Verification of Quantum Systems

Alex Della Schiava

Università degli Studi di Udine (DMIF) - V&V Techniques

March 14th, 2022

Error in Quantum Computing

- **Error** is a common matter of discussion in the field of quantum computing.
- Herein, the issue is tackled following the work done in:

  M. Lewis, S. Soudjani, P. Zuliani (2021), Formal Verification of Quantum Programs: Theory, Tools and Challenges

Error in Quantum Computing

- **Error** is a common matter of discussion in the field of quantum computing.
- Herein, the issue is tackled following the work done in:

  M. Lewis, S. Soudjani, P. Zuliani (2021), Formal Verification of Quantum Programs: Theory, Tools and Challenges
- Three main causes to quantum error can be identified:

Error in Quantum Computing

- **Error** is a common matter of discussion in the field of quantum computing.
- Herein, the issue is tackled following the work done in:

  M. Lewis, S. Soudjani, P. Zuliani (2021), Formal Verification of Quantum Programs: Theory, Tools and Challenges

- Three main causes to quantum error can be identified:
  1. **Randomness.** Due to the probabilistic nature of quantum mechanics.

Error in Quantum Computing

- **Error** is a common matter of discussion in the field of quantum computing.
- Herein, the issue is tackled following the work done in:

  M. Lewis, S. Soudjani, P. Zuliani (2021), Formal Verification of Quantum Programs: Theory, Tools and Challenges

- Three main causes to quantum error can be identified:
  1. **Randomness.** Due to the probabilistic nature of quantum mechanics.
  2. **Hardware.** It is hard to work on *closed quantum systems* avoiding external *interference.*

Error in Quantum Computing

- **Error** is a common matter of discussion in the field of quantum computing.
- Herein, the issue is tackled following the work done in:

  M. Lewis, S. Soudjani, P. Zuliani (2021), Formal Verification of Quantum Programs: Theory, Tools and Challenges

- Three main causes to quantum error can be identified:
  1. **Randomness.** Due to the probabilistic nature of quantum mechanics.
  2. **Hardware.** It is hard to work on *closed quantum systems* avoiding external *interference*.
  3. **Defective software.** Typical, well known issue concerning faulty implementations.

Quantum Computing meets Formal Verification

- Focus on **defective software**: common issue for both *classical* and *quantum* computing.

Quantum Computing meets Formal Verification

- Focus on **defective software**: common issue for both *classical* and *quantum* computing.
- **Classical case.** Plenty of successful solutions thanks to the work done in *formal verification*.

Quantum Computing meets Formal Verification

- Focus on **defective software**: common issue for both *classical* and *quantum* computing.
- **Classical case.** Plenty of successful solutions thanks to the work done in *formal verification*.
- **Quantum case.** Is it possible to somehow translate these results into the quantum context?

Quantum Computing meets Formal Verification

- Focus on **defective software**: common issue for both *classical* and *quantum* computing.
- **Classical case.** Plenty of successful solutions thanks to the work done in *formal verification*.
- **Quantum case.** Is it possible to somehow translate these results into the quantum context?
- The answer is **yes**. This seminar shall try to motivate this answer.

Outline

1. Background
2. Quantum **while**-programs
3. Quantum Deductive Verification
4. Model Checking on Quantum CTL
5. Extra: Case study on Deutsch's Algorithm

Classical Verification Techniques

The results here presented are heavily inspired by the following classical verification techniques:

- Model Checking for CTL
  - Automatic, exhaustive, provides counterexamples.
  - But... state explosion problem, no fairness conditions in CTL.
- Deductive Verification
  - Use of inference rules to prove validity of the desired properties with respect to a system/program.
  - Floyd-Hoare logic.
  - Weakest precondition.

Quantum Mechanics

Dirac notation:

- Writing $|\psi\rangle$ denotes a *ket*, that is, a column vector in some vector space (Hilbert space in our case).

Quantum Mechanics

Dirac notation:

- Writing $|\psi\rangle$ denotes a *ket*, that is, a column vector in some vector space (Hilbert space in our case).
- Writing $\langle\psi|$ denotes a *bra*, the row vector version of $|\psi\rangle$ where its coordinates are replaced with their respective complex conjugates.

Dirac notation:

- Writing $|\psi\rangle$ denotes a *ket*, that is, a column vector in some vector space (Hilbert space in our case).
- Writing $\langle\psi|$ denotes a *bra*, the row vector version of $|\psi\rangle$ where its coordinates are replaced with their respective complex conjugates.

Example

$$|\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \quad \langle\psi| = \begin{pmatrix} \alpha^* & \beta^* \end{pmatrix}$$

where $\alpha, \beta \in \mathbb{C}$.

Quantum Mechanics

Quantum mechanics can be formulated in terms of:

State vectors

$$|\psi\rangle$$

- Simple vectors.
- More intuitive.
- Example: $|\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$

Density operators (matrices)

$$\rho = |\psi\rangle \langle\psi|$$

- Positive operators.
- Trace: $\mathsf{tr}(\rho) = 1$.
- Can describe *mixed states*:

$$\rho = \sum_i p_i |\psi_i\rangle \langle\psi_i|$$

Quantum Mechanics

Postulate 1

A quantum system is fully described by a state vector in a given Hilbert space with norm 1.

Example: a qubit.

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

- Hilbert space: $\mathcal{H}^2$.
- Basis: $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$; $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$
- Norm: $\sqrt{\langle\psi|\psi\rangle} = \sqrt{|\alpha|^2 + |\beta|^2} = 1$

$$\rho = |\psi\rangle\,\langle\psi|$$

Quantum Mechanics

Postulate 1

A quantum system is fully described by a state vector in a given Hilbert space with norm 1.

Example: a qubit.

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

- Hilbert space: $\mathcal{H}^2$.
- Basis: $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$; $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$
- Norm: $\sqrt{\langle\psi|\psi\rangle} = \sqrt{|\alpha|^2 + |\beta|^2} = 1$

$$\rho = |\psi\rangle\,\langle\psi|$$

- Matrix representation:

$$\rho = \begin{pmatrix} |\alpha|^2 & \alpha \cdot \beta^* \\ \alpha^* \cdot \beta & |\beta|^2 \end{pmatrix}$$

- Trace: $|\alpha|^2 + |\beta|^2 = 1$.

Quantum Mechanics

Postulate 2

The evolution of closed quantum systems is described by means of *unitary operators*.

- $U^\dagger$ is the complex conjugate transposed (*adjoint*) of U;
- Unitary operator U: $UU^\dagger = I$.

Quantum Mechanics

Postulate 2

The evolution of closed quantum systems is described by means of *unitary operators*.

- $U^\dagger$ is the complex conjugate transposed (*adjoint*) of U;
- Unitary operator U: $UU^\dagger = I$.

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

- Resulting state:

$$|\psi'\rangle = U|\psi\rangle$$

$$\rho = |\psi\rangle\langle\psi|$$

- Resulting state:

$$\begin{aligned}\rho' &= U|\psi\rangle\langle\psi|U^\dagger \\ &= U\rho U^\dagger\end{aligned}$$

Quantum Mechanics

Postulate 2

The evolution of closed quantum systems is described by means of *unitary operators*.

- $U^\dagger$ is the complex conjugate transposed (*adjoint*) of U;
- Unitary operator U: $UU^\dagger = I$.

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$$

- Resulting state:

$$|\psi'\rangle = U |\psi\rangle$$

$$\rho = |\psi\rangle \langle\psi|$$

- Resulting state:

$$\begin{aligned} \rho' &= U |\psi\rangle \langle\psi| U^\dagger \\ &= U\rho U^\dagger \end{aligned}$$

Quantum evolution is **reversible**: just apply $U^\dagger$ to the resulting state.

