

A Quantum Programming Language for Coherent Control

Kathleen Barsse Romain Péchoux Simon Perdrix

Inria team Mocqua, LORIA, Nancy

Journées Informatique Quantique

January 15, 2026

Classical and quantum control

- **“Quantum data, classical control”**

The execution flow is based on classical information (for instance measurement outcomes).

Classical and quantum control

- **“Quantum data, classical control”**

The execution flow is based on classical information (for instance measurement outcomes).

- **“Quantum data, quantum control”**

The execution flow is based on quantum information. We allow for the superposition of processes during the execution of a program.

Classical and quantum control

- **“Quantum data, classical control”**

The execution flow is based on classical information (for instance measurement outcomes).

- **“Quantum data, quantum control”**

The execution flow is based on quantum information. We allow for the superposition of processes during the execution of a program.

Elements of the control structure:

- **if** statements
- **while** statements
- Iteration

An example of quantum control: qcise

The operation qcise acts on a pair of qubits (“control” and “target”).
For U and V unitary:

- If the control qubit is in state $|0\rangle$, apply U to the target qubit.
- If the control qubit is in state $|1\rangle$, apply V to the target qubit.
- Otherwise, apply U and V in a coherent superposition.

An example of quantum control: qcse

The operation qcse acts on a pair of qubits (“control” and “target”).
For U and V unitary:

- If the control qubit is in state $|0\rangle$, apply U to the target qubit.
- If the control qubit is in state $|1\rangle$, apply V to the target qubit.
- Otherwise, apply U and V in a coherent superposition.

$$\text{qcse}(U, V) := \left[\begin{array}{c|c} U & 0 \\ \hline 0 & V \end{array} \right]$$

An example of quantum control: qcse

The operation qcse acts on a pair of qubits (“control” and “target”). For U and V unitary:

- If the control qubit is in state $|0\rangle$, apply U to the target qubit.
- If the control qubit is in state $|1\rangle$, apply V to the target qubit.
- Otherwise, apply U and V in a coherent superposition.

$$\text{qcse}(U, V) := \left[\begin{array}{c|c} U & 0 \\ \hline 0 & V \end{array} \right]$$

The goal is to develop a programming language that supports coherent control.

Main difficulties

Combining qcase with...

- ① **Measurement**
- ② **Recursion**

Main difficulties

Combining qcase with...

① Measurement

- Defining qcase beyond the unitary case: $\text{qcase}(U, V) = \begin{bmatrix} U & 0 \\ 0 & V \end{bmatrix}$

$$\begin{bmatrix} U & 0 \\ 0 & V \end{bmatrix}$$

② Recursion

Main difficulties

Combining qcase with...

① Measurement

- Defining qcase beyond the unitary case: $\text{qcase}(U, V) = \begin{bmatrix} U & 0 \\ 0 & V \end{bmatrix}$

② Recursion

- Incompatibility of qcase with the Löwner ordering.

Costin Bădescu and Prakash Panangaden. “Quantum alternation: Prospects and problems”. In: *12th International Workshop on Quantum Physics and Logic (2015)*

A programming language for quantum control

Syntax

The syntax is inspired by the programming language QPL.

Statements

$$\begin{aligned} S, S_0, S_1 ::= & \text{ skip } | \text{ new qbit } q | \text{ discard } q | q* = U | S_0; S_1 \\ & | \text{ meas } q (0 \rightarrow S_0, 1 \rightarrow S_1) | \text{ while } q \text{ do } S \\ & | \text{ qcase } q (0 \rightarrow S_0, 1 \rightarrow S_1) \end{aligned}$$

where U is a single-qubit unitary map.

Peter Selinger. “Towards a quantum programming language”. In: *Mathematical Structures in Computer Science* 14.4 (2004), pp. 527–586

Syntax

The syntax is inspired by the programming language QPL.

