else}\ (\textbf{qfalse}, x)$$

Measurement

$$meas \in \mathcal{Q}_\mathbf{2} \multimap \mathcal{Q}_\mathbf{2}$$

$$meas\ x = \textbf{if}\ x\ \textbf{then qtrue else qfalse}$$

QML By Example: Conditionals

Not operations

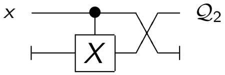
$$not_C, not_Q \in \mathcal{Q}_\mathbf{2} \multimap \mathcal{Q}_\mathbf{2}$$
$$not_C\ x = \textbf{if}\ \ x\ \textbf{then qfalse else qtrue} \quad \text{-- Classical}$$
$$not_Q\ x = \textbf{if}^\circ\ x\ \textbf{then qfalse else qtrue} \quad \text{-- Quantum}$$

Quantum Controlled-Not

$$cnot \in \mathcal{Q}_\mathbf{2} \multimap \mathcal{Q}_\mathbf{2} \otimes \mathcal{Q}_\mathbf{2}$$
$$cnot\ x = \textbf{if}^\circ\ x\ \textbf{then}\ (\textbf{qtrue}, not_Q\ x)\ \textbf{else}\ (\textbf{qfalse}, x)$$

Measurement

$$meas \in \mathcal{Q}_\mathbf{2} \multimap \mathcal{Q}_\mathbf{2}$$
$$meas\ x = \textbf{if}\ x\ \textbf{then qtrue else qfalse}$$



QML By Example: Conditionals

Not operations

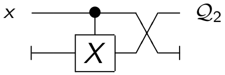
$$\mathit{not}_C, \mathit{not}_Q \in \mathcal{Q}_\mathbf{2} \multimap \mathcal{Q}_\mathbf{2}$$
$$\mathit{not}_C\ x = \mathbf{if}\ \ x\ \mathbf{then\ qfalse\ else\ qtrue} \quad \text{-- Classical}$$
$$\mathit{not}_Q\ x = \mathbf{if}^\circ\ x\ \mathbf{then\ qfalse\ else\ qtrue} \quad \text{-- Quantum}$$

Quantum Controlled-Not

$$\mathit{cnot} \in \mathcal{Q}_\mathbf{2} \multimap \mathcal{Q}_\mathbf{2} \otimes \mathcal{Q}_\mathbf{2}$$
$$\mathit{cnot}\ x = \mathbf{if}^\circ\ x\ \mathbf{then}\ (\mathbf{qtrue}, \mathit{not}_Q\ x)\ \mathbf{else}\ (\mathbf{qfalse}, x)$$

Measurement

$$\mathit{meas} \in \mathcal{Q}_\mathbf{2} \multimap \mathcal{Q}_\mathbf{2}$$
$$\mathit{meas}\ x = \mathbf{if}\ x\ \mathbf{then\ qtrue\ else\ qfalse}$$



EPR Pair

$$\mathit{epr} \in \mathcal{Q}_\mathbf{2} \otimes \mathcal{Q}_\mathbf{2}$$
$$\mathit{epr} = (\mathbf{qtrue}, \mathbf{qtrue}) + (\mathbf{qfalse}, \mathbf{qfalse})$$

QML By Example: Conditionals

Not operations

$not_C, not_Q \in \mathcal{Q}_\mathbf{2} \multimap \mathcal{Q}_\mathbf{2}$

$not_C\ x = \textbf{if}\ \ x\ \textbf{then qfalse else qtrue}$ -- Classical

$not_Q\ x = \textbf{if}^\circ\ x\ \textbf{then qfalse else qtrue}$ -- Quantum

Quantum Controlled-Not

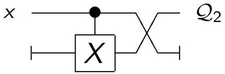
$cnot \in \mathcal{Q}_\mathbf{2} \multimap \mathcal{Q}_\mathbf{2} \otimes \mathcal{Q}_\mathbf{2}$

$cnot\ x = \textbf{if}^\circ\ x\ \textbf{then}\ (\textbf{qtrue}, not_Q\ x)\ \textbf{else}\ (\textbf{qfalse}, x)$

Measurement

$meas \in \mathcal{Q}_\mathbf{2} \multimap \mathcal{Q}_\mathbf{2}$

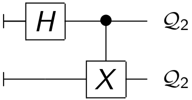
$meas\ x = \textbf{if}\ x\ \textbf{then qtrue else qfalse}$



