Model checking quantum systems

K. Grover

Outline

1. Density operator
2. Quantum program verification
3. Quantum Finite Automata

Density operator

Density Operator

For **closed** systems, (pure) state is a vector $|\psi\rangle$.

Density Operator

For **closed** systems, (pure) state is a vector $|\psi\rangle$.

For **open** systems, the state is not completely known.

Density Operator

For **closed** systems, (pure) state is a vector $|\psi\rangle$.

For **open** systems, the state is not completely known.
It's in one of the pure states $|\psi_i\rangle$, with probabilities p_i.

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

Density Operator

For **closed** systems, (pure) state is a vector $|\psi\rangle$.

For **open** systems, the state is not completely known.
It's in one of the pure states $|\psi_i\rangle$, with probabilities p_i.

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

Evolution by a unitary operator:

$$\rho = U\rho_0 U^\dagger$$

Quantum program verification

Classical programming language

$$
\begin{aligned}
S ::= \textbf{skip} \mid & \; x := a \\
\mid & \; S_1; S_2 \\
\mid & \; \textbf{if } b \textbf{ then } S_1 \textbf{ else } S_2 \\
\mid & \; \textbf{while } b \textbf{ do } S \textbf{ od}
\end{aligned}
$$

Hoare Logic

Hoare triple:

$$\{P\}S\{Q\}$$

Hoare Logic

Hoare triple:

$$\{P\}S\{Q\}$$

It is **valid** if it can be generated by a sequence of inference rules.

$$\overline{\{P\}\textbf{skip}\{P\}}$$

Hoare Logic

Hoare triple:

$$\{P\}S\{Q\}$$

It is **valid** if it can be generated by a sequence of inference rules.

$$\frac{}{\{P\}\textbf{skip}\{P\}}$$

$$\frac{}{\{P[a/x]\}x := a\{P\}}$$

Hoare Logic

Hoare triple:

$$\{P\}S\{Q\}$$

It is **valid** if it can be generated by a sequence of inference rules.

$$\frac{}{\{P\}\textbf{skip}\{P\}}$$

$$\frac{}{\{P[a/x]\}x := a\{P\}}$$

$$\frac{\{P\}S_1\{Q\}, \{Q\}S_2\{R\}}{\{P\}S_1; S_2\{R\}}$$

Hoare Logic

Hoare triple:

$$\{P\}S\{Q\}$$

It is **valid** if it can be generated by a sequence of inference rules.

$$\frac{}{\{P\}\textbf{skip}\{P\}}$$

$$\frac{}{\{P[a/x]\}x := a\{P\}}$$

$$\frac{\{P\}S_1\{Q\}, \{Q\}S_2\{R\}}{\{P\}S_1; S_2\{R\}}$$

$$\frac{\{b \wedge P\}S_1\{Q\}, \{\neg b \wedge P\}S_2\{Q\}}{\{P\}\textbf{if } b \textbf{ then } S_1 \textbf{ else } S_2\{Q\}}$$

Hoare Logic

Hoare triple:

$$\{P\}S\{Q\}$$

It is **valid** if it can be generated by a sequence of inference rules.

$$\frac{}{\{P\}\mathbf{skip}\{P\}}$$

$$\frac{}{\{P[a/x]\}x := a\{P\}}$$

$$\frac{\{P\}S_1\{Q\}, \{Q\}S_2\{R\}}{\{P\}S_1; S_2\{R\}}$$

$$\frac{\{b \wedge P\}S_1\{Q\}, \{\neg b \wedge P\}S_2\{Q\}}{\{P\}\mathbf{if}\ b\ \mathbf{then}\ S_1\ \mathbf{else}\ S_2\{Q\}}$$

$$\frac{\{b \wedge P\}S\{P\}}{\{P\}\mathbf{while}\ b\ \mathbf{do}\ S\{\neg b \wedge P\}}$$

Hoare Logic

Hoare triple:

$$\{P\}S\{Q\}$$

It is **valid** if it can be generated by a sequence of inference rules.