Postulate 3

A quantum projective measurement M is described by a collection of projectors $\{P_m\}$. The subscripts m denote the possible outcomes of the measurement.

Postulate 3

A quantum projective measurement M is described by a collection of projectors $\{P_m\}$. The subscripts m denote the possible outcomes of the measurement.

- Probability of measuring m:

$$p(m) = \langle\psi| P_m |\psi\rangle$$

$$p(m) = \mathsf{tr}(P_m \rho)$$

- Post-measurement state $|\psi'\rangle$:

$$|\psi'\rangle = \frac{P_m |\psi\rangle}{\sqrt{p(m)}}$$

$$\rho' = \frac{P_m \rho P_m^\dagger}{p(m)}$$

Quantum Mechanics

Postulate 4

The state space of a *composite* quantum system is given by the tensor product $\otimes$ of the state spaces of its components.

Example

Consider a composite quantum system A with two qubits $|\psi\rangle, |\varphi\rangle \in \mathcal{H}^2$. The state space of A corresponds to:

$$\mathcal{H}^2 \otimes \mathcal{H}^2 = \mathcal{H}^4$$

- **Note:** one can still interact with a single component, with $U, I \in \mathcal{H}^2$:

$$U \otimes I$$

- U is applied to $|\psi\rangle$ while $|\varphi\rangle$ is unaltered.

Quantum Mechanics

Now that the machinery of quantum mechanics has been formalized…

…time to break it.

Now that the machinery of quantum mechanics has been formalized...

...time to break it.

Definition (Partial Density Operators)

Given a Hilbert space $\mathcal{H}$, a partial density operator $\rho \in \mathcal{D}^{-}(\mathcal{H})$ is a density operator such that $\mathsf{tr}(\rho) \leq 1$, where $\mathcal{D}^{-}(\mathcal{H})$ is the set of all partial density operators in $\mathcal{H}$.

Partial density operators describe **non-normalized states**.

- What are they good for? This will be clearer when talking about quantum programs semantics.

Quantum **while**-programs

- An imperative, deterministic quantum programming language.
- First introduced in:

M. Ying (2011), Floyd-Hoare Logic for Quantum Programs
Association for Computing Machinery, Sections 3,4,5.

Quantum **while**-programs

- An imperative, deterministic quantum programming language.
- First introduced in:
  - M. Ying (2011), Floyd-Hoare Logic for Quantum Programs
    Association for Computing Machinery, Sections 3,4,5.
- (1)*Quantum data,* (2)*quantum control*
  1. Manipulate quantum variables.
  2. The state of a program is a quantum state: computational paths support *quantum branching*.

Quantum **while**-programs

- **Variables** represent quantum systems.
- Can be restricted to two types (w.l.o.g.):
  - **Boolean.**
  - **Integer.**

  Their domains are defined in terms of Hilbert spaces:

$$\mathcal{H}_{\text{Boolean}} = \mathcal{H}^2; \quad \mathcal{H}_{\text{Integer}} = \mathcal{H}^\infty$$

Quantum **while**-programs

- **Variables** represent quantum systems.
- Can be restricted to two types (w.l.o.g.):
  - **Boolean.**
  - **Integer.**

  Their domains are defined in terms of Hilbert spaces:

$$\mathcal{H}_{\text{Boolean}} = \mathcal{H}^2; \quad \mathcal{H}_{\text{Integer}} = \mathcal{H}^\infty$$

- **Quantum registers.** Finite sequences of variables: $\bar{q} = q_1, \ldots, q_n$. With domain:

$$\mathcal{H}_{\bar{q}} = \bigotimes_{i=1}^{n} \mathcal{H}_{q_i}$$

Syntax

The syntax of quantum **while**-programs is the following:

$$
\begin{aligned}
S ::= {} & \texttt{skip} \,\|\, q := |0\rangle \;\|\, \bar{q} := U\bar{q} \,\|\, S_1; S_2 \;\| \\
::= {} & \texttt{measure}\, M[\bar{q}] : \bar{S} \,\|\, \texttt{while}\, M[\bar{q}] = 1 \,\texttt{do}\, S
\end{aligned}
$$

Syntax

The syntax of quantum **while**-programs is the following:

$$\begin{aligned} S &::= \texttt{skip} \,\|\, q := |0\rangle \,\|\, \bar{q} := U\bar{q} \,\|\, S_1; S_2 \,\| \\ &::= \texttt{measure}\, M[\bar{q}] : \bar{S} \,\|\, \texttt{while}\, M[\bar{q}] = 1\, \texttt{do}\, S \end{aligned}$$

Some important remarks:

- $q := |0\rangle$ represents initialization. Assignment is not possible due to the No-cloning theorem.

Syntax

The syntax of quantum **while**-programs is the following:

$$S ::= \texttt{skip} \parallel q := |0\rangle \parallel \bar{q} := U\bar{q} \parallel S_1; S_2 \parallel$$
$$::= \texttt{measure}\, M[\bar{q}] : \bar{S} \parallel \texttt{while}\, M[\bar{q}] = 1\, \texttt{do}\, S$$

Some important remarks:

- $q := |0\rangle$ represents initialization. Assignment is not possible due to the No-cloning theorem.
- `measure` and `while` work by means of measurement. Coherently with Postulate 3 the state of $\bar{q}$ is altered.

Operational semantics

- A **state** of a program S is represented by a partial density operator ρ in the Hilbert space:

$$\mathcal{H}_S = \bigotimes_{q \in \mathcal{V}(S)} \mathcal{H}_q$$

where $\mathcal{V}(S)$ is the set of variables of S.

Operational semantics

- A **state** of a program S is represented by a partial density operator ρ in the Hilbert space:

$$\mathcal{H}_S = \bigotimes_{q \in \mathcal{V}(S)} \mathcal{H}_q$$

  where $\mathcal{V}(S)$ is the set of variables of S.
- A **configuration** of a program is a pair $\langle S, \rho \rangle$, where:
  - S is the program still to be executed;
  - ρ is the current state.

Operational semantics

- A **state** of a program S is represented by a partial density operator ρ in the Hilbert space:

$$\mathcal{H}_S = \bigotimes_{q \in \mathcal{V}(S)} \mathcal{H}_q$$

  where $\mathcal{V}(S)$ is the set of variables of S.
- A **configuration** of a program is a pair $\langle S, \rho \rangle$, where:
  - S is the program still to be executed;
  - ρ is the current state.
- The operational semantics is defined by means of the *transition relation*:

$$\langle S, \rho \rangle \to \langle S', \rho' \rangle$$

  $\langle \downarrow, \rho \rangle$ denotes a terminating configuration.

Transition rules: the Loop case.

- $\texttt{while}\, M[\bar{q}] = 1\, \texttt{do}\, S$

$$\frac{}{\langle \texttt{while}, \rho\rangle \to \langle \downarrow, M_0 \rho M_0^\dagger\rangle} \quad (\textit{Loop 0})$$

$$\frac{}{\langle \texttt{while}, \rho\rangle \to \langle S; \texttt{while}, M_1 \rho M_1^\dagger\rangle} \quad (\textit{Loop 1})$$

Transition rules: the Loop case.

- $\mathtt{while}\, M[\bar{q}] = 1\, \mathtt{do}\, S$

$$\frac{}{\langle \mathtt{while}, \rho \rangle \to \langle \downarrow, M_0 \rho M_0^\dagger \rangle} \qquad (\textit{Loop 0})$$

$$\frac{}{\langle \mathtt{while}, \rho \rangle \to \langle S; \mathtt{while}, M_1 \rho M_1^\dagger \rangle} \qquad (\textit{Loop 1})$$

Why $M_0 \rho M_0^\dagger$? Postulate 3 says that the post-measurement state should be:

$$\rho_0' = \frac{M_0 \rho M_0^\dagger}{p(0)}, \text{ with } p(0) = \mathsf{tr}(M_0 \rho M_0 \dagger),$$

Transition rules: the Loop case.

