QPEL (Quantum Program and Effect language) is a formal system for denoting:

- quantum programs
- effects (‘fuzzy’ quantum predicates) of quantum systems

It is intended for reasoning about quantum programs, and proving properties such as correctness.
State-and-Effect Triangles

These structures are used to give semantics to quantum programs:

- Hilbert spaces
- C*-algebras
- W*-algebras
Many of these settings fit this pattern:

- A category whose objects represent *quantum systems*, and whose arrows represent *quantum programs*;
- An *effect algebra* $E$ of probabilities (typically $[0, 1]$);
- A collection of *effects* (predicates) over each object, which form an *effect module* over $E$;
- A collection of *states* for each object, which form a *convex set* over $E$.

Let us call this a *state-and-effect triangle*. [Jac14]
Many of these settings fit this pattern:

\[
\begin{align*}
\text{EMod}_{[0,1]}^\text{op} & \quad \perp \quad \text{Conv}_{[0,1]} \\
\text{CStar}_{PU}^\text{op} & \quad P \quad S
\end{align*}
\]

Let us call this a state-and-effect triangle. [Jac14]
State-and-Effect Triangles

Many of these settings fit this pattern:

QPEL is the logic of state-and-effect triangles.
QPEL — Syntax and Semantics

Syntax
Semantics

Qubits

Rules
Superdense Coding
Definition

An effect algebra is a structure \((E, \ominus, (\cdot)^\perp)\) where

- \(\ominus : E^2 \rightharpoonup E\) (partial)
- \((\cdot)^\perp : E \to E\)

such that

- \(x \ominus y \simeq y \ominus x\)
- \(x \ominus (y \ominus z) \simeq (x \ominus y) \ominus z\)
- \(x \ominus 0 = x\)
- \(x \ominus y = 0^\perp\) iff \(y = x^\perp\)
- If \(x \perp 0^\perp\) then \(x = 0\).

Let \(1 = 0^\perp\)

Examples:

- The set \(\{0, 1\}\) under \(x \ominus y = x + y\) if \(x + y \leq 1\), \(x^\perp = 1 - x\)
- The set \([0, 1]\) under \(x \ominus y = x + y\) if \(x + y \leq 1\), \(x^\perp = 1 - x\)
- Any Boolean algebra with \(x \ominus y = x \lor y\) if \(x \land y = 0\), \(x^\perp = \neg x\)
Definition (Effect Monoid)

An **effect monoid** is an effect algebra \( E \) with a (total) operation \( \cdot : E^2 \to E \) such that:

- \( (x \oplus y) \cdot z \sim (x \cdot z) \oplus (y \cdot z) \)
- \( x \cdot (y \oplus z) \sim (x \cdot y) \oplus (x \cdot z) \)
- \( 1 \cdot x = x \cdot 1 = x \)
- \( x \cdot (y \cdot z) = (x \cdot y) \cdot z \)

Examples:

- \{0, 1\} and \([0, 1]\) under multiplication.
- Any Boolean algebra under \( \land \).
Definition (Effect Module)

An effect module over the effect monoid $E$ is an effect algebra $A$ with a (total) operation $\cdot : E \times A \rightarrow A$ such that:

- $r \cdot (x \lor y) \sim (r \cdot x) \lor (r \cdot y)$
- $(r \lor s) \cdot x \sim (r \cdot x) \lor (s \cdot x)$
- $(r \cdot s) \cdot x = r \cdot (s \cdot x)$
- $1 \cdot x = x$

Examples:

- The effects over a Hilbert space (positive operators less than $I$) form an effect module over $[0, 1]$.
- The effects in a $C^*$-algebra (positive elements below 1) form an effect module over $[0, 1]$. 
Definition (Convex Set)

A convex set consists of a set $X$ and an operation: given $r_1, \ldots, r_n \in E$ with

$$r_1 \searrow \cdots \searrow r_n = 1$$

and $x_1, \ldots, x_n \in X$, returns an element

$$r_1 x_1 + \cdots + r_n x_n \in X$$

such that certain equations hold.