EPR Pair

$epr \in \mathcal{Q}_\mathbf{2} \otimes \mathcal{Q}_\mathbf{2}$

$epr = (\textbf{qtrue}, \textbf{qtrue}) + (\textbf{qfalse}, \textbf{qfalse})$



Quantum control and orthogonality

- **if**$^{\circ}$ branches must be orthogonal

Quantum control and orthogonality

- **if**$^\circ$ branches must be orthogonal
- $$\frac{}{\textbf{qtrue} \perp \textbf{qfalse}} \qquad \frac{}{\textbf{qfalse} \perp \textbf{qtrue}}$$

Quantum control and orthogonality

- $\mathbf{if}^{\circ}$ branches must be orthogonal
- $$\frac{}{\mathbf{qtrue} \perp \mathbf{qfalse}} \qquad \frac{}{\mathbf{qfalse} \perp \mathbf{qtrue}}$$
- $$\frac{t \perp u}{(t, v) \perp (u, w)} \perp \mathbf{pair}_0 \qquad \frac{t \perp u}{(v, t) \perp (w, u)} \perp \mathbf{pair}_1$$

Quantum control and orthogonality

- $\mathbf{if}^\circ$ branches must be orthogonal
- $$\frac{}{\mathbf{qtrue} \perp \mathbf{qfalse}} \qquad \frac{}{\mathbf{qfalse} \perp \mathbf{qtrue}}$$
- $$\frac{t \perp u}{(t, v) \perp (u, w)} \perp \mathbf{pair}_0 \qquad \frac{t \perp u}{(v, t) \perp (w, u)} \perp \mathbf{pair}_1$$
- $$\frac{t \perp u \qquad t \perp u'}{t \perp \mathbf{if}^\circ\ c\ \mathbf{then}\ u\ \mathbf{else}\ u'} \perp \mathbf{if}^\circ_0 \qquad \frac{t \perp u \qquad t \perp u'}{\mathbf{if}^\circ\ c\ \mathbf{then}\ u\ \mathbf{else}\ u' \perp t} \perp \mathbf{if}^\circ_1$$

Quantum control and orthogonality

- $\mathbf{if}^\circ$ branches must be orthogonal
- $$\frac{}{\mathbf{qtrue} \perp \mathbf{qfalse}} \qquad \frac{}{\mathbf{qfalse} \perp \mathbf{qtrue}}$$
- $$\frac{t \perp u}{(t, v) \perp (u, w)} \perp \mathbf{pair}_0 \qquad \frac{t \perp u}{(v, t) \perp (w, u)} \perp \mathbf{pair}_1$$
- $$\frac{t \perp u \qquad t \perp u'}{t \perp \mathbf{if}^\circ\ c\ \mathbf{then}\ u\ \mathbf{else}\ u'} \perp \mathbf{if}^\circ_0 \quad \frac{t \perp u \qquad t \perp u'}{\mathbf{if}^\circ\ c\ \mathbf{then}\ u\ \mathbf{else}\ u' \perp t} \perp \mathbf{if}^\circ_1$$
- $$\frac{t \perp u \qquad \lambda_0^* \kappa_0 = -\lambda_1^* \kappa_1}{\lambda_0 \times t + \lambda_1 \times u \perp \kappa_0 \times t + \kappa_1 \times u} \perp \mathbf{sup}$$

Teleportation

Teleportation

$$
\begin{aligned}
&\mathit{tele} && \in \mathcal{Q}_{\mathbf{2}} \multimap \mathcal{Q}_{\mathbf{2}} \\
&\mathit{tele}\ q && = \mathbf{let}\ (a, b) = \mathit{epr} \\
& && \qquad\quad f \qquad = \mathit{bmeas}\ q\ a \quad \text{-- Alice} \\
& && \quad\ \mathbf{in}\ \mathit{corr}\ b\ f \qquad\qquad\qquad \text{-- Bob}
\end{aligned}
$$

