Quantum alternation

Prakash Panangaden[1]

[1]School of Computer Science
McGill University and
School of Informatics
University of Edinburgh

18th International Conference on Reversible Computation
Torino, Italy
10th July 2026

Outline

Outline

Outline

Outline

Outline

Outline

Quantum programming languages

- Quantum Turing machines: very messy!

Quantum programming languages

- Quantum Turing machines: very messy!
- Circuits: low level, OK for algorithm design. Very flexible.

Quantum programming languages

- Quantum Turing machines: very messy!
- Circuits: low level, OK for algorithm design. Very flexible.
- Quantum λ-calculus: beautiful, but semantics is far from operational intuitions.

Quantum programming languages

- Quantum Turing machines: very messy!
- Circuits: low level, OK for algorithm design. Very flexible.
- Quantum λ-calculus: beautiful, but semantics is far from operational intuitions.
- Measurement calculus: low-level, close to implementation.

Quantum programming languages

- Quantum Turing machines: very messy!
- Circuits: low level, OK for algorithm design. Very flexible.
- Quantum λ-calculus: beautiful, but semantics is far from operational intuitions.
- Measurement calculus: low-level, close to implementation.
- Selinger's Quantum Programming Language: Quantum data and classical control.

Quantum programming languages

- Quantum Turing machines: very messy!
- Circuits: low level, OK for algorithm design. Very flexible.
- Quantum λ-calculus: beautiful, but semantics is far from operational intuitions.
- Measurement calculus: low-level, close to implementation.
- Selinger's Quantum Programming Language: Quantum data and classical control.
- There are many more: Quipper, Qwire, Q#,....

Example

Simple program

input b:**bit**;
input p, q:**qbit**;
$b :=$ measure p;
if b then $q := N(q)$ else $p := N(p)$;
output p, q

Example

Simple program

input b:**bit**;
input p, q:**qbit**;
$b :=$ measure p;
if b then $q := N(q)$ else $p := N(p)$;
output p, q

- N is the **NOT** operation on a qubit.

Example

Simple program

input b:**bit**;
input p, q:**qbit**;
$b :=$ measure p;
if b then $q := N(q)$ else $p := N(p)$;
output p, q

- N is the **NOT** operation on a qubit.
- **bit** and **qbit** separate datatypes.

Example

Simple program

input b:**bit**;
input p, q:**qbit**;
$b :=$ measure p;
if b then $q := N(q)$ else $p := N(p)$;
output p, q

- N is the **NOT** operation on a qubit.
- **bit** and **qbit** separate datatypes.
- The conditional is based on a classical boolean.

What about quantum alternation?

Simple program

input p, q:**qbit**;
$q = |0\rangle$;
$q := H(q)$;
if q then skip else $p := N(p)$;
output p, q

What about quantum alternation?

Simple program

input p, q:**qbit**;
$q = |0\rangle$;
$q := H(q)$;
if q then skip else $p := N(p)$;
output p, q

- Here H is the one-qubit Hadamard gate.

What about quantum alternation?

Simple program

input p, q:**qbit**;
$q = |0\rangle$;
$q := H(q)$;
if q then skip else $p := N(p)$;
output p, q

- Here H is the one-qubit Hadamard gate.
- q is in the state $\frac{1}{\sqrt{2}}[|0\rangle + |1\rangle]$ just before the conditional.

What about quantum alternation?

Simple program

input p, q:**qbit**;
$q = |0\rangle$;
$q := H(q)$;
if q then skip else $p := N(p)$;
output p, q

- Here H is the one-qubit Hadamard gate.
- q is in the state $\frac{1}{\sqrt{2}}[|0\rangle + |1\rangle]$ just before the conditional.
- The **if** is producing a controlled not; a crucial ingredient of reversible computing.

What about quantum alternation?

Simple program

input p, q:**qbit**;
$q = |0\rangle$;
$q := H(q)$;
if q then skip else $p := N(p)$;
output p, q

- Here H is the one-qubit Hadamard gate.
- q is in the state $\frac{1}{\sqrt{2}}[|0\rangle + |1\rangle]$ just before the conditional.
- The **if** is producing a controlled not; a crucial ingredient of reversible computing.
- Does this make sense?

What about quantum alternation?

Simple program

input p, q:**qbit**;
$q = |0\rangle$;
$q := H(q)$;
if q then skip else $p := N(p)$;
output p, q

- Here H is the one-qubit Hadamard gate.
- q is in the state $\frac{1}{\sqrt{2}}[|0\rangle + |1\rangle]$ just before the conditional.
- The **if** is producing a controlled not; a crucial ingredient of reversible computing.
- Does this make sense?
- This *example* makes sense but

What about quantum alternation?

Simple program

input p, q:**qbit**;
$q = |0\rangle$;
$q := H(q)$;
if q then skip else $p := N(p)$;
output p, q

- Here H is the one-qubit Hadamard gate.
- q is in the state $\frac{1}{\sqrt{2}}[|0\rangle + |1\rangle]$ just before the conditional.
- The **if** is producing a controlled not; a crucial ingredient of reversible computing.
- Does this make sense?
- This *example* makes sense but
- Quantum alternation is problematic in general.

Basics of bits and qubits

- A **bit** is a classical bit: it takes two values, $0, 1$.

Basics of bits and qubits

- A **bit** is a classical bit: it takes two values, $0, 1$.
- It can be read and written and copied freely.

Basics of bits and qubits

- A **bit** is a classical bit: it takes two values, $0, 1$.
- It can be read and written and copied freely.
- A **quantum bit** is a two-dimensional system, mathematically $\mathbb{C}^2$.

Basics of bits and qubits

- A **bit** is a classical bit: it takes two values, $0, 1$.
- It can be read and written and copied freely.
- A **quantum bit** is a two-dimensional system, mathematically $\mathbb{C}^2$.
- We write $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$.

Basics of bits and qubits

- A **bit** is a classical bit: it takes two values, $0, 1$.
- It can be read and written and copied freely.
- A **quantum bit** is a two-dimensional system, mathematically $\mathbb{C}^2$.
- We write $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$.
- We assume as understood the notions of Hilbert space, bra and ket notation, pure and mixed states, density matrices, unitaries, Hermitian operators and projection operators.

Basics of bits and qubits

- A **bit** is a classical bit: it takes two values, $0, 1$.
- It can be read and written and copied freely.
- A **quantum bit** is a two-dimensional system, mathematically $\mathbb{C}^2$.
- We write $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$.
- We assume as understood the notions of Hilbert space, bra and ket notation, pure and mixed states, density matrices, unitaries, Hermitian operators and projection operators.
- Any time we say "operator" we mean "bounded linear (= continuous) operator."

Positive operators

- Let $\mathcal{H}$ be a Hilbert space. An operator $A : \mathcal{H} \to \mathcal{H}$ is **positive** if for all $x \in \mathcal{H}$ we have $(x, Ax) \geq 0$.

Positive operators

- Let $\mathcal{H}$ be a Hilbert space. An operator $A : \mathcal{H} \to \mathcal{H}$ is **positive** if for all $x \in \mathcal{H}$ we have $(x, Ax) \geq 0$.
- The positive operators are automatically Hermitian and define a partial order.

Positive operators

- Let $\mathcal{H}$ be a Hilbert space. An operator $A : \mathcal{H} \to \mathcal{H}$ is **positive** if for all $x \in \mathcal{H}$ we have $(x, Ax) \geq 0$.
- The positive operators are automatically Hermitian and define a partial order.
- Density matrices are positive operators with trace ≤ 1.