- $\texttt{while}\, M[\bar{q}] = 1\, \texttt{do}\, S$

$$\frac{}{\langle \texttt{while}, \rho\rangle \to \langle \downarrow, M_0 \rho M_0^\dagger\rangle} \quad (\textit{Loop 0})$$

$$\frac{}{\langle \texttt{while}, \rho\rangle \to \langle S; \texttt{while}, M_1 \rho M_1^\dagger\rangle} \quad (\textit{Loop 1})$$

Why $M_0 \rho M_0^\dagger$? Postulate 3 says that the post-measurement state should be:

$$\rho_0' = \frac{M_0 \rho M_0^\dagger}{p(0)}, \text{ with } p(0) = \mathsf{tr}(M_0 \rho M_0 \dagger),$$

Idea. Use *partial density operators* instead.

- Introduce *non-determinism* to avoid probabilistic transition rules:

$$\frac{}{\langle \texttt{while}, \rho\rangle \xrightarrow{p_m} \langle S_m, \rho_m\rangle}$$

Denotational semantics

- Given a program S, its denotational semantics is defined by means of the function $[\![S]\!] : \mathcal{D}^-(\mathcal{H}_S) \to \mathcal{D}^-(\mathcal{H}_S)$:

$$[\![S]\!](\rho) = \sum \{|\rho' : \langle S, \rho\rangle \xrightarrow{*} \langle \downarrow, \rho'\rangle|\}$$

  where $\xrightarrow{*}$ stands for n steps for any n.
- The sum of all terminating states.
- **Observation:** what effect does it have on the trace?

Denotational semantics

- Given a program S, its denotational semantics is defined by means of the function $[\![S]\!] : \mathcal{D}^-(\mathcal{H}_S) \to \mathcal{D}^-(\mathcal{H}_S)$:

$$[\![S]\!](\rho) = \sum \{|\rho' : \langle S, \rho\rangle \xrightarrow{*} \langle \downarrow, \rho'\rangle|\}$$

  where $\xrightarrow{*}$ stands for n steps for any n.
- The sum of all terminating states.
- **Observation:** what effect does it have on the trace?

$$\mathsf{tr}([\![S]\!](\rho)) \leq \mathsf{tr}(\rho)$$

- More interestingly: $\mathsf{tr}([\![S]\!](\rho)) < \mathsf{tr}(\rho)$ *if and only if S diverges.*

Denotational semantics

- Given a program S, its denotational semantics is defined by means of the function $[\![S]\!] : \mathcal{D}^-(\mathcal{H}_S) \to \mathcal{D}^-(\mathcal{H}_S)$:

$$[\![S]\!](\rho) = \sum \{|\rho' : \langle S, \rho \rangle \xrightarrow{*} \langle \downarrow, \rho' \rangle|\}$$

  where $\xrightarrow{*}$ stands for n steps for any n.
- The sum of all terminating states.
- **Observation:** what effect does it have on the trace?

$$\mathsf{tr}([\![S]\!](\rho)) \leq \mathsf{tr}(\rho)$$

- More interestingly: $\mathsf{tr}([\![S]\!](\rho)) < \mathsf{tr}(\rho)$ *if and only if S diverges.*

$$\text{Probability of diverging} = \mathsf{tr}(\rho) - \mathsf{tr}([\![S]\!](\rho))$$

Quantum Deductive Verification

- This section refers to the work done in:

M. Ying (2011), Floyd-Hoare Logic for Quantum Programs
Association for Computing Machinery, Sections 6-9.

E. D'Hondt, P. Panangaden (2006), Quantum Weakest Preconditions
Mathematical Structures in Computer Science, 16(3), 429-451.

Quantum Deductive Verification

Definition (Quantum Predicate)

A quantum predicate M is a *projective measurement*. As such, it has a *spectral decomposition* of the form:

$$M = \sum_m m P_m$$

where the possible outcomes m are given by the *eigenvalues*. In the case of quantum predicates, eigenvalues are bounded by 1.

Quantum Deductive Verification

Definition (Quantum Predicate)

A quantum predicate M is a *projective measurement*. As such, it has a *spectral decomposition* of the form:

$$M = \sum_m m P_m$$

where the possible outcomes m are given by the *eigenvalues*. In the case of quantum predicates, eigenvalues are bounded by 1.

- **Idea:** define satisfiability as the *expectation value* $\mathsf{tr}(M\rho)$.
- That is, the probability of ρ satisfying M.

Definition (Quantum Predicate)

A quantum predicate M is a *projective measurement*. As such, it has a *spectral decomposition* of the form:

$$M = \sum_m m P_m$$

where the possible outcomes m are given by the *eigenvalues*. In the case of quantum predicates, eigenvalues are bounded by 1.

- **Idea:** define satisfiability as the *expectation value* $\mathsf{tr}(M\rho)$.
- That is, the probability of ρ satisfying M.
- Natural translation of the satisfies relation:

$$\rho \models_r M \iff \mathsf{tr}(M\rho) \geq r,$$

where $r \in [0, 1]$.

Quantum Floyd-Hoare Logic (QHL)

To do:

1. Define the notion of quantum correctness;
2. Define the set of axioms and inference rules for quantum **while**-programs.

Definition (Quantum Hoare Triple)

Given a quantum program S, a state ρ and two quantum predicates P, Q, the quantum Hoare triple $\{P\}S\{Q\}$ denotes that:

$$\mathsf{tr}(P\rho) \leq \mathsf{tr}(Q[\![S]\!](\rho))$$

Quantum Floyd-Hoare Logic (QHL)

To do:

1. Define the notion of quantum correctness;
2. Define the set of axioms and inference rules for quantum **while**-programs.

Definition (Quantum Hoare Triple)

Given a quantum program S, a state ρ and two quantum predicates P, Q, the quantum Hoare triple $\{P\}S\{Q\}$ denotes that:

$$\mathsf{tr}(P\rho) \leq \mathsf{tr}(Q[\![S]\!](\rho))$$

- This definition is perhaps clearer when translated into the *quantum satisfies* relation:

$$\rho \models_r P \Longrightarrow [\![S]\!](\rho) \models_r Q$$

$\forall r \in [0, 1]$.

Quantum Floyd-Hoare Logic (QHL)

Definition (Quantum Total Correctness)

The quantum Hoare triple $\{P\}S\{Q\}$ is valid in terms of total correctness (formally $\models tot\{P\}S\{Q\}$) if $\forall \rho \in \mathcal{D}^{-}(\mathcal{H}_S)$

$$\mathsf{tr}(P\rho) \leq \mathsf{tr}(Q[\![S]\!](\rho))$$

Quantum Floyd-Hoare Logic (QHL)

Definition (Quantum Total Correctness)

The quantum Hoare triple $\{P\}S\{Q\}$ is valid in terms of total correctness (formally $\models tot\{P\}S\{Q\}$) if $\forall \rho \in \mathcal{D}^-(\mathcal{H}_S)$

$$\mathsf{tr}(P\rho) \leq \mathsf{tr}(Q[\![S]\!](\rho))$$

Definition (Quantum Partial Correctness)

The quantum Hoare triple $\{P\}S\{Q\}$ is valid in terms of partial correctness (formally $\models par\{P\}S\{Q\}$) if $\forall \rho \in \mathcal{D}^-(\mathcal{H}_S)$

$$\mathsf{tr}(P\rho) \leq \mathsf{tr}(Q[\![S]\!](\rho)) + \big(\mathsf{tr}(\rho) - \mathsf{tr}([\![S]\!](\rho)\big)$$