Teleportation

$$\begin{aligned}
&\mathit{tele} && \in \mathcal{Q}_{\mathbf{2}} \multimap \mathcal{Q}_{\mathbf{2}} \\
&\mathit{tele}\ q && = \mathbf{let}\ (a, b) = \mathit{epr} \\
& && \qquad\quad f \qquad = \mathit{bmeas}\ q\ a \quad \text{-- Alice} \\
& && \quad\ \mathbf{in}\ \mathit{corr}\ b\ f \qquad\qquad\quad\ \text{-- Bob}
\end{aligned}$$

$$\begin{aligned}
&\mathit{bmeas} && \in \mathcal{Q}_{\mathbf{2}} \multimap \mathcal{Q}_{\mathbf{2}} \multimap \mathcal{Q}_{\mathbf{2}} \otimes \mathcal{Q}_{\mathbf{2}} \\
&\mathit{bmeas}\ x\ y && = \mathbf{let}\ (x', y') = \mathit{cnot}\ x\ y \\
& && \quad\ \mathbf{in}\ (\mathit{meas}\ (\mathit{had}\ x'), \mathit{meas}\ y')
\end{aligned}$$

Teleportation

$$
\begin{aligned}
&\mathit{tele} \quad \in \mathcal{Q}_2 \multimap \mathcal{Q}_2 \\
&\mathit{tele}\ q = \textbf{let}\ (a, b) = \mathit{epr} \\
&\qquad\qquad\quad f \qquad = \mathit{bmeas}\ q\ a \qquad \text{-- Alice} \\
&\qquad\qquad \textbf{in}\ \mathit{corr}\ b\ f \qquad\qquad\qquad \text{-- Bob}
\end{aligned}
$$

$$
\begin{aligned}
&\mathit{bmeas} \quad \in \mathcal{Q}_2 \multimap \mathcal{Q}_2 \multimap \mathcal{Q}_2 \otimes \mathcal{Q}_2 \\
&\mathit{bmeas}\ x\ y = \textbf{let}\ (x', y') = \mathit{cnot}\ x\ y \\
&\qquad\qquad\quad \textbf{in}\ (\mathit{meas}\ (\mathit{had}\ x'), \mathit{meas}\ y')
\end{aligned}
$$

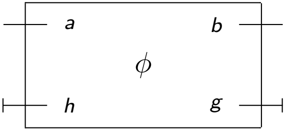
$$
\begin{aligned}
&\mathit{corr} \quad \in \mathcal{Q}_2 \multimap \mathcal{Q}_2 \otimes \mathcal{Q}_2 \multimap \mathcal{Q}_2 \\
&\mathit{corr}\ q\ xy = \textbf{let}\ (x, y) = xy\ \textbf{in if}\ x\ \textbf{then}\ (\textbf{if}\ y\ \textbf{then}\ U_{11}\ q\ \textbf{else}\ U_{10}\ q) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\quad \textbf{else}\ \ (\textbf{if}\ y\ \textbf{then}\ U_{01}\ q\ \textbf{else}\ q)
\end{aligned}
$$

$$
\begin{aligned}
&U_{01}, U_{10}, U_{11} \in \mathcal{Q}_2 \multimap \mathcal{Q}_2 \\
&U_{01}\ x = \textbf{if}^{\circ}\ x\ \textbf{then qfalse else qtrue} \\
&\ldots
\end{aligned}
$$

Operational Semantics: **QML** $\rightarrow$ **FQC**

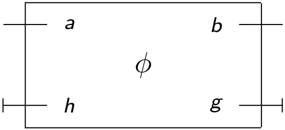
- Implemented in Haskell
- QML expressions compiled into **FQC** (Finite Quantum Computation) objects

Operational Semantics: **QML** $\rightarrow$ **FQC**

- Implemented in Haskell
- QML expressions compiled into **FQC** (Finite Quantum Computation) objects
- 

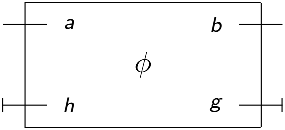
- ϕ = quantum circuit

Operational Semantics: **QML** → **FQC**

- Implemented in Haskell
- QML expressions compiled into **FQC** (Finite Quantum Computation) objects
- 

- ϕ = quantum circuit
- Circuit represented as simple combinators (Sequential/Parallel composition, permutations, conditionals, qubit rotations)

Operational Semantics: **QML** → **FQC**

- Implemented in Haskell
- QML expressions compiled into **FQC** (Finite Quantum Computation) objects
- 