Statements

$$\begin{aligned} S, S_0, S_1 ::= & \text{ skip } | \text{ new qbit } q | \text{ discard } q | q* = U | S_0; S_1 \\ & | \text{ meas } q (0 \rightarrow S_0, 1 \rightarrow S_1) | \text{ while } q \text{ do } S \\ & | \text{ qcase } q (0 \rightarrow S_0, 1 \rightarrow S_1) \end{aligned}$$

where U is a single-qubit unitary map.

- **meas** $q (0 \rightarrow S_0, 1 \rightarrow S_1)$ the qubit q , and executes S_0 or S_1 depending on the outcome.

Syntax

The syntax is inspired by the programming language QPL.

Statements

$$\begin{aligned} S, S_0, S_1 ::= & \text{ skip } | \text{ new qbit } q | \text{ discard } q | q* = U | S_0; S_1 \\ & | \text{ meas } q (0 \rightarrow S_0, 1 \rightarrow S_1) | \text{ while } q \text{ do } S \\ & | \text{ qcase } q (0 \rightarrow S_0, 1 \rightarrow S_1) \end{aligned}$$

where U is a single-qubit unitary map.

- **meas** $q (0 \rightarrow S_0, 1 \rightarrow S_1)$ the qubit q , and executes S_0 or S_1 depending on the outcome.
- **while** $q \text{ do } S$ is a loop where q is measured at every iteration.

Syntax

The syntax is inspired by the programming language QPL.

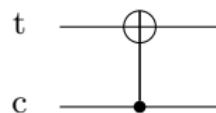
Statements

$$\begin{aligned} S, S_0, S_1 ::= & \text{ skip } | \text{ new qbit } q | \text{ discard } q | q* = U | S_0; S_1 \\ & | \text{ meas } q (0 \rightarrow S_0, 1 \rightarrow S_1) | \text{ while } q \text{ do } S \\ & | \text{ qcase } q (0 \rightarrow S_0, 1 \rightarrow S_1) \end{aligned}$$

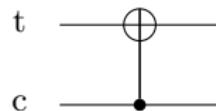
where U is a single-qubit unitary map.

- **meas** $q (0 \rightarrow S_0, 1 \rightarrow S_1)$ the qubit q , and executes S_0 or S_1 depending on the outcome.
- **while** $q \text{ do } S$ is a loop where q is measured at every iteration.
- **qcase** $q (0 \rightarrow S_0, 1 \rightarrow S_1)$ implements quantum control: S_0 and S_1 are executed in superposition depending on the state of q .

Example: CNOT



Example: CNOT



$S_{\text{CNOT}} := \text{qcase } c \ (0 \rightarrow \text{skip}, 1 \rightarrow t* = X)$

where $X := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$.

Operational semantics

Defining an operational semantics

The goal is to define the evolution of $|\psi\rangle$ induced by a program statement S :

$$[S, |\psi\rangle] \rightarrow |\psi'\rangle$$

Defining an operational semantics

The goal is to define the evolution of $|\psi\rangle$ induced by a program statement S :

$$[S, |\psi\rangle] \rightarrow |\psi'\rangle$$

$$\overline{[\text{new qbit } q, |\psi\rangle] \rightarrow |0\rangle_q \otimes |\psi\rangle}$$

$$\overline{[q* = U, |\psi\rangle] \rightarrow U_q |\psi\rangle}$$

Defining an operational semantics

The goal is to define the evolution of $|\psi\rangle$ induced by a program statement S :

$$[S, |\psi\rangle] \rightarrow |\psi'\rangle$$

$$\overline{[\mathbf{new} \; \mathbf{qbit} \; q, |\psi\rangle] \rightarrow |0\rangle_q \otimes |\psi\rangle} \quad \overline{[q* = U, |\psi\rangle] \rightarrow U_q |\psi\rangle}$$

$$\frac{[S_0, |0\rangle\langle 0|_q |\psi\rangle] \rightarrow |\psi'\rangle}{[\mathbf{meas} \; q \; (0 \rightarrow S_0, 1 \rightarrow S_1), |\psi\rangle] \rightarrow |\psi'\rangle} \quad (\text{M}_0)$$

$$\frac{[S_1, |1\rangle\langle 1|_q |\psi\rangle] \rightarrow |\psi'\rangle}{[\mathbf{meas} \; q \; (0 \rightarrow S_0, 1 \rightarrow S_1), |\psi\rangle] \rightarrow |\psi'\rangle} \quad (\text{M}_1)$$