$$\frac{}{\{P\}\textbf{skip}\{P\}}$$

$$\frac{}{\{P[a/x]\}x := a\{P\}}$$

$$\frac{\{P\}S_1\{Q\}, \{Q\}S_2\{R\}}{\{P\}S_1; S_2\{R\}}$$

$$\frac{\{b \wedge P\}S_1\{Q\}, \{\neg b \wedge P\}S_2\{Q\}}{\{P\}\textbf{if } b \textbf{ then } S_1 \textbf{ else } S_2\{Q\}}$$

$$\frac{\{b \wedge P\}S\{P\}}{\{P\}\textbf{while } b \textbf{ do } S\{\neg b \wedge P\}}$$

$$\frac{P_2 \rightarrow P_1, \{P_2\}S\{Q_2\}, Q_2 \rightarrow Q_1}{\{P_1\}S\{Q_1\}}$$

Quantum programming language

$$
\begin{aligned}
S ::= \mathbf{skip} \mid\ & q := |0\rangle \\
\mid\ & S_1; S_2 \\
\mid\ & \bar{q} := U[\bar{q}] \\
\mid\ & \mathbf{if}\ (\Box m \cdot M[\bar{q}] = m \rightarrow S_m)\ \mathbf{fi} \\
\mid\ & \mathbf{while}\ M[\bar{q}] = 1\ \mathbf{do}\ S\ \mathbf{od}
\end{aligned}
$$

Quantum Hoare logic

A **quantum predicate** is an observable (Hermitian operator) P such that $0 \sqsubseteq P \sqsubseteq I$.

Quantum Hoare logic

A **quantum predicate** is an observable (Hermitian operator) P such that $0 \sqsubseteq P \sqsubseteq I$.

If system is in state ρ, probability that P is satisfied: $tr(P\rho)$

Quantum Hoare logic

A **quantum predicate** is an observable (Hermitian operator) P such that $0 \sqsubseteq P \sqsubseteq I$.

If system is in state ρ, probability that P is satisfied: $tr(P\rho)$

A **Hoare triple** here is similar:

$$\{P\}S\{Q\}$$

Inference rules

$$\frac{}{\{P\}\textbf{skip}\{P\}}$$

Inference rules

$$\overline{\{P\}\mathbf{skip}\{P\}}$$

$$\overline{\{\sum_{n=0}^{1}|n\rangle_q\langle 0|P|0\rangle_q\langle n|\}q := |0\rangle\{P\}}$$

Inference rules

$$\frac{}{\{P\}\textbf{skip}\{P\}}$$

$$\frac{}{\{\sum_{n=0}^{1} |n\rangle_q \langle 0|P|0\rangle_q \langle n|\} q := |0\rangle \{P\}}$$

$$\frac{}{\{U^\dagger P U\}\bar{q} := U[\bar{q}]\{P\}}$$

Inference rules

$$\overline{\{P\}\textbf{skip}\{P\}}$$

$$\overline{\{\sum_{n=0}^{1} |n\rangle_q\langle 0|P|0\rangle_q\langle n|\}q := |0\rangle\{P\}}$$

$$\overline{\{U^\dagger PU\}\bar{q} := U[\bar{q}]\{P\}}$$

$$\frac{\{P\}S_1\{Q\}, \{Q\}S_2\{R\}}{\{P\}S_1; S_2\{R\}}$$

Inference rules

$$\overline{\{P\}\textbf{skip}\{P\}}$$

$$\overline{\{\sum_{n=0}^{1} |n\rangle_q \langle 0|P|0\rangle_q \langle n|\} q := |0\rangle \{P\}}$$

$$\overline{\{U^\dagger P U\} \bar{q} := U[\bar{q}] \{P\}}$$

$$\frac{\{P\}S_1\{Q\}, \{Q\}S_2\{R\}}{\{P\}S_1; S_2\{R\}}$$

$$\frac{\{P_m\}S_m\{Q\} \text{ for all } m}{\{\sum_m M_m^\dagger P_m M_m\} \textbf{ if } (\square m \cdot M[\bar{q}] = m \rightarrow S_m) \textbf{ fi } \{Q\}}$$