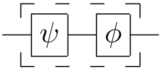
- ϕ = quantum circuit
- Circuit represented as simple combinators (Sequential/Parallel composition, permutations, conditionals, qubit rotations)
- Can be directly simulated
- Denotational semantics: Superoperators / Isometries

Compiler output: FQC

- *Morphisms* in $\mathbf{FQC}^{\simeq}$ a are characterised inductively

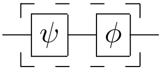
Compiler output: FQC

- *Morphisms* in $\mathbf{FQC}^{\simeq}$ *a* are characterised inductively
- Sequential Composition: $\phi \in \mathbf{FQC}^{\simeq}$ *a*, $\psi \in \mathbf{FQC}^{\simeq}$ *a*, then $\phi \circ \psi \in \mathbf{FQC}^{\simeq}$ *a* can be constructed.

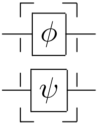


Compiler output: FQC

- *Morphisms* in $\mathbf{FQC}^{\simeq}\ a$ are characterised inductively
- Sequential Composition: $\phi \in \mathbf{FQC}^{\simeq}\ a$, $\psi \in \mathbf{FQC}^{\simeq}\ a$, then $\phi \circ \psi \in \mathbf{FQC}^{\simeq}\ a$ can be constructed.

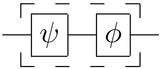


- Parallel Composition: $\phi \otimes \psi : \mathbf{FQC}^{\simeq}\ (a \otimes b)$

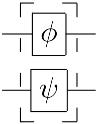


Compiler output: FQC

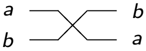
- *Morphisms* in $\mathbf{FQC}^{\simeq}$ *a* are characterised inductively
- Sequential Composition: $\phi \in \mathbf{FQC}^{\simeq}$ *a*, $\psi \in \mathbf{FQC}^{\simeq}$ *a*, then $\phi \circ \psi \in \mathbf{FQC}^{\simeq}$ *a* can be constructed.



- Parallel Composition: $\phi \otimes \psi : \mathbf{FQC}^{\simeq}\ (a \otimes b)$

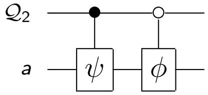


- Reordering: *wire* $\phi \in \mathbf{FQC}^{\simeq}$ *a* where $\phi : [a] \simeq [a]$ is a bijection



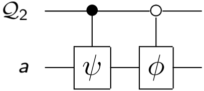
Compiler output: FQC

- Conditional: Given $\phi, \psi \in \mathbf{FQC}^{\simeq}\ a$, then $\phi | \psi \in \mathbf{FQC}^{\simeq}\ (1 \otimes a)$ can be constructed



Compiler output: FQC

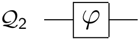
- Conditional: Given $\phi, \psi \in \textbf{FQC}^{\simeq}\ a$, then $\phi|\psi \in \textbf{FQC}^{\simeq}\ (1 \otimes a)$ can be constructed



- Rotation: *rot u* $\in \textbf{FQC}^{\simeq}\ 1$, where *u* is a unitary operation

$$\begin{pmatrix} \lambda_0 & \lambda_1 \\ \kappa_0 & \kappa_1 \end{pmatrix}$$

with $\lambda_0^* \kappa_0 + \lambda_1^* \kappa_1 = 0$

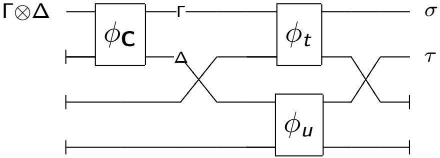


Example: Pairs of terms (t, u)

- $$\frac{\Gamma \vdash t : \sigma \quad \Delta \vdash u : \tau}{\Gamma \otimes \Delta \vdash (t, u) : \sigma \otimes \tau} \otimes \text{intro}$$

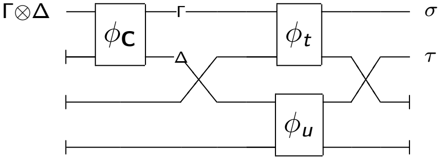
Example: Pairs of terms (t, u)

- $$\frac{\Gamma \vdash t : \sigma \quad \Delta \vdash u : \tau}{\Gamma \otimes \Delta \vdash (t, u) : \sigma \otimes \tau} \otimes \text{ intro}$$

- 

Example: Pairs of terms (t, u)