Defining an operational semantics

The goal is to define the evolution of $|\psi\rangle$ induced by a program statement S :

$$[S, |\psi\rangle] \rightarrow |\psi'\rangle$$

$$\overline{[\text{new qbit } q, |\psi\rangle] \rightarrow |0\rangle_q \otimes |\psi\rangle} \quad \overline{[q* = U, |\psi\rangle] \rightarrow U_q |\psi\rangle}$$

$$\frac{[S_0, |0\rangle\langle 0|_q |\psi\rangle] \rightarrow |\psi'\rangle}{[\text{meas } q (0 \rightarrow S_0, 1 \rightarrow S_1), |\psi\rangle] \rightarrow |\psi'\rangle} \quad (M_0)$$

$$\frac{[S_1, |1\rangle\langle 1|_q |\psi\rangle] \rightarrow |\psi'\rangle}{[\text{meas } q (0 \rightarrow S_0, 1 \rightarrow S_1), |\psi\rangle] \rightarrow |\psi'\rangle} \quad (M_1)$$

A transition occurs with probability $\frac{\| |\psi'\rangle \|^2}{\| |\psi\rangle \|^2}$.

Semantics of **qcase**?

Consider the statement **qcase** q $(0 \rightarrow S_0, 1 \rightarrow S_1)$.

Semantics of **qcase**?

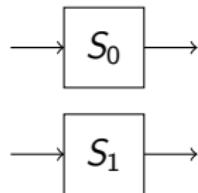
Consider the statement **qcase** q $(0 \rightarrow S_0, 1 \rightarrow S_1)$.

- If the input is $|0\rangle_q \otimes |\psi\rangle$:

Semantics of **qcase**?

Consider the statement **qcase** q $(0 \rightarrow S_0, 1 \rightarrow S_1)$.

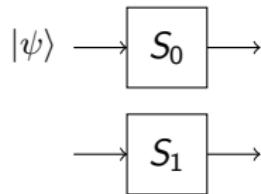
- If the input is $|0\rangle_q \otimes |\psi\rangle$:



Semantics of **qcase**?

Consider the statement **qcase** q $(0 \rightarrow S_0, 1 \rightarrow S_1)$.

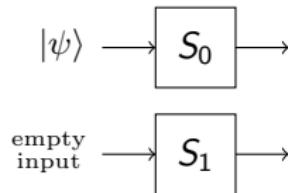
- If the input is $|0\rangle_q \otimes |\psi\rangle$:



Semantics of **qcase**?

Consider the statement **qcase** q $(0 \rightarrow S_0, 1 \rightarrow S_1)$.

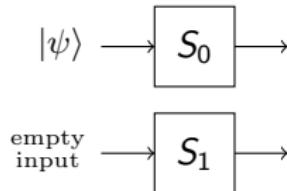
- If the input is $|0\rangle_q \otimes |\psi\rangle$:



Semantics of **qcase**?

Consider the statement **qcase** q $(0 \rightarrow S_0, 1 \rightarrow S_1)$.

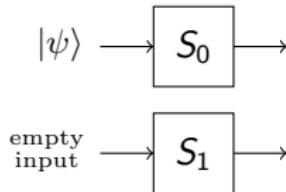
- If the input is $|0\rangle_q \otimes |\psi\rangle$:
- If the input is $|1\rangle_q \otimes |\psi\rangle$:



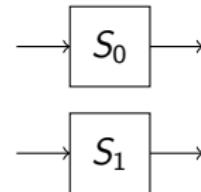
Semantics of **qcase**?