Examples

- The density matrices over a Hilbert space form a convex set over $[0, 1]$.
- For $A$ a $\mathbb{C}^*$-algebra, the positive unital maps $A \to \mathbb{C}$ form a convex set over $[0, 1]$.
State-and-Effect Triangles

The functors

\[
\text{Conv}_E[-, E] : \text{Conv}_E^{\text{op}} \to E\text{Mod}_E
\]
\[
\text{EMod}_E[-, E] : \text{EMod}_E^{\text{op}} \to \text{Conv}_E
\]

form an adjunction.
A state-and-effect triangle is a structure

\[
\text{EMod}_E^{\text{op}} \quad \perp \quad \text{Conv}_E
\]

where:

- \( E \) is an effect monoid
- \( \mathcal{V} \) is a symmetric monoidal category with binary coproducts that distribute over \( \otimes \) such that the tensor unit \( I \) is terminal
- \( P \) preserves finite coproducts and the terminal object
- \( S \) is a symmetric monoidal functor such that certain coherence conditions hold.
A state-and-effect triangle is a structure

\[
\begin{array}{c}
\text{EMod}_E^{\text{op}} \\
\downarrow \\
\text{Conv}_E
\end{array}
\]

where:

- given \( r_1 \otimes \cdots \otimes r_n = 1 \) in \( PA \), an arrow \( \text{meas}_A(r_1, \ldots, r_n) : A \rightarrow n \cdot I \) in \( \mathcal{V} \)
- natural transformations

\[
\alpha : P \rightarrow \text{Conv}_E[S- , E], \quad \beta : S \rightarrow \text{EMod}_E[P- , E]
\]

such that certain coherence conditions hold.
Examples

\[ \text{EMod}_{[0,1]}^{op} \quad \perp \quad \text{Conv}_{[0,1]} \]

\[ \text{CStar}_{PU}^{op} \]

\[ \text{EMod}_{[0,1]}^{op} \quad \perp \quad \text{Conv}_{[0,1]} \]

\[ KI(D) \]
Syntax of QPEL

Type $A ::= A \otimes A \mid I \mid A + B$

- Terms $s, t, \ldots$ intended to represent quantum programs.
- Effects $\phi, \psi, \ldots$ intended to represent predicates on quantum states.

Judgement forms:
- $\Gamma \vdash t : A$
- $\Gamma \vdash s = t : A$
- $\Gamma \vdash \phi$ eff
- $\Gamma \vdash \phi \leq \psi$
This is a *linear* type system – no Contraction:

\[
\Gamma, x : A, y : A \vdash t[x, y] : B \\
\Gamma, x : A \vdash t[x, x] : B
\]

Allowing Contraction would violate the no-cloning theorem
Effects

Effect Formation

\[
\begin{align*}
\Gamma \vdash 0 \text{ eff} \\
\Gamma \vdash \phi \text{ eff} \\
\Gamma \vdash \phi \perp \text{ eff} \\
\Gamma \vdash \phi \leq \psi \perp \text{ eff} \\
\Gamma \vdash \phi \nabla \psi \text{ eff} \\
\Gamma \vdash \phi \text{ eff} \\
\Gamma \vdash \psi \text{ eff} \\
\Gamma \vdash \phi \cdot \psi \text{ eff}
\end{align*}
\]

Derivability

\[
\begin{align*}
\Gamma \vdash \phi \text{ eff} \\
\Gamma \vdash \phi \leq \phi \perp \text{ eff} \\
\Gamma \vdash \phi \leq \psi \text{ eff} \\
\Gamma \vdash \psi \leq \chi \text{ eff} \\
\Gamma \vdash \phi \leq \chi \text{ eff} \\
\Gamma \vdash \phi \perp \leq \phi \perp \perp \text{ eff} \\
\Gamma \vdash \phi \perp \perp \leq \phi \text{ eff}
\end{align*}
\]

The *scalars* are the effects in the empty context
Typing System

Structural Rules

\[
\frac{\Gamma, x : A, y : B, \Delta \vdash J}{\Gamma, y : B, x : A, \Delta \vdash J} \quad \frac{x : A \vdash x : A}{x : A \vdash x : A}
\]