- $$\frac{\Gamma \vdash t : \sigma \quad \Delta \vdash u : \tau}{\Gamma \otimes \Delta \vdash (t, u) : \sigma \otimes \tau} \otimes \text{ intro}$$

- 

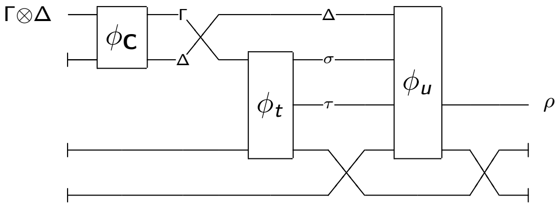
- $$\frac{\begin{array}{c}\mathbf{t} \in \mathbf{FQC}\ \Gamma\ \sigma \\ \mathbf{u} \in \mathbf{FQC}\ \Delta\ \tau\end{array}}{\begin{array}{l}\text{PAIR}_{\text{Op}}\ \mathbf{t}\ \mathbf{u} \in \mathbf{FQC}\ (\Gamma \otimes \Delta)\ (\sigma \otimes \tau) \\ \text{PAIR}_{\text{Op}}\ \mathbf{t}\ \mathbf{u} = (h_{\mathbf{C}} + h_{\mathbf{t}} + h_{\mathbf{u}}, g_{\mathbf{t}} + g_{\mathbf{u}}, \phi)\end{array}}$$

Example: $[\![\Gamma \otimes \Delta \vdash^a \textbf{let } (x, y) = t \textbf{ in } u : \rho]\!]^a_{\mathrm{Op}}$

- $$\frac{\Gamma \vdash^a t : \sigma \otimes \tau \qquad \Delta, x : \sigma, y : \tau \vdash^a u : \rho}{\Gamma \otimes \Delta \vdash^a \textbf{let } (x, y) = t \textbf{ in } u : \rho} \otimes \mathrm{elim}$$

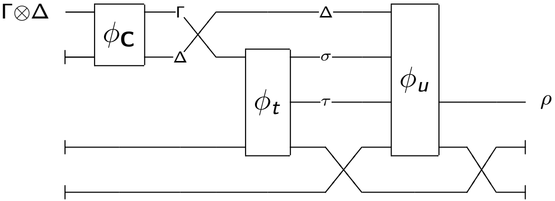
Example: $[\![\Gamma \otimes \Delta \vdash^a \textbf{let } (x, y) = t \textbf{ in } u : \rho]\!]^a_{\mathrm{Op}}$

- $$\frac{\Gamma \vdash^a t : \sigma \otimes \tau \qquad \Delta, x : \sigma, y : \tau \vdash^a u : \rho}{\Gamma \otimes \Delta \vdash^a \textbf{let } (x, y) = t \textbf{ in } u : \rho} \otimes \text{elim}$$

- 

Example: $[\![\Gamma \otimes \Delta \vdash^a \textbf{let } (x, y) = t \textbf{ in } u : \rho]\!]^a_{\mathrm{Op}}$

- $$\dfrac{\begin{array}{c}\Gamma \vdash^a t : \sigma \otimes \tau \\ \Delta, x : \sigma, y : \tau \vdash^a u : \rho\end{array}}{\Gamma \otimes \Delta \vdash^a \textbf{let } (x, y) = t \textbf{ in } u : \rho} \otimes \text{ elim}$$

- 

- $$\dfrac{\begin{array}{c}\mathbf{t} \in \mathbf{FQC}^a\ \Gamma\ (\sigma \otimes \tau) \\ \mathbf{u} \in \mathbf{FQC}^a\ (\Delta \otimes \sigma \otimes \tau)\ \rho\end{array}}{\begin{array}{l}\mathrm{LETP}^a_{\mathrm{Op}}\ \mathbf{t}\ \mathbf{u} \in \mathbf{FQC}^a\ (\Gamma \otimes \Delta)\ \rho \\ \mathrm{LETP}^a_{\mathrm{Op}}\ \mathbf{t}\ \mathbf{u} = (h_{\mathbf{C}} + h_{\mathbf{t}} + h_{\mathbf{u}}, g_{\mathbf{t}} + g_{\mathbf{u}}, \phi)\end{array}}$$