Inference rules

$$\frac{}{\{P\}\textbf{skip}\{P\}}$$

$$\frac{}{\{\sum_{n=0}^{1} |n\rangle_q\langle 0|P|0\rangle_q\langle n|\}q := |0\rangle\{P\}}$$

$$\frac{}{\{U^\dagger PU\}\bar{q} := U[\bar{q}]\{P\}}$$

$$\frac{\{P\}S_1\{Q\}, \{Q\}S_2\{R\}}{\{P\}S_1; S_2\{R\}}$$

$$\frac{\{P_m\}S_m\{Q\} \text{ for all } m}{\{\sum_m M_m^\dagger P_m M_m\} \textbf{ if } (\Box m \cdot M[\bar{q}] = m \rightarrow S_m) \textbf{ fi } \{Q\}}$$

$$\frac{\{Q\}S\{M_0^\dagger PM_0 + M_1^\dagger QM_1\}}{\{M_0^\dagger PM_0 + M_1^\dagger QM_1\} \textbf{ while } M[\bar{q}] = m \textbf{ do } S\{P\}}$$

Inference rules

$$\frac{}{\{P\}\textbf{skip}\{P\}}$$

$$\frac{}{\{\sum_{n=0}^{1} |n\rangle_q\langle 0|P|0\rangle_q\langle n|\}q := |0\rangle\{P\}}$$

$$\frac{}{\{U^\dagger PU\}\bar{q} := U[\bar{q}]\{P\}}$$

$$\frac{\{P\}S_1\{Q\}, \{Q\}S_2\{R\}}{\{P\}S_1; S_2\{R\}}$$

$$\frac{\{P_m\}S_m\{Q\} \text{ for all } m}{\{\sum_m M_m^\dagger P_m M_m\} \textbf{ if } (\square m \cdot M[\bar{q}] = m \to S_m) \textbf{ fi } \{Q\}}$$

$$\frac{\{Q\}S\{M_0^\dagger PM_0 + M_1^\dagger QM_1\}}{\{M_0^\dagger PM_0 + M_1^\dagger QM_1\} \textbf{ while } M[\bar{q}] = m \textbf{ do } S\{P\}}$$

$$\frac{P_1 \sqsubseteq P_2, \{P_2\}S\{Q_2\}, Q_2 \sqsubseteq Q_1}{\{P_1\}S\{Q_1\}}$$

Correctness

- **Total Correctness:**

$$\models_{tot} \{P\}S\{Q\} \text{ if}$$
$$tr(P\rho) \leq tr(Q[\![S]\!](\rho)) \text{ for all } \rho.$$

- **Partial Correctness:**

$$\models_{par} \{P\}S\{Q\} \text{ if}$$
$$tr(P\rho) \leq tr(Q[\![S]\!](\rho)) + [tr(\rho) - tr([\![S]\!](\rho)] \text{ for all } \rho.$$

Example

$$S = \textbf{while } M[q] = 1 \textbf{ do } q := \sigma_z q$$

Example

$$S = \textbf{while } M[q] = 1 \textbf{ do } q := \sigma_z q$$

$$P = |\psi\rangle_q\langle\psi| \otimes P' \text{ where } \psi = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \text{ and } P' \in \mathcal{P}(\mathcal{H}_{Var-\{q\}})$$

Example

$$S = \textbf{while } M[q] = 1 \textbf{ do } q := \sigma_z q$$

$$P = |\psi\rangle_q\langle\psi| \otimes P' \text{ where } \psi = \begin{pmatrix}\alpha \\ \beta\end{pmatrix} \text{ and } P' \in \mathcal{P}(\mathcal{H}_{Var-\{q\}})$$

Then,

$$\models_{tot} \{P\}S\{|0\rangle q\langle 0| \otimes P'\}$$

Example

$$S = \textbf{while } M[q] = 1 \textbf{ do } q := \sigma_z q$$

$$P = |\psi\rangle_q\langle\psi| \otimes P' \text{ where } \psi = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \text{ and } P' \in \mathcal{P}(\mathcal{H}_{Var-\{q\}})$$

Then,

$$\models_{tot} \{P\}S\{|0\rangle_q\langle 0| \otimes P'\}$$

does not hold if $\beta \neq 0$ and $P \neq 0_{\mathcal{H}_{Var-\{q\}}}$.