Tensor Products

\[
\frac{\Gamma \vdash M : A \quad \Delta \vdash N : B}{\Gamma, \Delta \vdash \langle M, N \rangle : A \otimes B}
\]

\[
\frac{\Gamma \vdash M : A \otimes B \quad \Delta, x : A, y : B \vdash N : C}{\Gamma, \Delta \vdash \text{let } \langle x, y \rangle = M \text{ in } N : C}
\]
Measurement

\[
\Gamma \vdash 1 \leq \phi_1 \otimes \cdots \otimes \phi_n \quad \Delta \vdash M_1 : A \quad \cdots \quad \Delta \vdash M_n : A
\]

\[
\Gamma, \Delta \vdash \text{measure } \phi_1 \mapsto M_1 \mid \cdots \mid \phi_n \mapsto M_n : A
\]
Measurement

\[
\Gamma \vdash 1 \leq \phi_1 \otimes \cdots \otimes \phi_n \quad \Delta \vdash M_1 : A \quad \cdots \quad \Delta \vdash M_n : A
\]

\[
\Gamma, \Delta \vdash \text{measure } \phi_1 \mapsto M_1 \mid \cdots \mid \phi_n \mapsto M_n : A
\]

\[
\Gamma \vdash 1 \leq \phi_1 \otimes \cdots \otimes \phi_n
\]

\[
\Delta \vdash M_1 : A \quad \cdots \quad \Delta \vdash M_n : A
\]

\[
\Gamma, \Delta \vdash (\text{measure } \phi_1 \mapsto M_1 \mid \cdots \mid \phi_n \mapsto M_n) = (\text{measure } \phi_{p(1)} \mapsto M_{p(1)} \mid \cdots \mid \phi_{p(n)} \mapsto M_{p(n)})
\]
\[ \begin{align*}
\Gamma & \vdash 1 \leq \phi_1 \otimes \cdots \otimes \phi_n \\
\Delta & \vdash M_1 : A \\
& \cdots \\
\Delta & \vdash M_n : A \\
\Gamma, \Delta & \vdash \text{measure } \phi_1 \mapsto M_1 | \cdots | \phi_n \mapsto M_n : A \\
\end{align*} \]

\[ \begin{align*}
\Gamma & \vdash 1 \leq \phi_1 \otimes \cdots \otimes \phi_n \\
\Delta & \vdash M_1 : A \\
& \cdots \\
\Delta & \vdash M_n : A \\
\Gamma, \Delta & \vdash \text{(measure } \phi_1 \mapsto M_1 | \cdots | \phi_n \mapsto M_n) = \\
(\text{measure } \phi_{p(1)} \mapsto M_{p(1)} | \cdots | \phi_{p(n)} \mapsto M_{p(n)}) \\
\end{align*} \]

\[ \begin{align*}
\Gamma & \vdash 1 \leq \phi_1 \otimes \cdots \otimes \phi_n \\
\Delta & \vdash M_1 : A \\
& \cdots \\
\Delta & \vdash M_{n+1} : A \\
\Gamma & \vdash \text{(measure } \phi_1 \mapsto M_1 | \cdots | \phi_n \mapsto M_n | 0 \mapsto M_{n+1}) = \\
\text{measure } \phi_1 \mapsto M_1 | \cdots | \phi_n \mapsto M_n : A \\
\end{align*} \]
\[
\Gamma \vdash M : A \\
\Gamma \vdash (\text{measure } 1 \mapsto M) = M : A
\]
Measurement

\[ \Gamma \vdash M : A \]
\[ \Gamma \vdash (\text{measure } 1 \mapsto M) = M : A \]

\[ \vdash 1 \leq \phi \oplus \psi \oplus \chi_1 \oplus \cdots \oplus \chi_n \quad \Gamma \vdash M : A \quad \Gamma \vdash P_1 : A \quad \cdots \]
\[ \Gamma \vdash (\text{measure } \phi \oplus \psi \mapsto M \mid \chi_1 \mapsto P_1 \mid \cdots \mid \chi_n \mapsto P_n) \]
\[ = (\text{measure } \phi \mapsto M \mid \psi \mapsto M \mid \chi_1 \mapsto P_1 \mid \cdots \mid \chi_n \mapsto P_n) \]
Semantics

Define:

• an object $[[A]] \in \mathcal{V}$ for each type $A$
• an object $[[\Gamma]] \in \mathcal{V}$ for each context $\Gamma$
• an arrow $[[M]] : [\Gamma] \to [A]$ for each term $\Gamma \vdash M : A$
• an element $[[\phi]] \in P[[\Gamma]]$ for each effect $\Gamma \vdash \phi$ eff
• an element $([[\phi]]) \in E$ for each effect $\vdash \phi$ eff.
Semantics

Define:

- an object $[[A]] \in \mathcal{V}$ for each type $A$
- an object $[[\Gamma]] \in \mathcal{V}$ for each context $\Gamma$
- an arrow $[[M]] : [[\Gamma]] \to [[A]]$ for each term $\Gamma \vdash M : A$
- an element $[[\phi]] \in P[[\Gamma]]$ for each effect $\Gamma \vdash \phi \text{ eff}$
- an element $([[\phi]]) \in E$ for each effect $\vdash \phi \text{ eff}$.