Positive operators

- Let $\mathcal{H}$ be a Hilbert space. An operator $A : \mathcal{H} \to \mathcal{H}$ is **positive** if for all $x \in \mathcal{H}$ we have $(x, Ax) \geq 0$.
- The positive operators are automatically Hermitian and define a partial order.
- Density matrices are positive operators with trace ≤ 1.
- Thus, we have a natural order structure on density matrices.

Positive operators

- Let $\mathcal{H}$ be a Hilbert space. An operator $A : \mathcal{H} \to \mathcal{H}$ is **positive** if for all $x \in \mathcal{H}$ we have $(x, Ax) \geq 0$.
- The positive operators are automatically Hermitian and define a partial order.
- Density matrices are positive operators with trace ≤ 1.
- Thus, we have a natural order structure on density matrices.
- We write $\mathcal{B}(\mathcal{H})$ for the bounded linear operators on $\mathcal{H}$.

Positive operators

- Let $\mathcal{H}$ be a Hilbert space. An operator $A : \mathcal{H} \to \mathcal{H}$ is **positive** if for all $x \in \mathcal{H}$ we have $(x, Ax) \geq 0$.
- The positive operators are automatically Hermitian and define a partial order.
- Density matrices are positive operators with trace ≤ 1.
- Thus, we have a natural order structure on density matrices.
- We write $\mathcal{B}(\mathcal{H})$ for the bounded linear operators on $\mathcal{H}$.
- A **positive map** is a map from $\mathcal{B}(\mathcal{H})$ to itself such that it takes positive operators to positive operators.

Completely positive maps

- An operator representing a physical transformation has to be positive, because it must take density matrices to density matrices.

Completely positive maps

- An operator representing a physical transformation has to be positive, because it must take density matrices to density matrices.
- It should also be trace non-increasing (trace preserving if we want normalized density matrices).

Completely positive maps

- An operator representing a physical transformation has to be positive, because it must take density matrices to density matrices.
- It should also be trace non-increasing (trace preserving if we want normalized density matrices).
- Is this enough?

Completely positive maps

- An operator representing a physical transformation has to be positive, because it must take density matrices to density matrices.
- It should also be trace non-increasing (trace preserving if we want normalized density matrices).
- Is this enough?
- It is possible to have a positive map A from $\mathcal{B}(\mathcal{H}) \to \mathcal{B}(\mathcal{H})$, such that $A \otimes I_{\mathcal{K}} : \mathcal{B}(\mathcal{H} \otimes \mathcal{K}) \to \mathcal{B}(\mathcal{H} \otimes \mathcal{K})$ is not positive.

Completely positive maps

- An operator representing a physical transformation has to be positive, because it must take density matrices to density matrices.
- It should also be trace non-increasing (trace preserving if we want normalized density matrices).
- Is this enough?
- It is possible to have a positive map A from $\mathcal{B}(\mathcal{H}) \to \mathcal{B}(\mathcal{H})$, such that $A \otimes I_{\mathcal{K}} : \mathcal{B}(\mathcal{H} \otimes \mathcal{K}) \to \mathcal{B}(\mathcal{H} \otimes \mathcal{K})$ is not positive.
- This is unphysical.

Completely positive maps

- An operator representing a physical transformation has to be positive, because it must take density matrices to density matrices.
- It should also be trace non-increasing (trace preserving if we want normalized density matrices).
- Is this enough?
- It is possible to have a positive map A from $\mathcal{B}(\mathcal{H}) \to \mathcal{B}(\mathcal{H})$, such that $A \otimes I_{\mathcal{K}} : \mathcal{B}(\mathcal{H} \otimes \mathcal{K}) \to \mathcal{B}(\mathcal{H} \otimes \mathcal{K})$ is not positive.
- This is unphysical.
- A positive map such that its tensor product with any identity map is positive is called **completely positive**.

Completely positive maps

- An operator representing a physical transformation has to be positive, because it must take density matrices to density matrices.
- It should also be trace non-increasing (trace preserving if we want normalized density matrices).
- Is this enough?
- It is possible to have a positive map A from $\mathcal{B}(\mathcal{H}) \to \mathcal{B}(\mathcal{H})$, such that $A \otimes I_{\mathcal{K}} : \mathcal{B}(\mathcal{H} \otimes \mathcal{K}) \to \mathcal{B}(\mathcal{H} \otimes \mathcal{K})$ is not positive.
- This is unphysical.
- A positive map such that its tensor product with any identity map is positive is called **completely positive**.
- Maps describing physical processes (e.g. channels) must be completely positive maps (cp maps).

Completely positive maps

- An operator representing a physical transformation has to be positive, because it must take density matrices to density matrices.
- It should also be trace non-increasing (trace preserving if we want normalized density matrices).
- Is this enough?
- It is possible to have a positive map A from $\mathcal{B}(\mathcal{H}) \to \mathcal{B}(\mathcal{H})$, such that $A \otimes I_{\mathcal{K}} : \mathcal{B}(\mathcal{H} \otimes \mathcal{K}) \to \mathcal{B}(\mathcal{H} \otimes \mathcal{K})$ is not positive.
- This is unphysical.
- A positive map such that its tensor product with any identity map is positive is called **completely positive**.
- Maps describing physical processes (e.g. channels) must be completely positive maps (cp maps).
- A **superoperator** is a cp map that is also trace non-increasing.

Three ways to understand CP maps

Let us consider maps on spaces of matrices. Suppose that ϕ is a CP map and A is a matrix:

Three ways to understand CP maps

Let us consider maps on spaces of matrices. Suppose that ϕ is a CP map and A is a matrix:

- Kraus: $\phi(A) = \sum_i K_i^\dagger A K_i$ where K_i are matrices satisfying $\sum_i K_i K_i^\dagger \leq I$.

Three ways to understand CP maps

Let us consider maps on spaces of matrices. Suppose that ϕ is a CP map and A is a matrix:

- Kraus: $\phi(A) = \sum_i K_i^\dagger A K_i$ where K_i are matrices satisfying $\sum_i K_i K_i^\dagger \leq I$.
- This decomposition is not unique. If ϕ is $M_n \to M_k$ then K_i are all $n \times k$ and there are fewer than $n \cdot k$ of them.

Three ways to understand CP maps

Let us consider maps on spaces of matrices. Suppose that ϕ is a CP map and A is a matrix:

- Kraus: $\phi(A) = \sum_i K_i^\dagger A K_i$ where K_i are matrices satisfying $\sum_i K_i K_i^\dagger \leq I$.
- This decomposition is not unique. If ϕ is $M_n \to M_k$ then K_i are all $n \times k$ and there are fewer than $n \cdot k$ of them.
- Choi: The action of $\phi \in CP(M_n, M_k)$ can be given explicitly as a matrix in M_{nk} depending on the particular Kraus decomposition.