Consider the statement **qcase** q $(0 \rightarrow S_0, 1 \rightarrow S_1)$.

- If the input is $|0\rangle_q \otimes |\psi\rangle$:



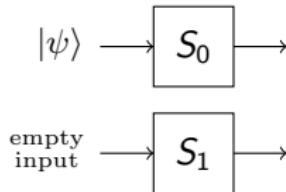
- If the input is $|1\rangle_q \otimes |\psi\rangle$:



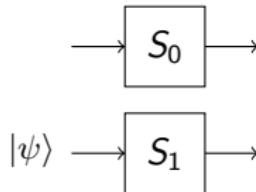
Semantics of **qcase**?

Consider the statement **qcase** q $(0 \rightarrow S_0, 1 \rightarrow S_1)$.

- If the input is $|0\rangle_q \otimes |\psi\rangle$:



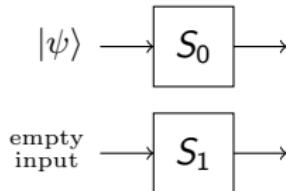
- If the input is $|1\rangle_q \otimes |\psi\rangle$:



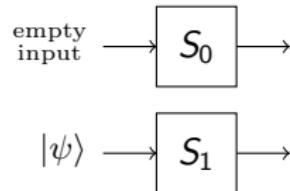
Semantics of **qcase**?

Consider the statement **qcase** q $(0 \rightarrow S_0, 1 \rightarrow S_1)$.

- If the input is $|0\rangle_q \otimes |\psi\rangle$:



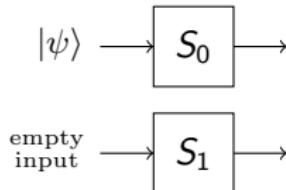
- If the input is $|1\rangle_q \otimes |\psi\rangle$:



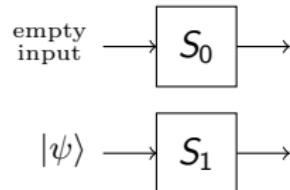
Semantics of **qcase**?

Consider the statement **qcase** q $(0 \rightarrow S_0, 1 \rightarrow S_1)$.

- If the input is $|0\rangle_q \otimes |\psi\rangle$:



- If the input is $|1\rangle_q \otimes |\psi\rangle$:

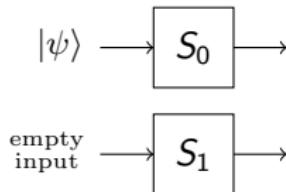


We define the action of S on the regular input $|\psi\rangle$ *and* on the empty input.

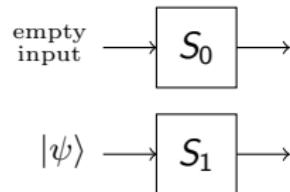
Semantics of **qcase**?

Consider the statement **qcase** q $(0 \rightarrow S_0, 1 \rightarrow S_1)$.

- If the input is $|0\rangle_q \otimes |\psi\rangle$:



- If the input is $|1\rangle_q \otimes |\psi\rangle$:



We define the action of S on the regular input $|\psi\rangle$ *and* on the empty input.

Therefore, we define default transitions.

Operational semantics

Transition system

Transitions have the form

$$[S, |\psi\rangle] \xrightarrow{\nu} |\psi'\rangle$$

- $\nu \in \{0, 1\}$ answers the question, “Is this the default transition?”