Example: If $\Gamma \vdash \phi_i \text{ eff}$ and $\Delta \vdash M_i : A$, then $[[\Gamma, \Delta \vdash \text{measure } \phi_1 \mapsto M_1 \mid \cdots \mid \phi_n \mapsto M_n : A]]$ is

$$[[\Gamma]] \otimes [[\Delta]] \xrightarrow{\text{meas}_A([[\phi_1]], \ldots, [[\phi_n]]), \otimes \mathbf{1}} n \cdot [[\Delta]] \xrightarrow{[[M_1]], \ldots, [[M_n]]} [[A]]$$
Theorem (Soundness)

Any derivable judgement is true in any state-and-effect triangle.
Completeness Theorem

Theorem (Soundness)

Any derivable judgement is true in any state-and-effect triangle.

Theorem (Completeness)

Any judgement that is true in every state-and-effect triangle is derivable.
Completeness Theorem

**Theorem (Soundness)**

Any derivable judgement is true in any state-and-effect triangle.

**Theorem (Completeness)**

Any judgement that is true in every state-and-effect triangle is derivable.

**Proof.**

Define a state-and-effect triangle as follows. The category $\mathcal{V}$ is the category with objects the types, and arrows $A \rightarrow B$ the terms $M$ such that $x : A \vdash M : B$, quotiented by:

$M = N$ iff $x : A \vdash M = N : B$. 

R Adams
5 June 2014
QPEL
Qubits

Based on the rules of the Measurement Calculus [DKPP09]. These can be interpreted in $FdHilb_{Un}$, $CStar^{op}$ and $WStar^{op}$, but not in an arbitrary state-and-effect triangle. Extend the system with:

\[
\begin{align*}
\text{Type } & A ::= \cdots \mid \text{qbit} \\
\downarrow & \\
\Gamma \vdash |0\rangle : \text{qbit} \\
\Gamma \vdash t : \text{qbit} & \quad \Gamma \vdash X t : \text{qbit} \\
\Gamma \vdash t : \text{qbit} & \quad \Gamma \vdash Z t : \text{qbit} \\
\Gamma \vdash s : \text{qbit} & \quad \Delta \vdash t : \text{qbit} \\
\Gamma, \Delta \vdash E s t : \text{qbit} \otimes \text{qbit} \\
\Gamma \vdash t : \text{qbit} & \quad \Gamma \vdash (t = |+\alpha\rangle) \text{ eff} \\
(0 \leq \alpha < 2\pi) \\
\end{align*}
\]

Define:
\[
|1\rangle = Z|0\rangle \quad (x = |1\rangle) = (x = |+0\rangle)^\perp
\]
Equations for Qubits

\[
E(\text{Xs})t = \text{let} \langle x, y \rangle = Est \text{ in } \langle \text{Xx}, \text{Zy} \rangle \\
E(\text{Zs})t = \text{let} \langle x, y \rangle = Est \text{ in } \langle \text{Zx}, y \rangle \\
(Xt = |+\alpha\rangle) = (t = |+_{-\alpha}\rangle) \\
(Zt = |+\alpha\rangle) = (t = |+_\alpha_{-\pi}\rangle) \\
X(Xt) = t \\
Z(Zt) = t \\
(t = |+\alpha\rangle)^\perp = (t = |+_{-\alpha}\rangle) \\
(X(Zt) = |+\alpha\rangle) = (Z(Xt) = |+\alpha\rangle)
\]
Alice prepares two entangled qubits in the state $|b_1\rangle = 1/\sqrt{2}(|00\rangle + |11\rangle)$ and sends one to Bob. She wishes to send an integer $1 \leq i \leq 4$ to Bob. She performs an operation on her qubit:

<table>
<thead>
<tr>
<th>$i$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation</td>
<td>I</td>
<td>X</td>
<td>Z</td>
<td>XZ</td>
</tr>
</tbody>
</table>