Three ways to understand CP maps

Let us consider maps on spaces of matrices. Suppose that ϕ is a CP map and A is a matrix:

- Kraus: $\phi(A) = \sum_i K_i^\dagger A K_i$ where K_i are matrices satisfying $\sum_i K_i K_i^\dagger \leq I$.
- This decomposition is not unique. If ϕ is $M_n \to M_k$ then K_i are all $n \times k$ and there are fewer than $n \cdot k$ of them.
- Choi: The action of $\phi \in CP(M_n, M_k)$ can be given explicitly as a matrix in M_{nk} depending on the particular Kraus decomposition.
- Stinespring: For any completely positive map $\theta : \mathcal{A} \to \mathcal{B}(\mathcal{H})$ there is a triple $(\pi, V, \mathcal{K})$ where $\mathcal{K}$ is a Hilbert space, $\pi : \mathcal{A} \to \mathcal{B}(\mathcal{K})$ is a representation and $V : \mathcal{H} \to \mathcal{K}$ such that

$$\theta(a) = V^\dagger \pi(a) V.$$

Stinespring

- Any completely positive map can be realized as a “twisted” homomorphism.

Stinespring

- Any completely positive map can be realized as a “twisted” homomorphism.
- There is even a special minimal such Stinespring representation.

Stinespring

- Any completely positive map can be realized as a “twisted” homomorphism.
- There is even a special minimal such Stinespring representation.
- For quantum information theory this tells one that any completely positive map can be realized as a unitary on an expanded space: purification.

Stinespring

- Any completely positive map can be realized as a “twisted” homomorphism.
- There is even a special minimal such Stinespring representation.
- For quantum information theory this tells one that any completely positive map can be realized as a unitary on an expanded space: purification.
- If $\theta \in CP(M_n, M_k)$ then the minimal Stinespring representation is in M_m where $m \leq n^2 k$.

Stinespring to Kraus

- Let $\mathcal{H}$ and $\mathcal{K}$ be two finite-dimensional Hilbert spaces and $\mathcal{B}(\mathcal{H}), \mathcal{B}(\mathcal{K})$ the Banach algebras of bounded linear operators.

Stinespring to Kraus

- Let $\mathcal{H}$ and $\mathcal{K}$ be two finite-dimensional Hilbert spaces and $\mathcal{B}(\mathcal{H}), \mathcal{B}(\mathcal{K})$ the Banach algebras of bounded linear operators.
- Let $\mathcal{E} : \mathcal{B}(\mathcal{H}) \to \mathcal{B}(\mathcal{K})$ be a superoperator.

Stinespring to Kraus

- Let $\mathcal{H}$ and $\mathcal{K}$ be two finite-dimensional Hilbert spaces and $\mathcal{B}(\mathcal{H}), \mathcal{B}(\mathcal{K})$ the Banach algebras of bounded linear operators.
- Let $\mathcal{E} : \mathcal{B}(\mathcal{H}) \to \mathcal{B}(\mathcal{K})$ be a superoperator.
- By Stinespring, there exists an ancilla $\mathcal{A}$ and an operator $V : \mathcal{K} \to \mathcal{H} \otimes \mathcal{A}$ such that

$$\mathcal{E}(\rho) = V^*(\rho \otimes \mathbb{I}_{\mathcal{A}})V.$$

Stinespring to Kraus

- Let $\mathcal{H}$ and $\mathcal{K}$ be two finite-dimensional Hilbert spaces and $\mathcal{B}(\mathcal{H}), \mathcal{B}(\mathcal{K})$ the Banach algebras of bounded linear operators.
- Let $\mathcal{E} : \mathcal{B}(\mathcal{H}) \to \mathcal{B}(\mathcal{K})$ be a superoperator.
- By Stinespring, there exists an ancilla $\mathcal{A}$ and an operator $V : \mathcal{K} \to \mathcal{H} \otimes \mathcal{A}$ such that

$$\mathcal{E}(\rho) = V^*(\rho \otimes \mathbb{I}_{\mathcal{A}})V.$$

- Choose a basis $\{e_i\}_{i=1}^{k}$ for $\mathcal{A}$ and define $V_i : \mathcal{K} \to \mathcal{H}$ by

$$\forall \psi \in \mathcal{K},\ V\psi = \sum_{i=1}^{k} (V_i\psi) \otimes e_i.$$

Stinespring to Kraus

- Let $\mathcal{H}$ and $\mathcal{K}$ be two finite-dimensional Hilbert spaces and $\mathcal{B}(\mathcal{H}), \mathcal{B}(\mathcal{K})$ the Banach algebras of bounded linear operators.
- Let $\mathcal{E} : \mathcal{B}(\mathcal{H}) \to \mathcal{B}(\mathcal{K})$ be a superoperator.
- By Stinespring, there exists an ancilla $\mathcal{A}$ and an operator $V : \mathcal{K} \to \mathcal{H} \otimes \mathcal{A}$ such that

$$\mathcal{E}(\rho) = V^*(\rho \otimes \mathbb{I}_{\mathcal{A}})V.$$

- Choose a basis $\{e_i\}_{i=1}^k$for $\mathcal{A}$ and define $V_i : \mathcal{K} \to \mathcal{H}$ by

$$\forall \psi \in \mathcal{K},\ V\psi = \sum_{i=1}^{k} (V_i\psi) \otimes e_i.$$

- Easy to check $\mathcal{E}(\rho) = \sum_{i=1}^k V_i^* \rho V_i$.

Stinespring to Kraus

- Let $\mathcal{H}$ and $\mathcal{K}$ be two finite-dimensional Hilbert spaces and $\mathcal{B}(\mathcal{H}), \mathcal{B}(\mathcal{K})$ the Banach algebras of bounded linear operators.
- Let $\mathcal{E} : \mathcal{B}(\mathcal{H}) \to \mathcal{B}(\mathcal{K})$ be a superoperator.
- By Stinespring, there exists an ancilla $\mathcal{A}$ and an operator $V : \mathcal{K} \to \mathcal{H} \otimes \mathcal{A}$ such that

$$\mathcal{E}(\rho) = V^*(\rho \otimes \mathbb{I}_{\mathcal{A}})V.$$

- Choose a basis $\{e_i\}_{i=1}^{k}$for $\mathcal{A}$ and define $V_i : \mathcal{K} \to \mathcal{H}$ by

$$\forall \psi \in \mathcal{K},\ V\psi = \sum_{i=1}^{k} (V_i\psi) \otimes e_i.$$

- Easy to check $\mathcal{E}(\rho) = \sum_{i=1}^{k} V_i^* \rho V_i$.
- The V_i give a Kraus representation for $\mathcal{E}$.

Löwner order on density matrices

- Recall that the positive operators form a cone, hence define a partial order: the Löwner order.

Löwner order on density matrices

- Recall that the positive operators form a cone, hence define a partial order: the Löwner order.
- $A \sqsubseteq B$ if $B - A$ is positive.

Löwner order on density matrices

- Recall that the positive operators form a cone, hence define a partial order: the Löwner order.
- $A \sqsubseteq B$ if $B - A$ is positive.
- Recall density matrices are defined to have trace ≤ 1, so the zero matrix is the smallest element in this order.

Löwner order on density matrices

- Recall that the positive operators form a cone, hence define a partial order: the Löwner order.
- $A \sqsubseteq B$ if $B - A$ is positive.
- Recall density matrices are defined to have trace ≤ 1, so the zero matrix is the smallest element in this order.
- In this order, every increasing sequence has a least upper bound (lub). Such a structure is called a **directed-complete** partial order (dcpo).

Löwner order on density matrices

- Recall that the positive operators form a cone, hence define a partial order: the Löwner order.
- $A \sqsubseteq B$ if $B - A$ is positive.
- Recall density matrices are defined to have trace ≤ 1, so the zero matrix is the smallest element in this order.
- In this order, every increasing sequence has a least upper bound (lub). Such a structure is called a **directed-complete** partial order (dcpo).
- Note it is not a lattice.