Operational semantics

Transition system

Transitions have the form

$$[S, |\psi\rangle] \xrightarrow{\nu} |\psi'\rangle$$

- $\nu \in \{0, 1\}$ answers the question, “Is this the default transition?”

$$[\mathbf{new \ qbit} \ q, |\psi\rangle] \xrightarrow{1} |0\rangle_q \otimes |\psi\rangle$$

$$[\mathbf{q* =} U, |\psi\rangle] \xrightarrow{1} U_q |\psi\rangle$$

Operational semantics

Transition system

Transitions have the form

$$[S, |\psi\rangle] \xrightarrow{\nu} |\psi'\rangle$$

- $\nu \in \{0, 1\}$ answers the question, “Is this the default transition?”

$$\overline{[\mathbf{new qbit} \ q, |\psi\rangle] \xrightarrow{1} |0\rangle_q \otimes |\psi\rangle}$$

$$\overline{[\mathbf{q*} = U, |\psi\rangle] \xrightarrow{1} U_q |\psi\rangle}$$

$$\overline{[S_0, |0\rangle\langle 0|_q |\psi\rangle] \xrightarrow{\nu} |\psi'\rangle} \quad (M_0)$$

$$\overline{[\mathbf{meas} \ q (0 \rightarrow S_0, 1 \rightarrow S_1), |\psi\rangle] \xrightarrow{\nu} |1\rangle\langle 1|_q |\psi\rangle} \quad (M_1)$$

$$[\mathbf{meas} \ q (0 \rightarrow S_0, 1 \rightarrow S_1), |\psi\rangle] \xrightarrow{0} |\psi'\rangle$$

$\left. \begin{array}{l} (M_0) \\ (M_1) \end{array} \right\} (M_0) \text{ is the default}$

Operational semantics

Transition system

Transitions have the form

$$[S, |\psi\rangle] \xrightarrow{\nu} |\psi'\rangle$$

- $\nu \in \{0, 1\}$ answers the question, “Is this the default transition?”

$$\overline{[\text{new qbit } q, |\psi\rangle] \xrightarrow{1} |0\rangle_q \otimes |\psi\rangle}$$

$$\overline{[q* = U, |\psi\rangle] \xrightarrow{1} U_q |\psi\rangle}$$

$$\left. \begin{array}{c} \overline{[S_0, |0\rangle\langle 0|_q |\psi\rangle] \xrightarrow{\nu} |\psi'\rangle} \\ \overline{[\text{meas } q (0 \rightarrow S_0, 1 \rightarrow S_1), |\psi\rangle] \xrightarrow{\nu} |\psi'\rangle} \\ \overline{[S_1, |1\rangle\langle 1|_q |\psi\rangle] \xrightarrow{\nu} |\psi'\rangle} \\ \overline{[\text{meas } q (0 \rightarrow S_0, 1 \rightarrow S_1), |\psi\rangle] \xrightarrow{0} |\psi'\rangle} \end{array} \right\} (M_0) \text{ is the default}$$

$$[S_0, \langle 0|_q |\psi\rangle] \xrightarrow{\nu_0} |\psi_0\rangle \quad [S_1, \langle 1|_q |\psi\rangle] \xrightarrow{\nu_1} |\psi_1\rangle$$

$$[\text{qcase } q (0 \rightarrow S_0, 1 \rightarrow S_1), |\psi\rangle] \xrightarrow{\nu_0 \nu_1} \nu_1 |0\rangle_q \otimes |\psi_0\rangle + \nu_0 |1\rangle_q \otimes |\psi_1\rangle$$

Denotational semantics

Denotational semantics

The goal is to assign to each program a mathematical operation that describes its behavior.

- First idea: Interpret programs as completely positive trace non-increasing maps:

$$\llbracket P \rrbracket = \mathcal{C}$$

Semantics of `qcase`?

Semantics of `qcase`?

- On unitaries:

$$\text{qcase}(U, V) = \begin{bmatrix} U & 0 \\ 0 & V \end{bmatrix}$$

Semantics of `qcase`?

- On unitaries:

$$\text{qcase}(U, V) = \begin{bmatrix} U & 0 \\ 0 & V \end{bmatrix}$$

- On unitary operations $\mathcal{U} = U(\cdot)U^\dagger$ and $\mathcal{V} = V(\cdot)V^\dagger$:

Semantics of `qcase`?