Example

$$S = \textbf{while } M[q] = 1 \textbf{ do } q := \sigma_z q$$

$$P = |\psi\rangle_q\langle\psi| \otimes P' \text{ where } \psi = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \text{ and } P' \in \mathcal{P}(\mathcal{H}_{Var-\{q\}})$$

Then,

$$\models_{tot} \{P\}S\{|0\rangle q\langle 0| \otimes P'\}$$

does not hold if $\beta \neq 0$ and $P \neq 0_{\mathcal{H}_{Var-\{q\}}}$.

In fact, put

$$\rho = |\psi\rangle_q\langle\psi| \otimes I_{\mathcal{H}_{Var-\{q\}}}$$

Example

$$S = \textbf{while } M[q] = 1 \textbf{ do } q := \sigma_z q$$

$$P = |\psi\rangle_q\langle\psi| \otimes P' \text{ where } \psi = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \text{ and } P' \in \mathcal{P}(\mathcal{H}_{Var-\{q\}})$$

Then,

$$\models_{tot} \{P\}S\{|0\rangle_q\langle 0| \otimes P'\}$$

does not hold if $\beta \neq 0$ and $P \neq 0_{\mathcal{H}_{Var-\{q\}}}$.

In fact, put

$$\rho = |\psi\rangle_q\langle\psi| \otimes I_{\mathcal{H}_{Var-\{q\}}}$$

Then

$$[\![S]\!](\rho) = |\alpha|^2|0\rangle_q\langle 0| \otimes I_{\mathcal{H}_{Var-\{q\}}}$$

Example

$$S = \textbf{while } M[q] = 1 \textbf{ do } q := \sigma_z q$$

$$P = |\psi\rangle_q\langle\psi| \otimes P' \text{ where } \psi = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \text{ and } P' \in \mathcal{P}(\mathcal{H}_{Var-\{q\}})$$

Then,

$$\models_{tot} \{P\}S\{|0\rangle q\langle 0| \otimes P'\}$$

does not hold if $\beta \neq 0$ and $P \neq 0_{\mathcal{H}_{Var-\{q\}}}$.

In fact, put

$$\rho = |\psi\rangle_q\langle\psi| \otimes I_{\mathcal{H}_{Var-\{q\}}}$$

Then

$$[\![S]\!](\rho) = |\alpha|^2|0\rangle_q\langle 0| \otimes I_{\mathcal{H}_{Var-\{q\}}}$$

$$tr(P\rho) = tr(P') > |\alpha|^2 tr(P') = tr((|0\rangle_q\langle 0| \otimes P')[\![S]\!](\rho))$$

Example

$$S = \textbf{while } M[q] = 1 \textbf{ do } q := \sigma_z q$$

$$P = |\psi\rangle_q\langle\psi| \otimes P' \text{ where } \psi = \begin{pmatrix}\alpha \\ \beta\end{pmatrix} \text{ and } P' \in \mathcal{P}(\mathcal{H}_{Var-\{q\}})$$

Then,

$$\models_{tot} \{P\}S\{|0\rangle q\langle 0| \otimes P'\}$$

does not hold if $\beta \neq 0$ and $P \neq 0_{\mathcal{H}_{Var-\{q\}}}$.

In fact, put

$$\rho = |\psi\rangle_q\langle\psi| \otimes I_{\mathcal{H}_{Var-\{q\}}}$$

Then

$$[\![S]\!](\rho) = |\alpha|^2|0\rangle_q\langle 0| \otimes I_{\mathcal{H}_{Var-\{q\}}}$$

$$tr(P\rho) = tr(P') > |\alpha|^2 tr(P') = tr((|0\rangle_q\langle 0| \otimes P')[\![S]\!](\rho))$$

On the other hand, we can show partial correctness always holds:

$$\models_{par} \{P\}S\{|0\rangle_q\langle 0| \otimes P'\}$$

Example

$$q := 0;\ \ q := Hq;\ \ q := O_f\, q;\ \ q := Hq;\ \ \textbf{if } \square m \cdot M[q] : \textbf{skip}, \textbf{skip fi}$$