Löwner order on density matrices

- Recall that the positive operators form a cone, hence define a partial order: the Löwner order.
- $A \sqsubseteq B$ if $B - A$ is positive.
- Recall density matrices are defined to have trace ≤ 1, so the zero matrix is the smallest element in this order.
- In this order, every increasing sequence has a least upper bound (lub). Such a structure is called a **directed-complete** partial order (dcpo).
- Note it is not a lattice.
- Least upper bounds of increasing sequences co-incide with topological limits in the euclidean topology.

Löwner order on density matrices

- Recall that the positive operators form a cone, hence define a partial order: the Löwner order.
- $A \sqsubseteq B$ if $B - A$ is positive.
- Recall density matrices are defined to have trace ≤ 1, so the zero matrix is the smallest element in this order.
- In this order, every increasing sequence has a least upper bound (lub). Such a structure is called a **directed-complete** partial order (dcpo).
- Note it is not a lattice.
- Least upper bounds of increasing sequences co-incide with topological limits in the euclidean topology.
- Any order preserving function on the operators will preserve lubs of increasing sequences if it is topologically continuous.

Löwner order on density matrices

- Recall that the positive operators form a cone, hence define a partial order: the Löwner order.
- $A \sqsubseteq B$ if $B - A$ is positive.
- Recall density matrices are defined to have trace ≤ 1, so the zero matrix is the smallest element in this order.
- In this order, every increasing sequence has a least upper bound (lub). Such a structure is called a **directed-complete** partial order (dcpo).
- Note it is not a lattice.
- Least upper bounds of increasing sequences co-incide with topological limits in the euclidean topology.
- Any order preserving function on the operators will preserve lubs of increasing sequences if it is topologically continuous.
- A function from a dcpo to another dcpo is called **Scott continuous** if it preserves lubs of increasing sequences.

Flowcharts

Quantum flowcharts are built out of:

1. new qbit $q := \mathbf{0}$ [allocate]

Flowcharts

Quantum flowcharts are built out of:

1. new qbit $q := \mathbf{0}$ [allocate]
2. discard q [discard]

Flowcharts

Quantum flowcharts are built out of:

1. new qbit $q := \mathbf{0}$ [allocate]
2. discard q [discard]
3. $\vec{q} * = U$ [apply unitary to $\vec{q}$]

Flowcharts

Quantum flowcharts are built out of:

1. new qbit $q := \mathbf{0}$ [allocate]
2. discard q [discard]
3. $\vec{q} * = U$ [apply unitary to $\vec{q}$]
4. measure q [measure and branch on the result]

Flowcharts

Quantum flowcharts are built out of:

1. new qbit $q := \mathbf{0}$ [allocate]
2. discard q [discard]
3. $\vec{q} * = U$ [apply unitary to $\vec{q}$]
4. measure q [measure and branch on the result]
5. merge [combine two possible paths]

Semantics of the flowcharts I

- new qbit $q := \mathbf{0}$; $\rho \mapsto \begin{pmatrix} \rho & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}$

Semantics of the flowcharts I

- new qbit $q := \mathbf{0}$; $\rho \mapsto \begin{pmatrix} \rho & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}$
- discard q; $\begin{pmatrix} \rho_{11} & \rho_{12} \\ \rho_{21} & \rho_{22} \end{pmatrix} \mapsto [\rho_{11} + \rho_{22}]$

Semantics of measurement

$$\begin{tikzcd}
 & \cdot \arrow[d, "\rho"] & \\
 & \text{measure } q \arrow[dl, "0"'] \arrow[dr, "1"] & \\
\rho_{11} \otimes \mathbf{0} & & \rho_{22} \otimes \mathbf{1}
\end{tikzcd}$$

Here $\rho = \begin{pmatrix} \rho_{11} & \rho_{12} \\ \rho_{21} & \rho_{22} \end{pmatrix}$

$\rho_{11} \otimes \mathbf{0} = \begin{pmatrix} \rho_{11} & 0 \\ 0 & 0 \end{pmatrix}$

$\rho_{22} \otimes \mathbf{1} = \begin{pmatrix} 0 & 0 \\ 0 & \rho_{22} \end{pmatrix}$

Iteration

- Loop in the flowchart.

Iteration

- Loop in the flowchart.
- When the loop is unwound one gets “formally” an infinite flowchart.

Iteration

- Loop in the flowchart.
- When the loop is unwound one gets “formally” an infinite flowchart.
- The meaning of this is given by an infinite sum.

Iteration

- Loop in the flowchart.
- When the loop is unwound one gets “formally” an infinite flowchart.
- The meaning of this is given by an infinite sum.
- This sum can be proven to converge yielding a density matrix with trace ≤ 1.

Recursion

- Part of the program can call itself.

Recursion

- Part of the program can call itself.
- The recursive call may allocate new qubits.

Recursion

- Part of the program can call itself.
- The recursive call may allocate new qubits.
- The recursion can be partially unwound.

Recursion

- Part of the program can call itself.
- The recursive call may allocate new qubits.
- The recursion can be partially unwound.
- The successive unwindings are given by $F(\mathbf{0}), F^2(\mathbf{0}), \ldots$

Recursion

- Part of the program can call itself.
- The recursive call may allocate new qubits.
- The recursion can be partially unwound.
- The successive unwindings are given by $F(\mathbf{0}), F^2(\mathbf{0}), \ldots$
- Each unwinding is greater than the previous in the Löwner order, because F is monotone.

Recursion

- Part of the program can call itself.
- The recursive call may allocate new qubits.
- The recursion can be partially unwound.
- The successive unwindings are given by $F(\mathbf{0}), F^2(\mathbf{0}), \ldots$
- Each unwinding is greater than the previous in the Löwner order, because F is monotone.
- The meaning is given by a least upper bound of the increasing sequence.

Recursion

- Part of the program can call itself.
- The recursive call may allocate new qubits.
- The recursion can be partially unwound.
- The successive unwindings are given by $F(\mathbf{0}), F^2(\mathbf{0}), \ldots$
- Each unwinding is greater than the previous in the Löwner order, because F is monotone.
- The meaning is given by a least upper bound of the increasing sequence.
- Because the density matrices form a dcpo we are sure that the lubs exist.

Recursion

- Part of the program can call itself.
- The recursive call may allocate new qubits.
- The recursion can be partially unwound.
- The successive unwindings are given by $F(\mathbf{0}), F^2(\mathbf{0}), \ldots$
- Each unwinding is greater than the previous in the Löwner order, because F is monotone.
- The meaning is given by a least upper bound of the increasing sequence.
- Because the density matrices form a dcpo we are sure that the lubs exist.
- Recursion can implement iteration but not the other way around.

What do we want?

- Suppose we have a qubit q and two superoperators $S, T : \mathcal{B}(\mathcal{H}) \to \mathcal{B}(\mathcal{K})$ then the quantum alternation $(qAlt)(q; S, T)$ should be a superoperator from $\mathcal{B}(\mathcal{Q} \otimes \mathcal{H}) \to \mathcal{B}(\mathcal{Q} \otimes \mathcal{K})$.

What do we want?

- Suppose we have a qubit q and two superoperators $S, T : \mathcal{B}(\mathcal{H}) \to \mathcal{B}(\mathcal{K})$ then the quantum alternation $(qAlt)(q; S, T)$ should be a superoperator from $\mathcal{B}(\mathcal{Q} \otimes \mathcal{H}) \to \mathcal{B}(\mathcal{Q} \otimes \mathcal{K})$.
- We want this to be compositional, so we can then use this new superoperator in any context without looking inside it.