- On unitaries:

$$\text{qcase}(U, V) = \begin{bmatrix} U & 0 \\ 0 & V \end{bmatrix}$$

- On unitary operations $\mathcal{U} = U(\cdot)U^\dagger$ and $\mathcal{V} = V(\cdot)V^\dagger$:

$$\text{QCASE}(\mathcal{U}, \mathcal{V}) = \begin{bmatrix} \rho_1 & \rho_2 \\ \rho_3 & \rho_4 \end{bmatrix} \mapsto \begin{bmatrix} U\rho_1 U^\dagger & U\rho_2 V^\dagger \\ V\rho_3 U^\dagger & V\rho_4 V^\dagger \end{bmatrix}$$

Semantics of `qcase`?

- On unitaries:

$$\text{qcase}(U, V) = \begin{bmatrix} U & 0 \\ 0 & V \end{bmatrix}$$

- On unitary operations $\mathcal{U} = U(\cdot)U^\dagger$ and $\mathcal{V} = V(\cdot)V^\dagger$:

$$\text{QCASE}(\mathcal{U}, \mathcal{V}) = \begin{bmatrix} \rho_1 & \rho_2 \\ \rho_3 & \rho_4 \end{bmatrix} \mapsto \begin{bmatrix} U\rho_1 U^\dagger & U\rho_2 V^\dagger \\ V\rho_3 U^\dagger & V\rho_4 V^\dagger \end{bmatrix}$$

- Therefore:

$$\text{QCASE}(\mathcal{C}, \mathcal{D}) = \begin{bmatrix} \rho_1 & \rho_2 \\ \rho_3 & \rho_4 \end{bmatrix} \mapsto \begin{bmatrix} \mathcal{C}(\rho_1) & * \\ * & \mathcal{D}(\rho_4) \end{bmatrix} ?$$

Semantics of `qcase`?

- On unitaries:

$$\text{qcase}(U, V) = \begin{bmatrix} U & 0 \\ 0 & V \end{bmatrix}$$

- On unitary operations $\mathcal{U} = U(\cdot)U^\dagger$ and $\mathcal{V} = V(\cdot)V^\dagger$:

$$\text{QCASE}(\mathcal{U}, \mathcal{V}) = \begin{bmatrix} \rho_1 & \rho_2 \\ \rho_3 & \rho_4 \end{bmatrix} \mapsto \begin{bmatrix} U\rho_1 U^\dagger & U\rho_2 V^\dagger \\ V\rho_3 U^\dagger & V\rho_4 V^\dagger \end{bmatrix}$$

- Therefore:

$$\text{QCASE}(\mathcal{C}, \mathcal{D}) = \begin{bmatrix} \rho_1 & \rho_2 \\ \rho_3 & \rho_4 \end{bmatrix} \mapsto \begin{bmatrix} \mathcal{C}(\rho_1) & * \\ * & \mathcal{D}(\rho_4) \end{bmatrix} ?$$

This is not possible: $\text{QCASE}(\mathcal{C}, \mathcal{D})$ is ill-defined.

Solution: We interpret a statement S as a vacuum-extended operation. The domain is extended by a vector $|\text{vac}\rangle$ in a one-dimensional Hilbert space Vac .

Solution: We interpret a statement S as a vacuum-extended operation. The domain is extended by a vector $|\text{vac}\rangle$ in a one-dimensional Hilbert space Vac .

Vacuum-extended operations

A vacuum extension of a quantum operation \mathcal{C} is a quantum operation $\tilde{\mathcal{C}}$ such that:

- On regular states: $\tilde{\mathcal{C}}(\rho) = \mathcal{C}(\rho)$.
- On the vacuum state: $\tilde{\mathcal{C}}(|\text{vac}\rangle\langle\text{vac}|) = |\text{vac}\rangle\langle\text{vac}|$.