Quantum Floyd-Hoare Logic (QHL)

Definition (Quantum Total Correctness)

The quantum Hoare triple $\{P\}S\{Q\}$ is valid in terms of total correctness (formally $\models tot\{P\}S\{Q\}$) if $\forall \rho \in \mathcal{D}^-(\mathcal{H}_S)$

$$\mathsf{tr}(P\rho) \leq \mathsf{tr}(Q[\![S]\!](\rho))$$

Definition (Quantum Partial Correctness)

The quantum Hoare triple $\{P\}S\{Q\}$ is valid in terms of partial correctness (formally $\models par\{P\}S\{Q\}$) if $\forall \rho \in \mathcal{D}^-(\mathcal{H}_S)$

$$\mathsf{tr}(P\rho) \leq \mathsf{tr}(Q[\![S]\!](\rho)) + \big(\mathsf{tr}(\rho) - \mathsf{tr}([\![S]\!](\rho)\big)$$

Remember that $\big(\mathsf{tr}(\rho) - \mathsf{tr}([\![S]\!](\rho)\big)$ represents the probability that S will not terminate.

Quantum Partial Correctness Proof System

- It is now possible to define an *axiomatic base* for quantum **while**-programs.

Quantum Partial Correctness Proof System

- It is now possible to define an *axiomatic base* for quantum **while**-programs.
- As in the classical case, two proof systems can be defined respectively for partial and total correctness.

Quantum Partial Correctness Proof System

- It is now possible to define an *axiomatic base* for quantum **while**-programs.
- As in the classical case, two proof systems can be defined respectively for partial and total correctness.
- **The Unitary transformation case:**

$$(Unitary) \quad \frac{}{\{U^\dagger P U\}\ \bar{q} := U\bar{q}\ \{P\}}$$

- Notice how, for any $\bar{q}$:

$$\mathsf{tr}(U^\dagger P U \bar{q}) = \mathsf{tr}(P U \bar{q} U^\dagger) = \mathsf{tr}(P[\![\bar{q} := U\bar{q}]\!])$$

Quantum Weakest Precondition

- What does one mean by **weakest**?
- A way to compare the **strength** of quantum predicates is required.

Quantum Weakest Precondition

- What does one mean by **weakest**?
- A way to compare the **strength** of quantum predicates is required.

Definition (Löwner partial order)

Given two predicates P, Q, the writing $P \sqsubseteq Q$ is used to denote that for any $\rho \in \mathcal{D}^-(\mathcal{H})$:

$$\mathsf{tr}(P\rho) \leq \mathsf{tr}(Q\rho)$$

- Translating into the r-satisfies relation:

$$\rho \models_r P \Longrightarrow \rho \models_r Q$$

Quantum Weakest Precondition

Definition (Quantum Weakest Precondition)

The weakest precondition of a predicate Q with respect to a program S is a quantum predicate $\mathsf{qwp}.S(Q) \in \mathcal{P}(\mathcal{H}_S)$ such that:

- $\models_{tot} \{\mathsf{qwp}.S(Q)\}S\{Q\}$
- For any $P \in \mathcal{P}(\mathcal{H}_S)$, $\models_{tot} \{P\}S\{Q\} \Rightarrow P \sqsubseteq \mathsf{qwp}.S(Q)$

- The function can be defined over the statements of quantum **while**-programs.
- Easily verify a quantum Hoare triple $\{P\}S\{Q\}$:

Quantum Weakest Precondition

Definition (Quantum Weakest Precondition)

The weakest precondition of a predicate Q with respect to a program S is a quantum predicate $\mathsf{qwp}.S(Q) \in \mathcal{P}(\mathcal{H}_S)$ such that:

- $\models_{tot} \{\mathsf{qwp}.S(Q)\}S\{Q\}$
- For any $P \in \mathcal{P}(\mathcal{H}_S)$, $\models_{tot} \{P\}S\{Q\} \Rightarrow P \sqsubseteq \mathsf{qwp}.S(Q)$

- The function can be defined over the statements of quantum **while**-programs.
- Easily verify a quantum Hoare triple $\{P\}S\{Q\}$:
  1. Compute $\mathsf{qwp}.S(Q)$;
  2. Check if $P \sqsubseteq \mathsf{qwp}.S(Q)$.

Model Checking on Quantum CTL

- On the lines of the work done in:

  P. Baltazar, R. Chadha, P. Mateus (2008), Quantum Computation Tree Logic - Model Checking and Complete Calculus
  International Journal of Quantum Information

- Quantum CTL (QCTL) is a temporal logic built on dEQPL (*decidable fragment of the Exogenous Quantum Propositional Logic*).
- Herein, a restricted version of dEQPL is presented:
  - Finite Hilbert spaces;
  - Closed formulae only.
- QCTL is obtained by simply enriching dEQPL with temporal modalities.

dEQPL

- **Idea:** Replace propositional letters with qubits.

dEQPL

- **Idea:** Replace propositional letters with qubits.
- Set of n qubits $\mathsf{qB} = \{q_1, \ldots, q_n\}$.
- A valuation over qB is a state $|\psi\rangle$ in the Hilbert space:

$$\bigotimes_{i=1}^{n} \mathcal{H}^2 = \mathcal{H}^{2^n} = \mathcal{H}_{\mathsf{qB}}$$

generated by the computational basis: $\{|v_i\rangle\}_{i=1}^{2^n}$.

dEQPL

- **Idea:** Replace propositional letters with qubits.
- Set of n qubits $\mathsf{qB} = \{q_1, \dots, q_n\}$.
- A valuation over qB is a state $|\psi\rangle$ in the Hilbert space:

$$\bigotimes_{i=1}^{n} \mathcal{H}^2 = \mathcal{H}^{2^n} = \mathcal{H}_{\mathsf{qB}}$$

  generated by the computational basis: $\{|v_i\rangle\}_{i=1}^{2^n}$.
- Each vector of the basis represents a possible valuation over qB:

$$|v_1\rangle = |0^n\rangle \,; |v_2\rangle = |0^{(n-1)}1\rangle \,; \dots$$

dEQPL

- **Idea:** Replace propositional letters with qubits.
- Set of n qubits $\mathsf{qB} = \{q_1, \ldots, q_n\}$.
- A valuation over qB is a state $|\psi\rangle$ in the Hilbert space:

$$\bigotimes_{i=1}^{n} \mathcal{H}^2 = \mathcal{H}^{2^n} = \mathcal{H}_{\mathsf{qB}}$$

  generated by the computational basis: $\{|v_i\rangle\}_{i=1}^{2^n}$.
- Each vector of the basis represents a possible valuation over qB:

$$|v_1\rangle = |0^n\rangle\,; |v_2\rangle = |0^{(n-1)}1\rangle\,; \ldots$$

- **Important:** it is possible to describe superpositions of valuations:

$$|\psi\rangle = \frac{1}{\sqrt{2}}\,|0^n\rangle + \frac{1}{\sqrt{2}}\,|1^n\rangle$$

dEQPL - Syntax

Syntax. Three syntactic categories:

- **Classical formulae** (where $q \in \mathsf{qB}$):

$$\alpha \coloneqq \perp \parallel q \parallel \alpha \Rightarrow \alpha$$

dEQPL - Syntax

Syntax. Three syntactic categories:

- **Classical formulae** (where $q \in \mathsf{qB}$):

$$\alpha := \bot \parallel q \parallel \alpha \Rightarrow \alpha$$

- **Term language** (where $m \in \mathbb{Z}$ and $A \subseteq \mathsf{qB}$):

$$t := m \parallel t + t \parallel t * t \parallel \mathsf{Re}(|\top\rangle_A) \parallel \mathsf{Im}(|\top\rangle_A) \parallel \int \alpha$$

dEQPL - Syntax

Syntax. Three syntactic categories:

- **Classical formulae** (where $q \in \mathsf{qB}$):

$$\alpha \coloneqq \bot \parallel q \parallel \alpha \Rightarrow \alpha$$

- **Term language** (where $m \in \mathbb{Z}$ and $A \subseteq \mathsf{qB}$):

$$t \coloneqq m \parallel t + t \parallel t * t \parallel \mathsf{Re}(|\top\rangle_A) \parallel \mathsf{Im}(|\top\rangle_A) \parallel \textstyle\int \alpha$$

- **Quantum formulae:**

$$\gamma \coloneqq t \leq t \parallel \perp\!\!\!\perp \parallel \gamma \sqsupset \gamma$$

dEQPL - Semantics

Semantics. Given a set of n qubits qB dEQPL formulae are interpreted over a state $|\psi\rangle \in \mathcal{H}_{\mathsf{qB}}$.