What do we want?

- Suppose we have a qubit q and two superoperators $S, T : \mathcal{B}(\mathcal{H}) \to \mathcal{B}(\mathcal{K})$ then the quantum alternation $(qAlt)(q; S, T)$ should be a superoperator from $\mathcal{B}(\mathcal{Q} \otimes \mathcal{H}) \to \mathcal{B}(\mathcal{Q} \otimes \mathcal{K})$.
- We want this to be compositional, so we can then use this new superoperator in any context without looking inside it.
- We want it to only depend on the superoperator and not on how the superoperator is described, *e.g.* through a specific Kraus form.

What do we want?

- Suppose we have a qubit q and two superoperators $S, T : \mathcal{B}(\mathcal{H}) \to \mathcal{B}(\mathcal{K})$ then the quantum alternation $(qAlt)(q; S, T)$ should be a superoperator from $\mathcal{B}(\mathcal{Q} \otimes \mathcal{H}) \to \mathcal{B}(\mathcal{Q} \otimes \mathcal{K})$.
- We want this to be compositional, so we can then use this new superoperator in any context without looking inside it.
- We want it to only depend on the superoperator and not on how the superoperator is described, *e.g.* through a specific Kraus form.
- We want the operation to be monotone so we can use this inside recursions.

Can we really do all this?

- No!

Can we really do all this?

- No!
- It is not possible to make it compositional and stick with superoperators.

Can we really do all this?

- No!
- It is not possible to make it compositional and stick with superoperators.
- Can we define it in a monotone way?

Can we really do all this?

- No!
- It is not possible to make it compositional and stick with superoperators.
- Can we define it in a monotone way?
- I am *almost* sure this is impossible.

Basic scheme

- $\mathcal{H}$ Hilbert space with orthonormal basis $\{e_i\}_{i=1}^{n}$, $\mathcal{K}$ another Hilbert space.

Basic scheme

- $\mathcal{H}$ Hilbert space with orthonormal basis $\{e_i\}_{i=1}^{n}$, $\mathcal{K}$ another Hilbert space.
- Let Π_i be the projection onto the subspace spanned by e_i.

Basic scheme

- $\mathcal{H}$ Hilbert space with orthonormal basis $\{e_i\}_{i=1}^{n}$, $\mathcal{K}$ another Hilbert space.
- Let Π_i be the projection onto the subspace spanned by e_i.
- For each $i \in \{1, 2, \ldots, k-1, k\}$ we have a *unitary* U_i acting on $\mathcal{K}$.

Basic scheme

- $\mathcal{H}$ Hilbert space with orthonormal basis $\{e_i\}_{i=1}^n$, $\mathcal{K}$ another Hilbert space.
- Let Π_i be the projection onto the subspace spanned by e_i.
- For each $i \in \{1, 2, \ldots, k-1, k\}$ we have a *unitary* U_i acting on $\mathcal{K}$.
- The quantum alternation of the U_i *controlled by* a state in $\mathcal{H}$ is defined to be the following unitary:

Basic scheme

- $\mathcal{H}$ Hilbert space with orthonormal basis $\{e_i\}_{i=1}^n$, $\mathcal{K}$ another Hilbert space.
- Let Π_i be the projection onto the subspace spanned by e_i.
- For each $i \in \{1, 2, \ldots, k-1, k\}$ we have a *unitary* U_i acting on $\mathcal{K}$.
- The quantum alternation of the U_i *controlled by* a state in $\mathcal{H}$ is defined to be the following unitary:
- $$\sum_{i=1}^{k} \Pi_i U_i : \mathcal{H} \otimes \mathcal{K} \to \mathcal{H} \otimes \mathcal{K}.$$

Basic scheme

- $\mathcal{H}$ Hilbert space with orthonormal basis $\{e_i\}_{i=1}^n$, $\mathcal{K}$ another Hilbert space.
- Let Π_i be the projection onto the subspace spanned by e_i.
- For each $i \in \{1, 2, \ldots, k-1, k\}$ we have a *unitary* U_i acting on $\mathcal{K}$.
- The quantum alternation of the U_i *controlled by* a state in $\mathcal{H}$ is defined to be the following unitary:
- $$\sum_{i=1}^{k} \Pi_i U_i : \mathcal{H} \otimes \mathcal{K} \to \mathcal{H} \otimes \mathcal{K}.$$
- If $\mathcal{H}$ is a qubit then we have $(|\ 0\ \rangle\langle\ 0\ | \otimes U_0) + (|\ 1\ \rangle\langle\ 1\ | \otimes U_1)$.

Basic scheme

- $\mathcal{H}$ Hilbert space with orthonormal basis $\{e_i\}_{i=1}^n$, $\mathcal{K}$ another Hilbert space.
- Let Π_i be the projection onto the subspace spanned by e_i.
- For each $i \in \{1, 2, \ldots, k-1, k\}$ we have a *unitary* U_i acting on $\mathcal{K}$.
- The quantum alternation of the U_i *controlled by* a state in $\mathcal{H}$ is defined to be the following unitary:
- $$\sum_{i=1}^{k} \Pi_i U_i : \mathcal{H} \otimes \mathcal{K} \to \mathcal{H} \otimes \mathcal{K}.$$
- If $\mathcal{H}$ is a qubit then we have $(|\,0\,\rangle\langle\,0\,| \otimes U_0) + (|\,1\,\rangle\langle\,1\,| \otimes U_1)$.
- Action: $(\sum_i e_i \otimes \psi_i) \mapsto (\sum_i e_i \otimes U_i \psi_i)$.

Examples I

- Syntax: **if** q **then** U_0 **else** U_1.

Examples I

- Syntax: **if** q **then** U_0 **else** U_1.
- Controlled NOT: **if** q **then skip else** $q_1 * = N$.

Examples I

- Syntax: **if** q **then** U_0 **else** U_1.
- Controlled NOT: **if** q **then skip else** $q_1 * = N$.
- Controlled Hadamard: **if** q **then skip else** $q_1 * = H$.

Examples I

- Syntax: **if** q **then** U_0 **else** U_1.
- Controlled NOT: **if** q **then skip else** $q_1 * = N$.
- Controlled Hadamard: **if** q **then skip else** $q_1 * = H$.
- Controlled phase **if** q **then** U_0 **else** $q_1 * = e^{i\theta}$.

Examples I

- Syntax: **if** q **then** U_0 **else** U_1.
- Controlled NOT: **if** q **then skip else** $q_1 * = N$.
- Controlled Hadamard: **if** q **then skip else** $q_1 * = H$.
- Controlled phase **if** q **then** U_0 **else** $q_1 * = e^{i\theta}$.
- Toffoli gate uses nested if:
  if q_0 **then skip else if** q_1 **then skip else** $q_2 * = N$.

Examples I

- Syntax: $\mathbf{if}\ q\ \mathbf{then}\ U_0\ \mathbf{else}\ U_1$.
- Controlled NOT: $\mathbf{if}\ q\ \mathbf{then\ skip\ else}\ q_1 * = N$.
- Controlled Hadamard: $\mathbf{if}\ q\ \mathbf{then\ skip\ else}\ q_1 * = H$.
- Controlled phase $\mathbf{if}\ q\ \mathbf{then}\ U_0\ \mathbf{else}\ q_1 * = e^{i\theta}$.
- Toffoli gate uses nested if:
  $\mathbf{if}\ q_0\ \mathbf{then\ skip\ else\ if}\ q_1\ \mathbf{then\ skip\ else}\ q_2 * = N$.
- Very useful for describing algorithms especially if there are only unitaries.