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

Example

$$q := 0;\ \ q := Hq;\ \ q := O_f q;\ \ q := Hq;\ \ \textbf{if } \Box m \cdot M[q] : \textbf{skip}, \textbf{skip fi}$$

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

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

Example

$$q := 0;\ \ q := Hq;\ \ q := O_f\, q;\ \ q := Hq;\ \ \mathbf{if}\ \square m \cdot M[q] : \mathbf{skip}, \mathbf{skip}\ \mathbf{fi}$$

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

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

$$wp.(\mathbf{if}\ \square m \cdot M[q] : (\mathbf{skip}, \mathbf{skip})\ \mathbf{fi}).Q := M_0^\dagger(wp.(\mathbf{skip}).Q)M_0 + M_1^\dagger(wp.(\mathbf{skip}).Q)M_1$$

Example

$$q := 0;\ \ q := Hq;\ \ q := O_f q;\ \ q := Hq;\ \ \textbf{if}\ \Box m \cdot M[q] : \textbf{skip}, \textbf{skip}\ \textbf{fi}$$

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

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

$$wp.(\textbf{if}\ \Box m \cdot M[q] : (\textbf{skip}, \textbf{skip})\ \textbf{fi}).Q := M_0^\dagger(wp.(\textbf{skip}).Q)M_0 + M_1^\dagger(wp.(\textbf{skip}).Q)M_1$$

$$= M_0^\dagger Q M_0 + M_1^\dagger Q M_1$$

Example

$$q := 0;\ \ q := Hq;\ \ q := O_f q;\ \ q := Hq;\ \ \textbf{if}\ \square m \cdot M[q] : \textbf{skip}, \textbf{skip}\ \textbf{fi}$$

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

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

$$wp.(\textbf{if}\ \square m \cdot M[q] : (\textbf{skip}, \textbf{skip})\ \textbf{fi}).Q := M_0^\dagger(wp.(\textbf{skip}).Q)M_0 + M_1^\dagger(wp.(\textbf{skip}).Q)M_1$$

$$= M_0^\dagger Q M_0 + M_1^\dagger Q M_1$$

$$= Q$$

Example

$$q := 0;\ \ q := Hq;\ \ q := O_f q;\ \ q := Hq;\ \ \mathbf{if}\ \Box m \cdot M[q] : \mathbf{skip}, \mathbf{skip}\ \mathbf{fi}$$

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

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

$wp.(\mathbf{if}\ \Box m \cdot M[q] : (\mathbf{skip}, \mathbf{skip})\ \mathbf{fi}).Q := M_0^\dagger(wp.(\mathbf{skip}).Q)M_0 + M_1^\dagger(wp.(\mathbf{skip}).Q)M_1$

$$= M_0^\dagger Q M_0 + M_1^\dagger Q M_1$$

$$= Q$$

Keep on finding weakest pre-condition and validate the Hoare triple $\{I\}S\{Q\}$

What else?

- Can generate some loop invariants.

What else?

- Can generate some loop invariants.
- Still cannot implement Shor's algorithm. But, there's an extension.

What else?

- Can generate some loop invariants.
- Still cannot implement Shor's algorithm. But, there's an extension.
- Termination analysis of Non-deterministic quantum programs.

Quantum Finite Automata

Quantum Finite Automata

FA

- S : Finite set of states (n).

QFA

- $\mathcal{H}$: Hilbert space (n-dim).

Quantum Finite Automata

FA

- S : Finite set of states (n).
- Σ : Alphabet, finite set of letters.

QFA

- $\mathcal{H}$: Hilbert space (n-dim).
- Act : Finite set of actions.

Quantum Finite Automata

FA

- S : Finite set of states (n).
- Σ : Alphabet, finite set of letters.
- Δ : Transition function.

QFA

- $\mathcal{H}$: Hilbert space (n-dim).
- Act : Finite set of actions.
- $\{U_\alpha \mid \alpha \in Act\}$: Set of transition matrices.

$$|\phi\rangle = U_\alpha|\psi\rangle$$

Quantum Finite Automata

FA

- S : Finite set of states (n).
- Σ : Alphabet, finite set of letters.
- Δ : Transition function.
- S_0: Set of initial states.