An algebra for (pure) QML

- A sound and complete equational theory for QML
- Proof of completeness gives rise to a normalisation algorithm (normalisation by evaluation)
- Focuses on the pure fragment of QML (omitting measurements)

An algebra for (pure) QML

- A sound and complete equational theory for QML
- Proof of completeness gives rise to a normalisation algorithm (normalisation by evaluation)
- Focuses on the pure fragment of QML (omitting measurements)
- $\textbf{if}^{\circ}\ (\lambda\ t_0 + \kappa\ t_1)\ \textbf{then}\ u_0\ \textbf{else}\ u_1\ =$ $\lambda\ (\textbf{if}^{\circ}\ t_0\ \textbf{then}\ u_0\ \textbf{else}\ u_1) + \kappa\ (\textbf{if}^{\circ}\ t_1\ \textbf{then}\ u_0\ \textbf{else}\ u_1)$

Algebra Example: *had* (*had* x)

- How can *had* (*had* x) $=_{obs}$ x be verified?

Algebra Example: *had* (*had* x)

- How can *had* (*had* x) $=_{obs}$ x be verified?
  - $= \textbf{if}^\circ\ (\textbf{if}^\circ\ x\ \textbf{then}\ (\textbf{qfalse} - \textbf{qtrue})\ \textbf{else}\ (\textbf{qfalse} + \textbf{qtrue}))$
    $\textbf{then}\ (\textbf{qfalse} - \textbf{qtrue})\ \textbf{else}\ (\textbf{qfalse} + \textbf{qtrue})$

Algebra Example: *had* (*had* x)

- How can $had\ (had\ x) =_{obs} x$ be verified?
  - $= \textbf{if}^\circ\ (\textbf{if}^\circ\ x\ \textbf{then}\ (\textbf{qfalse} - \textbf{qtrue})\ \textbf{else}\ (\textbf{qfalse} + \textbf{qtrue}))$
    $\textbf{then}\ (\textbf{qfalse} - \textbf{qtrue})\ \textbf{else}\ (\textbf{qfalse} + \textbf{qtrue})$
  - -- by commuting conversion for $\textbf{if}^\circ$
    $= \textbf{if}^\circ\ x\ \textbf{then}\ \textbf{if}^\circ\ (\textbf{qfalse} - \textbf{qtrue})\ \textbf{then}\ (\textbf{qfalse} - \textbf{qtrue})$
    $\textbf{else}\ (\textbf{qfalse} + \textbf{qtrue})$
    $\textbf{else}\ \textbf{if}^\circ\ (\textbf{qfalse} + \textbf{qtrue})\ \textbf{then}\ (\textbf{qfalse} - \textbf{qtrue})$
    $\textbf{else}\ (\textbf{qfalse} + \textbf{qtrue})$

Algebra Example: *had* (*had* *x*)

- How can $had\ (had\ x) =_{obs} x$ be verified?
  - $$= \textbf{if}^\circ\ (\textbf{if}^\circ\ x\ \textbf{then}\ (\textbf{qfalse} - \textbf{qtrue})\ \textbf{else}\ (\textbf{qfalse} + \textbf{qtrue}))$$
    $$\textbf{then}\ (\textbf{qfalse} - \textbf{qtrue})\ \textbf{else}\ (\textbf{qfalse} + \textbf{qtrue})$$
  - -- by commuting conversion for **if**$^\circ$
    $$= \textbf{if}^\circ\ x\ \textbf{then}\ \textbf{if}^\circ\ (\textbf{qfalse} - \textbf{qtrue})\ \textbf{then}\ (\textbf{qfalse} - \textbf{qtrue})$$
    $$\textbf{else}\ (\textbf{qfalse} + \textbf{qtrue})$$
    $$\textbf{else}\ \textbf{if}^\circ\ (\textbf{qfalse} + \textbf{qtrue})\ \textbf{then}\ (\textbf{qfalse} - \textbf{qtrue})$$
    $$\textbf{else}\ (\textbf{qfalse} + \textbf{qtrue})$$
  - -- by **if**$^\circ$
    $$= \textbf{if}^\circ\ x\ \textbf{then}\ (\textbf{qfalse} - \textbf{qfalse} + \textbf{qtrue} + \textbf{qtrue})$$
    $$\textbf{else}\ (\textbf{qfalse} + \textbf{qfalse} + \textbf{qtrue} - \textbf{qtrue})$$