Examples II: Deutsch's algorithm

- Given a function $f : \{0, 1\} \to \{0, 1\}$ we can determine if f is a constant function or not, $f(0) = f(1)$ or not using only one computation of f.

Examples II: Deutsch's algorithm

- Given a function $f : \{0,1\} \to \{0,1\}$ we can determine if f is a constant function or not, $f(0) = f(1)$ or not using only one computation of f.
- Use qubits $|0\rangle, |1\rangle$ and build quantum circuit to compute $f(0) \oplus f(1)$ using one call to f. Measure the output.

Examples II: Deutsch's algorithm

- Given a function $f : \{0,1\} \to \{0,1\}$ we can determine if f is a constant function or not, $f(0) = f(1)$ or not using only one computation of f.
- Use qubits $|0\rangle, |1\rangle$ and build quantum circuit to compute $f(0) \oplus f(1)$ using one call to f. Measure the output.
- Let $U_i, i = 1, 2$ be unitaries mapping $|0\rangle$ to $|f(i)\rangle$.

Examples II: Deutsch's algorithm

- Given a function $f : \{0,1\} \to \{0,1\}$ we can determine if f is a constant function or not, $f(0) = f(1)$ or not using only one computation of f.
- Use qubits $|0\rangle, |1\rangle$ and build quantum circuit to compute $f(0) \oplus f(1)$ using one call to f. Measure the output.
- Let $U_i, i = 1,2$ be unitaries mapping $|0\rangle$ to $|f(i)\rangle$.
- $$\begin{array}{l} \textbf{new qbit } x, y \\ x{*} = H \\ y{*} = N; H \\ \textbf{if } x \textbf{ then } y{*} = U_0 \textbf{ else } y{*} = U_1 \\ x{*} = H \end{array}$$

Examples II: Deutsch's algorithm

- Given a function $f : \{0,1\} \to \{0,1\}$ we can determine if f is a constant function or not, $f(0) = f(1)$ or not using only one computation of f.
- Use qubits $|0\rangle, |1\rangle$ and build quantum circuit to compute $f(0) \oplus f(1)$ using one call to f. Measure the output.
- Let $U_i, i = 1,2$ be unitaries mapping $|0\rangle$ to $|f(i)\rangle$.
-
$$
\begin{array}{l}
\textbf{new qbit } x, y \\
x* = H \\
y* = N; H \\
\textbf{if } x \textbf{ then } y* = U_0 \textbf{ else } y* = U_1 \\
x* = H
\end{array}
$$
- Can write quantum Fourier transform in a few lines.

Examples II: Deutsch's algorithm

- Given a function $f : \{0,1\} \to \{0,1\}$ we can determine if f is a constant function or not, $f(0) = f(1)$ or not using only one computation of f.
- Use qubits $|0\rangle, |1\rangle$ and build quantum circuit to compute $f(0) \oplus f(1)$ using one call to f. Measure the output.
- Let $U_i, i = 1, 2$ be unitaries mapping $|0\rangle$ to $|f(i)\rangle$.
- $$\begin{array}{l} \textbf{new qbit } x, y \\ x* = H \\ y* = N; H \\ \textbf{if } x \textbf{ then } y* = U_0 \textbf{ else } y* = U_1 \\ x* = H \end{array}$$
- Can write quantum Fourier transform in a few lines.
- Simple and intuitive, but

Examples II: Deutsch's algorithm

- Given a function $f : \{0,1\} \to \{0,1\}$ we can determine if f is a constant function or not, $f(0) = f(1)$ or not using only one computation of f.
- Use qubits $|0\rangle, |1\rangle$ and build quantum circuit to compute $f(0) \oplus f(1)$ using one call to f. Measure the output.
- Let $U_i, i = 1, 2$ be unitaries mapping $|0\rangle$ to $|f(i)\rangle$.
- $$\begin{array}{l} \textbf{new qbit } x, y \\ x* = H \\ y* = N; H \\ \textbf{if } x \textbf{ then } y* = U_0 \textbf{ else } y* = U_1 \\ x* = H \end{array}$$
- Can write quantum Fourier transform in a few lines.
- Simple and intuitive, but
- can we extend it to quantum operations that are not unitaries?

What is a Kraus form?

- A superoperator describes the most general physical transformation of a system.

What is a Kraus form?

- A superoperator describes the most general physical transformation of a system.
- According to Stinespring, every transformation can be regarded as a unitary acting on an enlarged space followed by a partial trace.

What is a Kraus form?

- A superoperator describes the most general physical transformation of a system.
- According to Stinespring, every transformation can be regarded as a unitary acting on an enlarged space followed by a partial trace.
- This extra space is the environment which interacts with the system.

What is a Kraus form?

- A superoperator describes the most general physical transformation of a system.
- According to Stinespring, every transformation can be regarded as a unitary acting on an enlarged space followed by a partial trace.
- This extra space is the environment which interacts with the system.
- A superoperator is always represented by a Kraus form, but this is not unique.

What is a Kraus form?

- A superoperator describes the most general physical transformation of a system.
- According to Stinespring, every transformation can be regarded as a unitary acting on an enlarged space followed by a partial trace.
- This extra space is the environment which interacts with the system.
- A superoperator is always represented by a Kraus form, but this is not unique.
- A particular Kraus form comes from a particular choice of basis of the environment, as we saw.

What is a Kraus form?

- A superoperator describes the most general physical transformation of a system.
- According to Stinespring, every transformation can be regarded as a unitary acting on an enlarged space followed by a partial trace.
- This extra space is the environment which interacts with the system.
- A superoperator is always represented by a Kraus form, but this is not unique.
- A particular Kraus form comes from a particular choice of basis of the environment, as we saw.
- A basis corresponds to a particular choice of measurement. Thus the particular Kraus representation is dictated by how the experimenter chooses to describe the environment.

Semantics in terms of Kraus forms

- Our position: Do not try to give semantics in terms of superoperators, give the semantics in terms of the Kraus forms.

Semantics in terms of Kraus forms

- Our position: Do not try to give semantics in terms of superoperators, give the semantics in terms of the Kraus forms.
- Basic idea: we can form quantum alternation of the Kraus operators just as we did for unitaries; details on the next slide.

Semantics in terms of Kraus forms

- Our position: Do not try to give semantics in terms of superoperators, give the semantics in terms of the Kraus forms.
- Basic idea: we can form quantum alternation of the Kraus operators just as we did for unitaries; details on the next slide.
- Idea (Mingsheng Ying): Define quantum alternation by using *all possible* Kraus forms for a superoperator and define the meaning of quantum alternation to be the set of all possible combinations of quantum alternations of Kraus forms.

Semantics in terms of Kraus forms

- Our position: Do not try to give semantics in terms of superoperators, give the semantics in terms of the Kraus forms.
- Basic idea: we can form quantum alternation of the Kraus operators just as we did for unitaries; details on the next slide.
- Idea (Mingsheng Ying): Define quantum alternation by using *all possible* Kraus forms for a superoperator and define the meaning of quantum alternation to be the set of all possible combinations of quantum alternations of Kraus forms.
- Not compositional, already noted by M. Ying.