Hlér Kristjánsson et al. "Resource theories of communication". In: *New Journal of Physics* 22.7 (2020), p. 073014

Solution: We interpret a statement S as a vacuum-extended operation. The domain is extended by a vector $|\text{vac}\rangle$ in a one-dimensional Hilbert space Vac .

Vacuum-extended operations

A vacuum extension of a quantum operation \mathcal{C} is a quantum operation $\tilde{\mathcal{C}}$ such that:

- On regular states: $\tilde{\mathcal{C}}(\rho) = \mathcal{C}(\rho)$.
- On the vacuum state: $\tilde{\mathcal{C}}(|\text{vac}\rangle\langle\text{vac}|) = |\text{vac}\rangle\langle\text{vac}|$.

$\tilde{\mathcal{C}}$ is characterized by a pair (\mathcal{C}, F) , where F is called a *transformation matrix*.

Solution: We interpret a statement S as a vacuum-extended operation. The domain is extended by a vector $|\text{vac}\rangle$ in a one-dimensional Hilbert space Vac .

Vacuum-extended operations

A vacuum extension of a quantum operation \mathcal{C} is a quantum operation $\tilde{\mathcal{C}}$ such that:

- On regular states: $\tilde{\mathcal{C}}(\rho) = \mathcal{C}(\rho)$.
- On the vacuum state: $\tilde{\mathcal{C}}(|\text{vac}\rangle\langle\text{vac}|) = |\text{vac}\rangle\langle\text{vac}|$.

$\tilde{\mathcal{C}}$ is characterized by a pair (\mathcal{C}, F) , where F is called a *transformation matrix*.

QCASE $((\mathcal{C}, F), (\mathcal{D}, G))$ is well defined.

$$\text{QCASE}((\mathcal{C}, F), (\mathcal{D}, G)) = \left(\begin{bmatrix} \rho_1 & \rho_2 \\ \rho_3 & \rho_4 \end{bmatrix} \mapsto \begin{bmatrix} \mathcal{C}(\rho_1) & F\rho_2G^\dagger \\ G\rho_3F^\dagger & \mathcal{D}(\rho_4) \end{bmatrix}, \begin{bmatrix} F & 0 \\ 0 & G \end{bmatrix} \right)$$

Denotational semantics

Interpretation of programs

A statement S is interpreted as a vacuum-extended operation
 $\llbracket S \rrbracket = (\mathcal{C}, F)$.

Denotational semantics

Interpretation of programs

A statement S is interpreted as a vacuum-extended operation
 $\llbracket S \rrbracket = (\mathcal{C}, F)$.

Some statement interpretations:

$$\llbracket \mathbf{new} \; \mathbf{qbit} \; q \rrbracket := (|0\rangle\langle 0|_q \otimes -, |0\rangle_q \otimes I_\Gamma)$$

$$\llbracket q* = U \rrbracket := (U_q(\cdot)U_q^\dagger, U_q)$$

$$\llbracket \mathbf{qcase} \; q \; (0 \rightarrow P, 1 \rightarrow Q) \rrbracket := \mathbf{QCASE}_q[\llbracket P \rrbracket, \llbracket Q \rrbracket]$$

$$\llbracket \mathbf{while} \; q \; \mathbf{do} \; S \rrbracket := \mathbf{lfp}(\mathcal{F}_q^S)$$

where $\mathcal{F}_q^S(\mathcal{C}, F) := \mathbf{meas}_q[(\mathcal{I}, I), (\mathcal{C}, F) \circ \llbracket S \rrbracket]$.

Results

Universality and adequacy

Universality

For all vacuum extension (\mathcal{C}, F) , there exists a statement S whose interpretation satisfies $\llbracket S \rrbracket = (\mathcal{C}, F)$.

Universality and adequacy

Universality

For all vacuum extension (\mathcal{C}, F) , there exists a statement S whose interpretation satisfies $\llbracket S \rrbracket = (\mathcal{C}, F)$.