- **Terms denotations:**

$$\begin{aligned}
[\![\mathsf{Re}(|\top\rangle_A)]\!]_{|\psi\rangle} &= \mathsf{Re}(\langle v_A|\psi\rangle) && \text{(Real part)}\\
[\![\mathsf{Im}(|\top\rangle_A)]\!]_{|\psi\rangle} &= \mathsf{Im}(\langle v_A|\psi\rangle) && \text{(Imaginary part)}\\
[\![\textstyle\int \alpha]\!]_{|\psi\rangle} &= \mu_{|\psi\rangle}(\mathcal{E}(\alpha)) && \text{(Probability map)}
\end{aligned}$$

dEQPL - Semantics

Semantics. Given a set of n qubits qB dEQPL formulae are interpreted over a state $|\psi\rangle \in \mathcal{H}_{\mathsf{qB}}$.

- **Terms denotations:**

$$
\begin{aligned}
[\![\mathsf{Re}(|\top\rangle_A)]\!]_{|\psi\rangle} &= \mathsf{Re}(\langle v_A|\psi\rangle) && \text{(Real part)}\\
[\![\mathsf{Im}(|\top\rangle_A)]\!]_{|\psi\rangle} &= \mathsf{Im}(\langle v_A|\psi\rangle) && \text{(Imaginary part)}\\
[\![\textstyle\int \alpha]\!]_{|\psi\rangle} &= \mu_{|\psi\rangle}(\mathcal{E}(\alpha)) && \text{(Probability map)}
\end{aligned}
$$

- $\mathsf{Re}(\langle v_A|\psi\rangle)$ and $\mathsf{Im}(\langle v_A|\psi\rangle)$ denote the real and imaginary of the logical amplitude of $\langle v_A|\psi\rangle$.
- Any $A \subseteq \mathsf{qB}$ is mapped onto a unique valutation $|v_A\rangle$.
  - *e.g.* $A = \{q_1, q_3\}$ is mapped onto $|v_A\rangle = |1010\ldots0\rangle$.
- From $|\psi\rangle = \frac{1}{\sqrt{2}}|0^n\rangle + \frac{1}{\sqrt{2}}|1^n\rangle$ one gets $\langle 0^n|\psi\rangle = \frac{1}{\sqrt{2}}$.

- $\mu_{|\psi\rangle}(\mathcal{E}(\alpha))$ is the probability of state $|\psi\rangle$ satisfying α.

dEQPL - Semantics

- $\mu_{|\psi\rangle}(\mathcal{E}(\alpha))$ is the probability of state $|\psi\rangle$ satisfying α.

1. Compute the *extent* of α:

$$\mathcal{E}(\alpha) = \{v \in 2^{\mathsf{qB}} : v \Vdash_c \alpha\}$$

namely, the set of all valuations satisfying α.

- $\mu_{|\psi\rangle}(\mathcal{E}(\alpha))$ is the probability of state $|\psi\rangle$ satisfying α.

1. Compute the *extent* of α:

$$\mathcal{E}(\alpha) = \{v \in 2^{\mathsf{qB}} : v \Vdash_c \alpha\}$$

namely, the set of all valuations satisfying α.

2. Calculate the probability of $|\psi\rangle$ *collapsing* into any of the valuations in the extent:

$$\mu_{|\psi\rangle}(\mathcal{E}(\alpha)) = \sum_{v \in \mathcal{E}(\alpha)} \| \langle v|\psi\rangle \|^2$$

dEQPL - Semantics

Quantum formulae. Semantics are defined by means of the $\Vdash_d$ relation.

$$
\begin{aligned}
&|\psi\rangle \Vdash_d (t_1 \leq t_2) \iff [\![t_1]\!]_{|\psi\rangle} \leq [\![t_2]\!]_{|\psi\rangle} \\
&|\psi\rangle \nVdash_d \perp\!\!\!\perp \\
&|\psi\rangle \Vdash_d (\gamma_1 \sqsupset \gamma_2) \iff (|\psi\rangle \nVdash_d \gamma_1) \vee (|\psi\rangle \Vdash_d \gamma_2)
\end{aligned}
$$

where $\Vdash_d$ is used to denote dEQPL satisfaction.

dEQPL - Model Checking

Given a set of qubits $\mathsf{qB} = \{q_1, \ldots, q_n\}$, a dEQPL formula γ and a state $|\psi\rangle \in \mathcal{H}_{\mathsf{qB}}$, check whether:

$$|\psi\rangle \Vdash_d \gamma$$

dEQPL - Model Checking

Given a set of qubits $\mathsf{qB} = \{q_1, \ldots, q_n\}$, a dEQPL formula γ and a state $|\psi\rangle \in \mathcal{H}_{\mathsf{qB}}$, check whether:

$$|\psi\rangle \Vdash_d \gamma$$

- **Complexity:** $\mathcal{O}(|\gamma| \cdot 2^n)$ (assuming arithmetic operations take time $\mathcal{O}(1)$).

dEQPL - Model Checking

Given a set of qubits $\mathsf{qB} = \{q_1, \ldots, q_n\}$, a dEQPL formula γ and a state $|\psi\rangle \in \mathcal{H}_{\mathsf{qB}}$, check whether:

$$|\psi\rangle \Vdash_d \gamma$$

- **Complexity:** $\mathcal{O}(|\gamma| \cdot 2^n)$ (assuming arithmetic operations take time $\mathcal{O}(1)$).
- Terms of the form $\int \alpha$ are responsible for the exponential factor.

Given a set of qubits $\mathsf{qB} = \{q_1, \ldots, q_n\}$, a dEQPL formula γ and a state $|\psi\rangle \in \mathcal{H}_{\mathsf{qB}}$, check whether:

$$|\psi\rangle \Vdash_d \gamma$$

- **Complexity:** $\mathcal{O}(|\gamma| \cdot 2^n)$ (assuming arithmetic operations take time $\mathcal{O}(1)$).
- Terms of the form $\int \alpha$ are responsible for the exponential factor.
- Computing $\mathcal{E}(\alpha)$ requires an iteration over all 2^n valuations:

$$\mathcal{E}(\alpha) = \{v \in 2^{\mathsf{qB}} : v \Vdash_c \alpha\}$$

QCTL

Syntax. Enrich dEQPL with *temporal modalities*:

$$\theta := \gamma \parallel \theta \sqsupset \theta \parallel \mathsf{EX}\theta \parallel \mathsf{AF}\theta \parallel \mathsf{E}[\theta \mathsf{U} \theta]$$

QCTL

Syntax. Enrich dEQPL with *temporal modalities*:

$$\theta := \gamma \parallel \theta \sqsupset \theta \parallel \mathsf{EX}\theta \parallel \mathsf{AF}\theta \parallel \mathsf{E}[\theta \mathsf{U} \theta]$$

Before defining the **semantics** of QCTL, an appropriate formalism for system modeling is required.