Semantics in terms of Kraus forms

- Our position: Do not try to give semantics in terms of superoperators, give the semantics in terms of the Kraus forms.
- Basic idea: we can form quantum alternation of the Kraus operators just as we did for unitaries; details on the next slide.
- Idea (Mingsheng Ying): Define quantum alternation by using *all possible* Kraus forms for a superoperator and define the meaning of quantum alternation to be the set of all possible combinations of quantum alternations of Kraus forms.
- Not compositional, already noted by M. Ying.
- Our claim: No compositional semantics in terms of superoperators is possible.

Semantics in terms of Kraus forms

- Our position: Do not try to give semantics in terms of superoperators, give the semantics in terms of the Kraus forms.
- Basic idea: we can form quantum alternation of the Kraus operators just as we did for unitaries; details on the next slide.
- Idea (Mingsheng Ying): Define quantum alternation by using *all possible* Kraus forms for a superoperator and define the meaning of quantum alternation to be the set of all possible combinations of quantum alternations of Kraus forms.
- Not compositional, already noted by M. Ying.
- Our claim: No compositional semantics in terms of superoperators is possible.
- We give compositional semantics but in terms of specific choices of Kraus operators, we do not try to give compositional superoperator semantics.

Quantum alternation of unitaries

Given unitary operators U, V on $\mathcal{H}$ and a qubit q (space $\mathcal{Q}$) we define

$$|\,0\,\rangle\langle\,0\,|\otimes U + |\,1\,\rangle\langle\,1\,|\otimes V = \begin{pmatrix} U & 0 \\ 0 & V \end{pmatrix}$$

as the quantum alternation of U and V.

Alternation of Kraus forms

- Given superoperators $\mathcal{E}, \mathcal{F}$ with Kraus forms

Alternation of Kraus forms

- Given superoperators $\mathcal{E}, \mathcal{F}$ with Kraus forms
- $\mathcal{E}\rho = \sum_{i=1}^{m} E_i^* \rho E_i$ and $\mathcal{F}\rho = \sum_{j=1}^{n} F_j^* \rho F_j$,

Alternation of Kraus forms

- Given superoperators $\mathcal{E}, \mathcal{F}$ with Kraus forms
- $\mathcal{E}\rho = \sum_{i=1}^{m} E_i^* \rho E_i$ and $\mathcal{F}\rho = \sum_{j=1}^{n} F_j^* \rho F_j$,
- we define a family of operators $K_{i,j}$ by

$$K_{i,j} = | 0 \rangle\langle 0 | \otimes (\frac{1}{\sqrt{m}} E_i) + | 1 \rangle\langle 1 | \otimes (\frac{1}{\sqrt{n}}) F_j) = \begin{pmatrix} \frac{1}{\sqrt{m}} E_i & 0 \\ 0 & \frac{1}{\sqrt{n}} F_j \end{pmatrix}.$$

Alternation of Kraus forms

- Given superoperators $\mathcal{E}, \mathcal{F}$ with Kraus forms
- $\mathcal{E}\rho = \sum_{i=1}^{m} E_i^* \rho E_i$ and $\mathcal{F}\rho = \sum_{j=1}^{n} F_j^* \rho F_j$,
- we define a family of operators $K_{i,j}$ by

$$K_{i,j} = \mid 0 \rangle\langle 0 \mid \otimes (\frac{1}{\sqrt{m}} E_i) + \mid 1 \rangle\langle 1 \mid \otimes (\frac{1}{\sqrt{n}}) F_j) = \begin{pmatrix} \frac{1}{\sqrt{m}} E_i & 0 \\ 0 & \frac{1}{\sqrt{n}} F_j \end{pmatrix}.$$

- This defines a superoperator

$$\mathcal{S}(\rho) = \sum_{i,j} K_{i,j}^* \rho K_{i,j}.$$

What Stinespring says

If one looks at the Stinespring dilation corresponding to the above construction we see that the ancilla spaces (environments) of the two Kraus forms are tensored together.

Kraus semantics

- We think of a superoperator as being given by a specific Kraus form.

Kraus semantics

- We think of a superoperator as being given by a specific Kraus form.
- We write the composition of Kraus forms as $S \bullet T$ where S and T are specific Kraus forms for the superoperators.

Kraus semantics

- We think of a superoperator as being given by a specific Kraus form.
- We write the composition of Kraus forms as $S \bullet T$ where S and T are specific Kraus forms for the superoperators.
- We interpret commands in the quantum programming language as specific Kraus forms. So we can think of a superoperator as a set of Kraus operators.

Kraus semantics

- We think of a superoperator as being given by a specific Kraus form.
- We write the composition of Kraus forms as $S \bullet T$ where S and T are specific Kraus forms for the superoperators.
- We interpret commands in the quantum programming language as specific Kraus forms. So we can think of a superoperator as a set of Kraus operators.
- The meaning of a construct will be given by a set of Kraus operators.

Kraus semantics

- We think of a superoperator as being given by a specific Kraus form.
- We write the composition of Kraus forms as $S \bullet T$ where S and T are specific Kraus forms for the superoperators.
- We interpret commands in the quantum programming language as specific Kraus forms. So we can think of a superoperator as a set of Kraus operators.
- The meaning of a construct will be given by a set of Kraus operators.
- Sequential composition

$$[\![P;Q]\!] = [\![Q]\!] \circ [\![P]\!] = \{E_i \circ F_j \mid E_i \in [\![P]\!], F_j \in [\![Q]\!]\}.$$

Kraus semantics

- We think of a superoperator as being given by a specific Kraus form.
- We write the composition of Kraus forms as $S \bullet T$ where S and T are specific Kraus forms for the superoperators.
- We interpret commands in the quantum programming language as specific Kraus forms. So we can think of a superoperator as a set of Kraus operators.
- The meaning of a construct will be given by a set of Kraus operators.
- Sequential composition

$$[\![P;Q]\!] = [\![Q]\!] \circ [\![P]\!] = \{E_i \circ F_j \mid E_i \in [\![P]\!], F_j \in [\![Q]\!]\}.$$

- Applying a unitary

$$[\![q* = U]\!] = \{U\}.$$

More semantics

- Measure q, this has type $\tau \to \tau \oplus \tau$

$$[\![\textbf{measure } q]\!] = \{\mathrm{in}_0 \circ \Pi_0, \mathrm{in}_1 \circ \Pi_1\}.$$

More semantics

- Measure q, this has type $\tau \to \tau \oplus \tau$

$$[\![\textbf{measure } q]\!] = \{\mathrm{in}_0 \circ \Pi_0, \mathrm{in}_1 \circ \Pi_1\}.$$

- Quantum alternation $[\![\textbf{if } q \textbf{ then } P \textbf{ else } Q]\!] =$

$$[\![P]\!] \bullet [\![Q]\!].$$

More semantics

- Measure q, this has type $\tau \to \tau \oplus \tau$

$$[\![\textbf{measure } q]\!] = \{\mathrm{in}_0 \circ \Pi_0, \mathrm{in}_1 \circ \Pi_1\}.$$

- Quantum alternation $[\![\textbf{if } q \textbf{ then } P \textbf{ else } Q]\!] =$

$$[\![P]\!] \bullet [\![Q]\!].$$

- We do not give semantics for loops and conditionals.

Quantum alternation cannot be compositional

- More precisely: If the semantics is based on superoperators it cannot be compositional.

Quantum alternation cannot be compositional

- More precisely: If the semantics is based on superoperators it cannot be compositional.
- consider $P \equiv e^{i\theta} I$ and I, as superoperators these are identical.