Adequacy

The operational and denotational semantics describe the same behavior of programs.

Full abstraction

S and S' are *observationally equivalent* ($S \approx S'$) if their behavior is indistinguishable in any context.

Full abstraction

$$S \approx S' \quad \text{iff} \quad \llbracket P \rrbracket = \llbracket Q \rrbracket$$

Full abstraction

S and S' are *observationally equivalent* ($S \approx S'$) if their behavior is indistinguishable in any context.

Full abstraction

$$S \approx S' \quad \text{iff} \quad \llbracket P \rrbracket = \llbracket Q \rrbracket$$

Exemples :

new qbit q; discard q \approx **skip**

Full abstraction

S and S' are *observationally equivalent* ($S \approx S'$) if their behavior is indistinguishable in any context.

Full abstraction

$$S \approx S' \quad \text{iff} \quad \llbracket P \rrbracket = \llbracket Q \rrbracket$$

Exemples :

new qbit q; discard q \approx **skip**

qcase q (0 \rightarrow S_0 , 1 \rightarrow S_1); qcase q (0 \rightarrow S'_0 , 1 \rightarrow S'_1)

Full abstraction

S and S' are *observationally equivalent* ($S \approx S'$) if their behavior is indistinguishable in any context.

Full abstraction

$$S \approx S' \quad \text{iff} \quad \llbracket P \rrbracket = \llbracket Q \rrbracket$$

Exemples :

$$\mathbf{new\ qbit\ q;\ discard\ q\ \approx\ skip}$$

$$\begin{aligned} \mathbf{qcase\ q\ (0 \rightarrow S_0,\ 1 \rightarrow S_1);\ qcase\ q\ (0 \rightarrow S'_0,\ 1 \rightarrow S'_1)} \\ \approx \\ \mathbf{qcase\ q\ (0 \rightarrow S_0; S'_0,\ 1 \rightarrow S_1; S'_1)} \end{aligned}$$

Full abstraction

S and S' are *observationally equivalent* ($S \approx S'$) if their behavior is indistinguishable in any context.

Full abstraction

$$S \approx S' \quad \text{iff} \quad \llbracket P \rrbracket = \llbracket Q \rrbracket$$

Exemples :

$$\mathbf{new\ qbit\ } q; \mathbf{discard\ } q \approx \mathbf{skip}$$

$$\begin{aligned} \mathbf{qcase\ } q \ (0 \rightarrow S_0, 1 \rightarrow S_1); \mathbf{qcase\ } q \ (0 \rightarrow S'_0, 1 \rightarrow S'_1) \\ \approx \\ \mathbf{qcase\ } q \ (0 \rightarrow S_0; S'_0, 1 \rightarrow S_1; S'_1) \end{aligned}$$

$$\mathbf{qcase\ } q \ (0 \rightarrow S, 1 \rightarrow S) \not\approx S \quad \text{in general}$$

Conclusion

Conclusion

We introduced a programming language combining **qcase**, measurement and recursion.

Conclusion

We introduced a programming language combining **qcase**, measurement and recursion.

The semantics of **qcase** requires some extra information:

- Default transitions in the operational semantics,
- Transformation matrices in the denotational semantics.

Conclusion

We introduced a programming language combining **qcase**, measurement and recursion.

The semantics of **qcase** requires some extra information:

- Default transitions in the operational semantics,
- Transformation matrices in the denotational semantics.

Results:

- ① Universality,
- ② Adequacy between the two semantics,
- ③ Full abstraction.

Conclusion

We introduced a programming language combining **qcase**, measurement and recursion.

The semantics of **qcase** requires some extra information:

- Default transitions in the operational semantics,
- Transformation matrices in the denotational semantics.

Results:

- ① Universality,
- ② Adequacy between the two semantics,
- ③ Full abstraction.

Questions?