QCTL

Syntax. Enrich dEQPL with *temporal modalities*:

$$\theta := \gamma \,\|\, \theta \sqsupset \theta \,\|\, \mathsf{EX}\theta \,\|\, \mathsf{AF}\theta \,\|\, \mathsf{E}[\theta \mathsf{U} \theta]$$

Before defining the **semantics** of QCTL, an appropriate formalism for system modeling is required.

Definition (Quantum Kripke Structure)

Given a finite set of qubits qB, a *quantum Kripke structure* is a pair $\mathcal{T} = (S, R)$ where:

- $S \subset \mathcal{H}_{\mathsf{qB}}$ is the set of *states*. Each state $|\psi\rangle$ is a unit vector in $\mathcal{H}_{\mathsf{qB}}$.
- $R \subseteq S \times S$ is a *transition relation* such that for all $|\psi\rangle \in S$, there exists $|\psi\rangle' \in S$ such that $(|\psi\rangle, |\psi\rangle') \in R$.

QCTL

Syntax. Enrich dEQPL with *temporal modalities*:

$$\theta \coloneqq \gamma \,\|\, \theta \sqsupset \theta \,\|\, \mathsf{EX}\theta \,\|\, \mathsf{AF}\theta \,\|\, \mathsf{E}[\theta \mathsf{U} \theta]$$

Before defining the **semantics** of QCTL, an appropriate formalism for system modeling is required.

Definition (Quantum Kripke Structure)

Given a finite set of qubits qB, a *quantum Kripke structure* is a pair $\mathcal{T} = (S, R)$ where:

- $S \subset \mathcal{H}_{\mathsf{qB}}$ is the set of *states*. Each state $|\psi\rangle$ is a unit vector in $\mathcal{H}_{\mathsf{qB}}$.
- $R \subseteq S \times S$ is a *transition relation* such that for all $|\psi\rangle \in S$, there exists $|\psi\rangle' \in S$ such that $(|\psi\rangle, |\psi\rangle') \in R$.

- No labelling function needed!

Semantics.

- Given a quantum Kripke structure $\mathcal{T} = (S, R)$, a state $|\psi\rangle$ and a QCTL formulae θ the semantics of QCTL are defined by the relation $\Vdash_{\mathsf{Q}}$:

$$
\begin{aligned}
\mathcal{T}, |\psi_i\rangle \Vdash_{\mathsf{Q}} \gamma &\iff |\psi_i\rangle \Vdash_d \gamma \\
\mathcal{T}, |\psi_i\rangle \Vdash_{\mathsf{Q}} \theta_1 \sqsupset \theta_2 &\iff \mathcal{T}, |\psi_i\rangle \nVdash_{\mathsf{Q}} \theta_1 \vee \mathcal{T}, |\psi_i\rangle \Vdash_{\mathsf{Q}} \theta_2 \\
\mathcal{T}, |\psi_i\rangle \Vdash_{\mathsf{Q}} \mathsf{EX}\theta &\iff \exists\, |\psi'\rangle \in S,\ (|\psi_i\rangle, |\psi'\rangle) \in R \text{ and } \mathcal{T}, |\psi'\rangle \Vdash_{\mathsf{Q}} \theta \\
\mathcal{T}, |\psi_i\rangle \Vdash_{\mathsf{Q}} \mathsf{AF}\theta &\iff \forall\pi = |\psi_i\rangle\, |\psi_{i+1}\rangle\, |\psi_{i+2}\rangle \ldots, \exists\, j \geq i, \\
& \qquad (\mathcal{T}, |\psi_j\rangle \Vdash_{\mathsf{Q}} \theta) \\
\mathcal{T}, |\psi_i\rangle \Vdash_{\mathsf{Q}} \mathsf{E}[\theta_1 \,\mathsf{U}\, \theta_2] &\iff \exists\pi = |\psi_i\rangle\, |\psi_{i+1}\rangle\, |\psi_{i+2}\rangle \ldots, \exists\, j \geq i \\
& \qquad (\mathcal{T}, |\psi_j\rangle \Vdash_{\mathsf{Q}} \theta_2, \forall\, k,\ i \leq k < j,\ \mathcal{T}, |\psi_k\rangle \Vdash_{\mathsf{Q}} \theta_1)
\end{aligned}
$$

Model Checking.

- Given a QCTL formula θ and a quantum Kripke structure $\mathcal{T} = (S, R)$ compute:

$$Sat_{\mathcal{T}}(\theta) \coloneqq \{|\psi\rangle \in S : \mathcal{T}, |\psi\rangle \Vdash_{\mathsf{Q}} \theta\}$$

QCTL

Model Checking.

- Given a QCTL formula θ and a quantum Kripke structure $\mathcal{T} = (S, R)$ compute:

$$Sat_{\mathcal{T}}(\theta) \coloneqq \{|\psi\rangle \in S : \mathcal{T}, |\psi\rangle \Vdash_{\mathsf{Q}} \theta\}$$

- **Idea:** Drawing inspiration from *symbolic Model Checking*, QCTL formulae are characterized by sets of states.
- Temporal operators? Fixpoint characterization:
  - Consider the complete lattice $(\wp(S), \subseteq)$.
  - Bottom and top elements are respectively represented by $\perp\!\!\!\perp$ and $\top\!\!\!\top$.
  - Describe temporal modalities by suiting monotonic predicate transformers.

QCTL

- S is finite, the same results from fixpoint theory can be exploited.
- QCTL formulae are characterized as follows:

$$
\begin{aligned}
Sat_{\mathcal{T}}(\gamma) &= \{|\psi\rangle \in S : \psi \Vdash_{\mathsf{Q}} \gamma\} \\
Sat_{\mathcal{T}}(\theta_1 \sqsupset \theta_2) &= \big(S \setminus Sat_{\mathcal{T}}(\theta_1)\big) \cup Sat_{\mathcal{T}}(\theta_2) \\
Sat_{\mathcal{T}}(\mathsf{EX}\,\theta) &= \{|\psi\rangle \in S : \exists\, |\psi'\rangle \big((|\psi\rangle, |\psi'\rangle) \in R \wedge |\psi'\rangle \in Sat_{\mathcal{T}}(\theta)\big)\} \\
Sat_{\mathcal{T}}(\mathsf{AF}\,\theta) &= \mu Z.(\theta \vee \mathsf{AX} Z) \\
Sat_{\mathcal{T}}(\mathsf{E}[\theta_1 \mathsf{U} \theta_2]) &= \mu Z.(\theta_2 \vee (\theta_1 \wedge \mathsf{EX} Z))
\end{aligned}
$$

where $\mu Z.\tau(Z)$ represents the least fixpoint of τ.

QCTL - Model Checking

Complexity.

- Baltazar et al. analyze the complexity starting from the complexity of Model Checking on CTL: $\mathcal{O}(|\theta| \cdot (|S| + |R|))$

QCTL - Model Checking

Complexity.

- Baltazar et al. analyze the complexity starting from the complexity of Model Checking on CTL: $\mathcal{O}(|\theta| \cdot (|S| + |R|))$
- Note, there is a difference in checking atoms:

QCTL - Model Checking

Complexity.

- Baltazar et al. analyze the complexity starting from the complexity of Model Checking on CTL: $\mathcal{O}(|\theta| \cdot (|S| + |R|))$
- Note, there is a difference in checking atoms:
  - Classical atom: Constant time.
  - Quantum atom (or dEQPL formula γ): $\mathcal{O}(|\gamma| \cdot 2^n)$.

QCTL - Model Checking

Complexity.