She then sends this qubit to Bob. Bob measures the pair of qubits in the basis $\{ |b_1\rangle, |b_2\rangle, |b_3\rangle, |b_4\rangle \}$ and learns the value of $i$. 
Let

\[ Ht = \text{let } \langle x, y \rangle = Et|1\rangle \text{ in } \]
\[ \text{measure } \]
\[ x = |0\rangle \mapsto Xy | \]
\[ x = |1\rangle \mapsto y \]

\[ \text{CNOT } s \ t = \text{let } \langle x, y \rangle = Es(Ht) \text{ in } \]
\[ \langle x, Hy \rangle \]

Add the axioms:

\[ H(Ht) = t \]
\[ \text{let } \langle x, y \rangle = \text{CNOT } s \ t \]
\[ \text{in } \text{CNOT } x \ y = \langle s, t \rangle \]
Axioms for \texttt{qbit} \otimes \texttt{qbit}

Let

\[ |e_1\rangle = |11\rangle, |e_2\rangle = |10\rangle, |e_3\rangle = |01\rangle, |e_4\rangle = |00\rangle. \]

Add the axioms:

\[ q : \texttt{qbit} \otimes \texttt{qbit} \vdash 1 \leq q = |e_1\rangle \otimes q = |e_2\rangle \otimes q = |e_3\rangle \otimes q = |e_4\rangle \]

\[ \vdash 1 \leq |e_i\rangle = |e_i\rangle \]

\[ \vdash (|e_i\rangle = |e_j\rangle) \leq 0 \quad (i \neq j) \]

\[ (\langle X(Zs), t \rangle = |e_i\rangle) = (\langle Z(Xs), t \rangle = |e_i\rangle) \]
\[ z : I + I + I + I + I \vdash sdc(z) : I + I + I + I \]

\[ sdc(z) \equiv \text{let } \langle x, y \rangle = \text{CNOT}(H|1\rangle)(|1\rangle) \text{ in} \]

\[ \text{let } t_A = \text{case } z \text{ of} \]

\[ 1 \mapsto \langle x, y \rangle \]

\[ 2 \mapsto \langle Xx, y \rangle \]

\[ 3 \mapsto \langle Zx, y \rangle \]

\[ 4 \mapsto \langle XZx, y \rangle \text{ in} \]

\[ \text{measure} \]

\[ t_A = |b_1\rangle \mapsto 1 | \]

\[ t_A = |b_2\rangle \mapsto 2 | \]

\[ t_A = |b_3\rangle \mapsto 3 | \]

\[ t_A = |b_4\rangle \mapsto 4 \]
Here

\[ |b_1\rangle = \text{CNOT}(H|1\rangle)|1\rangle \]
\[ |b_2\rangle = \text{CNOT}(H|1\rangle)|0\rangle \]
\[ |b_3\rangle = \text{CNOT}(H|0\rangle)|1\rangle \]
\[ |b_4\rangle = \text{CNOT}(H|0\rangle)|0\rangle \]

Let

\((q = |b_i\rangle) \equiv (\text{let } \langle x, y \rangle = q \text{ in}
\text{let } \langle x, y \rangle = \text{CNOT} x y \text{ in}
\langle Hx, y \rangle) = |e_i\rangle\)

Then

\[ q : qbit \otimes qbit \vdash 1 \leq q = |b_1\rangle \otimes \cdots \otimes q = |b_4\rangle \]
\[ \vdash 1 \leq (|b_i\rangle = |b_i\rangle) \]
\[ \vdash (|b_i\rangle = |b_j\rangle) \leq 0 \quad (i \neq j) \]
\[ sdc(3) = \text{let } \langle x, y \rangle = \text{CNOT}(H |1\rangle)(|1\rangle) \text{ in } \]
\[ \text{let } t_A = \langle Zx, y \rangle \text{ in } \]
\[ \text{measure } \]
\[ t_A = |b_1\rangle \mapsto 1 | \]
\[ t_A = |b_2\rangle \mapsto 2 | \]
\[ t_A = |b_3\rangle \mapsto 3 | \]
\[ t_A = |b_4\rangle \mapsto 4 \]
\[ sdc(3) = \text{let } \langle x, y \rangle = \text{CNOT}(H |1\rangle)(|1\rangle) \text{ in} \]