Quantum alternation cannot be compositional

- More precisely: If the semantics is based on superoperators it cannot be compositional.
- consider $P \equiv e^{i\theta} I$ and I, as superoperators these are identical.
- But **if** q **then** I **else** P is definitely not the same as **if** q **then** I **else** I; the latter is clearly the same as I and the first is the controlled-phase gate.

Quantum alternation cannot be compositional

- More precisely: If the semantics is based on superoperators it cannot be compositional.
- consider $P \equiv e^{i\theta} I$ and I, as superoperators these are identical.
- But **if** q **then** I **else** P is definitely not the same as **if** q **then** I **else** I; the latter is clearly the same as I and the first is the controlled-phase gate.
- This example arose from discussions with Mingsheng Ying and Yuan Feng at UTS Sydney based on an example due to Nengkun Yu.

Quantum alternation cannot be compositional

- More precisely: If the semantics is based on superoperators it cannot be compositional.
- consider $P \equiv e^{i\theta} I$ and I, as superoperators these are identical.
- But $\mathbf{if}\ q\ \mathbf{then}\ I\ \mathbf{else}\ P$ is definitely not the same as $\mathbf{if}\ q\ \mathbf{then}\ I\ \mathbf{else}\ I$; the latter is clearly the same as I and the first is the controlled-phase gate.
- This example arose from discussions with Mingsheng Ying and Yuan Feng at UTS Sydney based on an example due to Nengkun Yu.
- One can think of quantum alternation as an algorithmic notation, it is not clear what it means *physically*.

Non-monotonicity

Theorem

Quantum control even with just unitary operators, is not monotone with respect to the Löwner order.

Non-monotonicity

Theorem

Quantum control even with just unitary operators, is not monotone with respect to the Löwner order.

Non-monotonicity

Theorem

Quantum control even with just unitary operators, is not monotone with respect to the Löwner order.

- Let U, V be one-qubit unitaries and $\lambda, \mu \in [0, 1]$.

Non-monotonicity

Theorem

Quantum control even with just unitary operators, is not monotone with respect to the Löwner order.

- Let U, V be one-qubit unitaries and $\lambda, \mu \in [0, 1]$.
- Let $S(\rho) = U\rho U^\dagger$, $T(\rho) = V\rho V^\dagger$ be associated superoperators.

Non-monotonicity

Theorem

Quantum control even with just unitary operators, is not monotone with respect to the Löwner order.

- Let U, V be one-qubit unitaries and $\lambda, \mu \in [0, 1]$.
- Let $S(\rho) = U\rho U^\dagger$, $T(\rho) = V\rho V^\dagger$ be associated superoperators.
- We have $\lambda^2 S \leq S$ and $\mu^2 T \leq T$ in the Löwner order.

Non-monotonicity

Theorem

Quantum control even with just unitary operators, is not monotone with respect to the Löwner order.

- Let U, V be one-qubit unitaries and $\lambda, \mu \in [0, 1]$.
- Let $S(\rho) = U\rho U^\dagger$, $T(\rho) = V\rho V^\dagger$ be associated superoperators.
- We have $\lambda^2 S \leq S$ and $\mu^2 T \leq T$ in the Löwner order.
- Define $R(\sigma) = W\sigma W^\dagger$ where

$$W = \begin{pmatrix} U & 0 \\ 0 & V \end{pmatrix}$$

Non-monotonicity

Theorem

Quantum control even with just unitary operators, is not monotone with respect to the Löwner order.

- Let U, V be one-qubit unitaries and $\lambda, \mu \in [0, 1]$.
- Let $S(\rho) = U\rho U^\dagger$, $T(\rho) = V\rho V^\dagger$ be associated superoperators.
- We have $\lambda^2 S \leq S$ and $\mu^2 T \leq T$ in the Löwner order.
- Define $R(\sigma) = W\sigma W^\dagger$ where

$$W = \begin{pmatrix} U & 0 \\ 0 & V \end{pmatrix}$$

- Define Define $R'(\sigma) = W'\sigma W'^\dagger$ where

$$W = \begin{pmatrix} \lambda U & 0 \\ 0 & \mu V \end{pmatrix}$$

Non-monotonicity

Theorem

Quantum control even with just unitary operators, is not monotone with respect to the Löwner order.

- Let U, V be one-qubit unitaries and $\lambda, \mu \in [0, 1]$.
- Let $S(\rho) = U\rho U^\dagger$, $T(\rho) = V\rho V^\dagger$ be associated superoperators.
- We have $\lambda^2 S \leq S$ and $\mu^2 T \leq T$ in the Löwner order.
- Define $R(\sigma) = W\sigma W^\dagger$ where

$$W = \begin{pmatrix} U & 0 \\ 0 & V \end{pmatrix}$$

- Define Define $R'(\sigma) = W'\sigma W'^\dagger$ where

$$W = \begin{pmatrix} \lambda U & 0 \\ 0 & \mu V \end{pmatrix}$$

- By explicit calculation we can show that $R' \not\leq R$.

Grattage-Altenkirch 2005

- Defined a language and type system for quantum alternation.

Grattage-Altenkirch 2005

- Defined a language and type system for quantum alternation.
- They defined "strict" maps; only these can be put in alternation.

Grattage-Altenkirch 2005

- Defined a language and type system for quantum alternation.
- They defined "strict" maps; only these can be put in alternation.
- Strict maps correspond to having singleton Kraus decompositions.

Grattage-Altenkirch 2005

- Defined a language and type system for quantum alternation.
- They defined “strict” maps; only these can be put in alternation.
- Strict maps correspond to having singleton Kraus decompositions.
- For example, one cannot nest quantum conditionals.

Grattage-Altenkirch 2005

- Defined a language and type system for quantum alternation.
- They defined “strict” maps; only these can be put in alternation.
- Strict maps correspond to having singleton Kraus decompositions.
- For example, one cannot nest quantum conditionals.
- No discussion of what quantum alternation means physically or mathematically.

Ying et. al. $\sim$ 2015

- They use sets of superoperators.

Ying et. al. $\sim$ 2015

- They use sets of superoperators.
- They look at all Kraus decompositions and then combine them in all possible ways.

Ying et. al. $\sim$ 2015

- They use sets of superoperators.
- They look at all Kraus decompositions and then combine them in all possible ways.
- Then they forget the decomposition to get a superoperator semantics.

Ying et. al. $\sim$ 2015

- They use sets of superoperators.
- They look at all Kraus decompositions and then combine them in all possible ways.
- Then they forget the decomposition to get a superoperator semantics.
- Not compositional.

Summary

- Quantum alternation is troublesome: non-compositional and non-monotone.

Summary

- Quantum alternation is troublesome: non-compositional and non-monotone.
- Is it a sensible thing to even consider? It came from programming languages without thinking about physics.

Summary

- Quantum alternation is troublesome: non-compositional and non-monotone.
- Is it a sensible thing to even consider? It came from programming languages without thinking about physics.
- One should look at real physical situations, e.g. Mach-Zehnder interferometers, quantum random walks or adiabatic quantum computation and extract a notion of quantum alternation.

Summary

- Quantum alternation is troublesome: non-compositional and non-monotone.
- Is it a sensible thing to even consider? It came from programming languages without thinking about physics.
- One should look at real physical situations, e.g. Mach-Zehnder interferometers, quantum random walks or adiabatic quantum computation and extract a notion of quantum alternation.
- Perhaps quantum alternation and unrestricted recursion is not allowed in nature!

Thank you!