- Baltazar et al. analyze the complexity starting from the complexity of Model Checking on CTL: $\mathcal{O}(|\theta| \cdot (|S| + |R|))$
- Note, there is a difference in checking atoms:
  - Classical atom: Constant time.
  - Quantum atom (or dEQPL formula γ): $\mathcal{O}(|\gamma| \cdot 2^n)$.
- Putting these results together:

$$\mathcal{O}(|\theta|^2 \cdot (|S| + |R|) \cdot 2^n)$$

where $n = |\mathsf{qB}|$.

QCTL - Model Checking

Complexity.

- Baltazar et al. analyze the complexity starting from the complexity of Model Checking on CTL: $\mathcal{O}(|\theta| \cdot (|S| + |R|))$
- Note, there is a difference in checking atoms:
  - Classical atom: Constant time.
  - Quantum atom (or dEQPL formula γ): $\mathcal{O}(|\gamma| \cdot 2^n)$.
- Putting these results together:

  $$\mathcal{O}(|\theta|^2 \cdot (|S| + |R|) \cdot 2^n)$$

  where $n = |\mathsf{qB}|$.
- **Last remark.** Time is polynomial w.r.t. the dimension of the model...

Complexity.

- Baltazar et al. analyze the complexity starting from the complexity of Model Checking on CTL: $\mathcal{O}(|\theta| \cdot (|S| + |R|))$
- Note, there is a difference in checking atoms:
  - Classical atom: Constant time.
  - Quantum atom (or dEQPL formula γ): $\mathcal{O}(|\gamma| \cdot 2^n)$.
- Putting these results together:

$$\mathcal{O}(|\theta|^2 \cdot (|S| + |R|) \cdot 2^n)$$

  where $n = |\mathsf{qB}|$.
- **Last remark.** Time is polynomial w.r.t. the dimension of the model...
  - **... but** simulating a quantum model with classical machinery requires exponential space.
  - More space is required to encode all possible 2^{2^n} superpositions of states.

Conclusion

Other techniques:

- ZX Circuits;

Conclusion

Other techniques:

- ZX Circuits;
- Path sums;

Conclusion

Other techniques:

- ZX Circuits;
- Path sums;
- Probabilistic Model Checking;

Conclusion

Other techniques:

- ZX Circuits;
- Path sums;
- Probabilistic Model Checking;
- ...

Conclusion

Other techniques:

- ZX Circuits;
- Path sums;
- Probabilistic Model Checking;
- ...

Quantum Hoare Logic is being constantly improved and extended:

- Theorem provers have been developed (*e.g.* QHLProver);
- Current problem: consider the Hoare triple $\{P\}S\{Q\}$
  - The dimensions of both P and Q grow exponentially w.r.t. the number of qubits in S.
  - Is there a way to *reason locally* on the satisfaction of quantum predicates?

Conclusion

QCTL. Ways to extend the work:

- Generalization to arbitrary measurements.

Conclusion

QCTL. Ways to extend the work:

- Generalization to arbitrary measurements.
- More rigorous analysis of the problem of Model Checking.

Conclusion

QCTL. Ways to extend the work:

- Generalization to arbitrary measurements.
- More rigorous analysis of the problem of Model Checking.
- *Quantum State Explosion Problem*:

Conclusion

QCTL. Ways to extend the work:

- Generalization to arbitrary measurements.
- More rigorous analysis of the problem of Model Checking.
- *Quantum State Explosion Problem*:
  - Cartesian products of sets of states? Worse.
  - Tensor products of the entire Hilbert spaces of each component system:

Conclusion

QCTL. Ways to extend the work:

- Generalization to arbitrary measurements.
- More rigorous analysis of the problem of Model Checking.
- *Quantum State Explosion Problem*:
    - Cartesian products of sets of states? Worse.
    - Tensor products of the entire Hilbert spaces of each component system:

> *"Hilbert space is a big place."*
>
> – Carlton Caves

The End

Thank you for your attention.

The End

Thank you for your attention.

Unless... there is more time!

Case Study: Deutsch's Algorithm

- **The problem:** given a function $f : \{0, 1\} \to \{0, 1\}$ compute $f(0) \oplus f(1)$.

Case Study: Deutsch's Algorithm

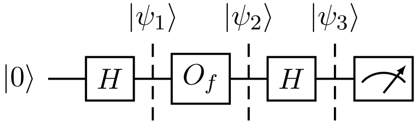
- **The problem:** given a function $f : \{0,1\} \to \{0,1\}$ compute $f(0) \oplus f(1)$.
  - Classical case: Two calls of function of f.

Case Study: Deutsch's Algorithm

- **The problem:** given a function $f : \{0,1\} \to \{0,1\}$ compute $f(0) \oplus f(1)$.
  - Classical case: Two calls of function of f.
  - Quantum case: **One call**, thanks to *quantum parallelism*.

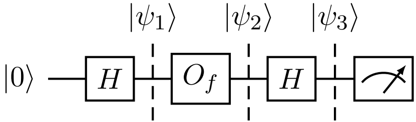
Case Study: Deutsch's Algorithm

- **The problem:** given a function $f : \{0,1\} \to \{0,1\}$ compute $f(0) \oplus f(1)$.
  - Classical case: Two calls of function of f.
  - Quantum case: **One call**, thanks to *quantum parallelism*.
- The algorithm can be described by the following circuit:



Case Study: Deutsch's Algorithm

- **The problem:** given a function $f : \{0,1\} \to \{0,1\}$ compute $f(0) \oplus f(1)$.
  - Classical case: Two calls of function of f.
  - Quantum case: **One call**, thanks to *quantum parallelism*.
- The algorithm can be described by the following circuit:

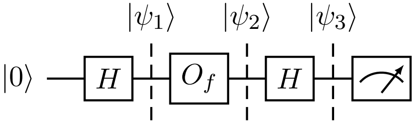


- H denotes a Hadamard gate, briefly it acts on $\mathcal{H}^2$ as follows:

$$H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} = |+\rangle \qquad H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} = |-\rangle$$

$$H\frac{|0\rangle + |1\rangle}{\sqrt{2}} = |0\rangle \qquad H\frac{|0\rangle - |1\rangle}{\sqrt{2}} = |1\rangle$$

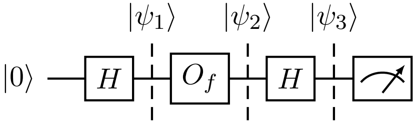
Case Study: Deutsch's Algorithm



- O_f represents an oracle implementing a call to function f:

$$O_f = \begin{pmatrix} (-1)^{f(0)} & 0 \\ 0 & (-1)^{f(1)} \end{pmatrix}$$

Case Study: Deutsch's Algorithm

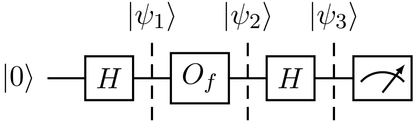


- O_f represents an oracle implementing a call to function f:

$$O_f = \begin{pmatrix} (-1)^{f(0)} & 0 \\ 0 & (-1)^{f(1)} \end{pmatrix}$$

- The last gate is a measurement on the computational basis.