\[
\begin{align*}
\langle Zx, y \rangle &= |b_1\rangle \mapsto 1 \\
\langle Zx, y \rangle &= |b_2\rangle \mapsto 2 \\
\langle Zx, y \rangle &= |b_3\rangle \mapsto 3 \\
\langle Zx, y \rangle &= |b_4\rangle \mapsto 4
\end{align*}
\]
\[ sdc(3) = \text{let } \langle x, y \rangle = \text{CNOT}(H|1\rangle)(|1\rangle) \text{ in } \]
\[ \text{measure } \]
\[ \langle Zx, y \rangle = |b_i\rangle \mapsto i \]
\[ sdc(3) = \text{let } \langle x, y \rangle = \text{CNOT}(H|1\rangle)(|1\rangle) \text{ in} \]
\[ \text{measure} \]
\[ \text{let } \langle x, y \rangle = \text{CNOT}(Zx)y \text{ in} \]
\[ \langle Hx, y \rangle = |e_i\rangle \mapsto i \]
\[ sdc(3) = \text{let } \langle x, y \rangle = \text{CNOT}(H|1\rangle)|1\rangle \text{ in } \]
\[ \text{measure} \]
\[ \text{let } \langle x, y \rangle = \text{CNOT}_{xy} \text{ in } \]
\[ \text{measure} \]
\[ \langle H(Zx), y \rangle = |e_i\rangle \mapsto i \]
\[ sdc(3) = \text{measure} \]
\[ \langle H(Z(H|1\rangle)), |1\rangle \rangle = |e_i\rangle \mapsto i \]
\[ sdc(3) = \text{measure} \]
\[ \langle H(H|0\rangle), |1\rangle \rangle = |e_i\rangle \mapsto i \]
\[ sdc(3) = \text{measure} \]
\[ \langle |0\rangle, |1\rangle = |e_i\rangle \mapsto i \]
sdc(3) = measure

\[ |0\rangle, |1\rangle = |1\rangle, |1\rangle \mapsto 1 \]
\[ |0\rangle, |1\rangle = |1\rangle, |0\rangle \mapsto 2 \]
\[ |0\rangle, |1\rangle = |0\rangle, |1\rangle \mapsto 3 \]
\[ |0\rangle, |1\rangle = |0\rangle, |0\rangle \mapsto 4 \]
\( sdc(3) \quad \text{=measure} \)

- 0 \( \mapsto \) 1
- 0 \( \mapsto \) 2
- 1 \( \mapsto \) 3
- 0 \( \mapsto \) 4
\[ sdc(3) = \text{measure} \]
\[ 1 \mapsto 3 \]
\[ sdc(3) = 3 \]
\[ z : I + I + I + I \vdash sdc(z) = z : I + I + I + I \]
Related Work

• Baltag and Smets [BS04, B14] work on LQP (Logic of Quantum Programs)
  • Based on propositional dynamic logic
  • Includes $[P] \phi$ — ‘after $P$, $\phi$ is true’
  • Language for terms/states is an underspecification language.
Related Work

- Baltag and Smets [BS04, B14] work on LQP (Logic of Quantum Programs)
  - Based on propositional dynamic logic
  - Includes $[P] \phi$ — ‘after $P$, $\phi$ is true’
  - Language for terms/states is an underspecification language.
- d’Hondt, Panangaden and Ying [dP06, Yin11] give a Floyd-Hoare logic for quantum programs.
  - includes $\{\phi\} P \{\psi\}$ — ‘if $\phi$ is true before $P$ is run, then $\psi$ will be true after’
  - Syntax for quantum programs
  - No syntax for logic — predicates are operators on Hilbert spaces
QPEL is a sound and complete system for state-and-effect triangles. Within its framework, we can give a theory of qubits and reason about quantum programs.

For the future:

- Complete axiomatization of qubits.
Conclusion

QPEL is a sound and complete system for state-and-effect triangles. Within its framework, we can give a theory of qubits and reason about quantum programs. For the future:

- Complete axiomatization of qubits.
- Incorporate $[\phi?]\psi$ and $\langle\phi?\rangle\psi$ into logic.
QPEL is a sound and complete system for state-and-effect triangles. Within its framework, we can give a theory of qubits and reason about quantum programs.

For the future:

- Complete axiomatization of qubits.
- Incorporate $[\phi?]\psi$ and $\langle\phi?\rangle\psi$ into logic.
- Make more use of the state space — three judgement forms?
Conclusion

QPEL is a sound and complete system for state-and-effect triangles. Within its framework, we can give a theory of qubits and reason about quantum programs.

For the future:

- Complete axiomatization of qubits.
- Incorporate $[\phi?]\psi$ and $\langle \phi? \rangle \psi$ into logic.
- Make more use of the state space — three judgement forms?
- Other triangles — classical logic, probabilistic logic, ...
Bibliography