Algebra Example: *had* (*had* x)

- How can *had* (*had* x) $=_{obs}$ x be verified?
  - $= \textbf{if}^\circ\ (\textbf{if}^\circ\ x\ \textbf{then}\ (\textbf{qfalse} - \textbf{qtrue})\ \textbf{else}\ (\textbf{qfalse} + \textbf{qtrue}))$
    $\textbf{then}\ (\textbf{qfalse} - \textbf{qtrue})\ \textbf{else}\ (\textbf{qfalse} + \textbf{qtrue})$
  - -- by commuting conversion for $\textbf{if}^\circ$
    $= \textbf{if}^\circ\ x\ \textbf{then}\ \textbf{if}^\circ\ (\textbf{qfalse} - \textbf{qtrue})\ \textbf{then}\ (\textbf{qfalse} - \textbf{qtrue})$
    $\textbf{else}\ (\textbf{qfalse} + \textbf{qtrue})$
    $\textbf{else}\ \textbf{if}^\circ\ (\textbf{qfalse} + \textbf{qtrue})\ \textbf{then}\ (\textbf{qfalse} - \textbf{qtrue})$
    $\textbf{else}\ (\textbf{qfalse} + \textbf{qtrue})$
  - -- by $\textbf{if}^\circ$
    $= \textbf{if}^\circ\ x\ \textbf{then}\ (\textbf{qfalse} - \textbf{qfalse} + \textbf{qtrue} + \textbf{qtrue})$
    $\textbf{else}\ (\textbf{qfalse} + \textbf{qfalse} + \textbf{qtrue} - \textbf{qtrue})$
  - -- by simplification and normalisation
    $= \textbf{if}^\circ\ x\ \textbf{then}\ \textbf{qtrue}\ \textbf{else}\ \textbf{qfalse}$

Algebra Example: *had* (*had* *x*)

- How can $had\ (had\ x) =_{obs} x$ be verified?
  - $= \mathbf{if}^\circ\ (\mathbf{if}^\circ\ x\ \mathbf{then}\ (\mathbf{qfalse} - \mathbf{qtrue})\ \mathbf{else}\ (\mathbf{qfalse} + \mathbf{qtrue}))$
    $\mathbf{then}\ (\mathbf{qfalse} - \mathbf{qtrue})\ \mathbf{else}\ (\mathbf{qfalse} + \mathbf{qtrue})$
  - -- by commuting conversion for $\mathbf{if}^\circ$
    $= \mathbf{if}^\circ\ x\ \mathbf{then}\ \mathbf{if}^\circ\ (\mathbf{qfalse} - \mathbf{qtrue})\ \mathbf{then}\ (\mathbf{qfalse} - \mathbf{qtrue})$
    $\mathbf{else}\ (\mathbf{qfalse} + \mathbf{qtrue})$
    $\mathbf{else}\ \mathbf{if}^\circ\ (\mathbf{qfalse} + \mathbf{qtrue})\ \mathbf{then}\ (\mathbf{qfalse} - \mathbf{qtrue})$
    $\mathbf{else}\ (\mathbf{qfalse} + \mathbf{qtrue})$
  - -- by $\mathbf{if}^\circ$
    $= \mathbf{if}^\circ\ x\ \mathbf{then}\ (\mathbf{qfalse} - \mathbf{qfalse} + \mathbf{qtrue} + \mathbf{qtrue})$
    $\mathbf{else}\ (\mathbf{qfalse} + \mathbf{qfalse} + \mathbf{qtrue} - \mathbf{qtrue})$
  - -- by simplification and normalisation
    $= \mathbf{if}^\circ\ x\ \mathbf{then}\ \mathbf{qtrue}\ \mathbf{else}\ \mathbf{qfalse}$
  - -- by η-rule for $\mathbf{if}^\circ$
    $= x$

Extensions

- Extend QML algebra to include measurement
- Extend orthogonality judgements
- Datastructures, more algorithms
- Add classical types, coproducts?

Extensions

- Extend QML algebra to include measurement
- Extend orthogonality judgements
- Datastructures, more algorithms
- Add classical types, coproducts?

- Project website: `fop.cs.nott.ac.uk/qml`
- Includes Haskell QML compiler (which generates circuits or superoperators from QML programs)

- Email `jonathan.grattage@ens-lyon.fr`