Case Study: Deutsch's Algorithm



- O_f represents an oracle implementing a call to function f:

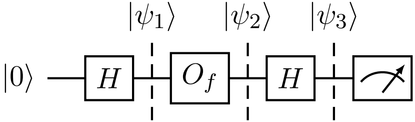
$$O_f = \begin{pmatrix} (-1)^{f(0)} & 0 \\ 0 & (-1)^{f(1)} \end{pmatrix}$$

- The last gate is a measurement on the computational basis.
- Dashed lines denote the evolution of the system throughout the circuit:

$$|\psi_1\rangle = H\,|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

$$|\psi_2\rangle = O_f\,|\psi_1\rangle = \frac{(-1)^{f(0)}\,|0\rangle + (-1)^{f(1)}\,|1\rangle}{\sqrt{2}}$$

Case Study: Deutsch's Algorithm



- O_f represents an oracle implementing a call to function f:

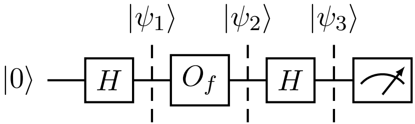
$$O_f = \begin{pmatrix} (-1)^{f(0)} & 0 \\ 0 & (-1)^{f(1)} \end{pmatrix}$$

- The last gate is a measurement on the computational basis.
- Dashed lines denote the evolution of the system throughout the circuit:

$$|\psi_1\rangle = H\,|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

$$|\psi_2\rangle = O_f\,|\psi_1\rangle = \frac{(-1)^{f(0)}\,|0\rangle + (-1)^{f(1)}\,|1\rangle}{\sqrt{2}}$$

Case Study: Deutsch's Algorithm



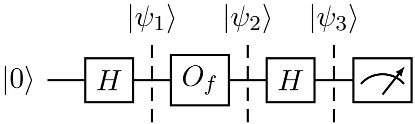
$$|\psi_2\rangle = O_f |\psi_1\rangle = \frac{(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle}{\sqrt{2}}$$

- $|\psi_2\rangle$ can be represented as follows:

$$|\psi_2\rangle = \begin{cases} \pm\frac{|0\rangle+|1\rangle}{\sqrt{2}}, \text{ if } f(0) = f(1) \\ \pm\frac{|0\rangle-|1\rangle}{\sqrt{2}}, \text{ if } f(0) \neq f(1) \end{cases}$$

$$|\psi_3\rangle = H|\psi_2\rangle = \begin{cases} \pm|0\rangle, \text{ if } f(0) = f(1) \\ \pm|1\rangle, \text{ if } f(0) \neq f(1) \end{cases}$$

Case Study: Deutsch's Algorithm



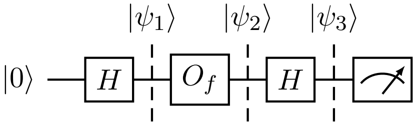
$$|\psi_2\rangle = O_f\,|\psi_1\rangle = \frac{(-1)^{f(0)}\,|0\rangle + (-1)^{f(1)}\,|1\rangle}{\sqrt{2}}$$

- $|\psi_2\rangle$ can be represented as follows:

$$|\psi_2\rangle = \begin{cases} \pm\frac{|0\rangle+|1\rangle}{\sqrt{2}}, \text{ if } f(0) = f(1) \\ \pm\frac{|0\rangle-|1\rangle}{\sqrt{2}}, \text{ if } f(0) \neq f(1) \end{cases}$$

$$|\psi_3\rangle = H\,|\psi_2\rangle = \begin{cases} \pm\,|0\rangle\,, \text{ if } f(0) = f(1) \\ \pm\,|1\rangle\,, \text{ if } f(0) \neq f(1) \end{cases}$$

Case Study: Deutsch's Algorithm



$$|\psi_3\rangle = H|\psi_2\rangle = \begin{cases} \pm|0\rangle, \text{ if } f(0) = f(1) \\ \pm|1\rangle, \text{ if } f(0) \neq f(1) \end{cases}$$

- Now the measurement $M = \{|0\rangle\langle 0|, |1\rangle\langle 1|\}$ will give rise to:
  - $|0\rangle$ with probability 1 if $f(0) = f(1)$.
  - $|1\rangle$ with probability 1 if $f(0) \neq f(1)$.

Quantum Weakest Precondition

- First, describe Deutsch's algorithm into the quantum **while**-language:

$$\begin{aligned} Deutsch \equiv [&q := 0; \\ &q := Hq; \\ &q := O_f q; \\ &q := Hq; \\ &\texttt{measure}\, M[q] : \texttt{skip}; \texttt{skip}] \end{aligned}$$

- The Postcondition:

$$Post = (1 - f(0) \oplus f(1)) \left|0\right\rangle \left\langle 0\right| + f(0) \oplus f(1) \left|1\right\rangle \left\langle 1\right|$$

- *Post* states a property that is wished to be proven valid.

- The Postcondition:

$$Post = (1 - f(0) \oplus f(1)) |0\rangle \langle 0| + f(0) \oplus f(1) |1\rangle \langle 1|$$

- *Post* states a property that is wished to be proven valid.
- Now, running backwards through the statements of *Deutsch*:

$$\mathsf{qwp}.[\texttt{measure}\, M[q] : \texttt{skip}; \texttt{skip}](Post) = M_0^\dagger Post M_0 + M_1^\dagger Post M_1$$
$$= |0\rangle \langle 0| c |0\rangle \langle 0|0\rangle \langle 0| + |1\rangle \langle 1| b |1\rangle \langle 1|1\rangle |1\rangle$$
$$= c |0\rangle \langle 0| + b |1\rangle \langle 1| = Post$$

where $c = 1 - f(0) \oplus f(1)$ and $b = f(0) \oplus f(1)$.

Case Study: Deutsch's Algorithm

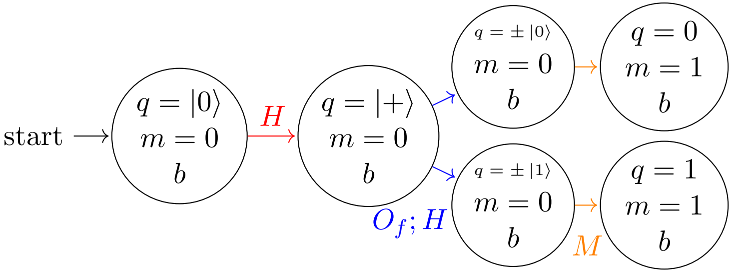
- Eventually one obtains I as the weakest precondition of the first statement.
- Applying the function on the composition of statements:

$$\mathsf{qwp}.[Deutsch](Post) = I$$

- The Hoare triple $\{I\}Deutsch\{Post\}$ is totally correct.
- Any other precondition P is such that $P \sqsubseteq I$.
- **Meaning:** Under any assumption, *Deutsch's algorithm is always correct.*

Case Study: Deutsch's Algorithm

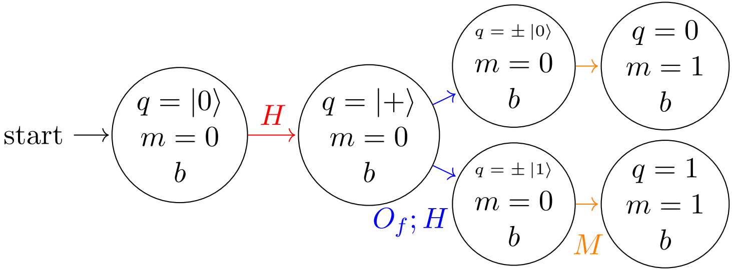
QCTL: The quantum Kripke structure:



where:

- q is the qubit
- m denotes whether q has been measured $(= 1)$ or not $(= 0)$
- $b = f(0) \oplus f(1)$.

Case Study: Deutsch's Algorithm



- Zuliani et al. describe the correctness of Deutsch's algorithm through the following QCTL formula:

$$\theta = \mathsf{A}\Big[(\boxminus(\Box m))\,\mathsf{U}\Big(\Box m \sqcap (\Box b \equiv (\textstyle\int q = 1))\Big)\Big]$$

- Shorthands:
  - $\Box x$ states $x = 1$;
  - $\boxminus$: quantum negation.
  - $\sqcap$: quantum conjunction.

The End

(for real this time)

Thank you for your attention.