QFA

- $\mathcal{H}$: Hilbert space (n-dim).
- Act : Finite set of actions.
- $\{U_\alpha \mid \alpha \in Act\}$: Set of transition matrices.
- $\mathcal{H}_0$: Subspace of initial states.

Quantum Finite Automata

FA

- S : Finite set of states (n).
- Σ : Alphabet, finite set of letters.
- Δ : Transition function.
- S_0: Set of initial states.

QFA

- $\mathcal{H}$: Hilbert space (n-dim).
- Act : Finite set of actions.
- $\{U_\alpha \mid \alpha \in Act\}$: Set of transition matrices.
- $\mathcal{H}_0$: Subspace of initial states.

A **path** is defined in an obvious way and a state is **reachable** if there is a path from an initial state to that state.

Birkhoff-von Neumann Quantum Logic

- Atomic Propositions: $|\phi\rangle \in \mathcal{X}$

Birkhoff-von Neumann Quantum Logic

- Atomic Propositions: $|\phi\rangle \in \mathcal{X}$
- Each subspace $\mathcal{Z}$ represents a formula.

Birkhoff-von Neumann Quantum Logic

- Atomic Propositions: $|\phi\rangle \in \mathcal{X}$
- Each subspace $\mathcal{Z}$ represents a formula.
- Labeling of each state: the set of atomic propositions true there.

Birkhoff-von Neumann Quantum Logic

- Atomic Propositions: $|\phi\rangle \in \mathcal{X}$
- Each subspace $\mathcal{Z}$ represents a formula.
- Labeling of each state: the set of atomic propositions true there.
- A state satisfies a formula $\mathcal{Z} \in \mathcal{S}(\mathcal{H})$

$$|\phi\rangle \models \mathcal{Z} \text{ if } \bigcap_{\mathcal{Y} \in L(|\psi\rangle)} \mathcal{Y} \subseteq \mathcal{Z}$$

Example

$\mathcal{H}$ is an n-dim Hilbert space.

Example

$\mathcal{H}$ is an n-dim Hilbert space.

$$\psi = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

Example

$\mathcal{H}$ is an n-dim Hilbert space.

$$\psi = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

$$AP = \{\mathcal{Y} \in \mathcal{S}(\mathcal{H}) : |0\rangle \perp \mathcal{Y}\}$$

Example

$\mathcal{H}$ is an *n*-dim Hilbert space.

$$\psi = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

$$AP = \{\mathcal{Y} \in \mathcal{S}(\mathcal{H}) : |0\rangle \perp \mathcal{Y}\}$$

then $L(|\psi\rangle) = \phi$ and

Example

$\mathcal{H}$ is an n-dim Hilbert space.

$$\psi = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

$$AP = \{\mathcal{Y} \in \mathcal{S}(\mathcal{H}) : |0\rangle \perp \mathcal{Y}\}$$

then $L(|\psi\rangle) = \phi$ and

$$\bigcap_{\mathcal{Y} \in L(|\psi\rangle)} \mathcal{Y} = \mathcal{H}$$

Example

$\mathcal{H}$ is an n-dim Hilbert space.

$$\psi = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

$$AP = \{\mathcal{Y} \in \mathcal{S}(\mathcal{H}) : |0\rangle \perp \mathcal{Y}\}$$

then $L(|\psi\rangle) = \phi$ and

$$\bigcap_{\mathcal{Y} \in L(|\psi\rangle)} \mathcal{Y} = \mathcal{H}$$

Thus, for any $\mathcal{X} \in \mathcal{S}(\mathcal{H})$, $|\psi\rangle \models \mathcal{X}$ if and only if $\mathcal{X} = \mathcal{H}$.

Example

Let AP be the 2-dimensional subspaces of $\mathcal{H}$.

Example

Let AP be the 2-dimensional subspaces of $\mathcal{H}$.

If $n = 2$,

$$\bigcap_{\mathcal{Y} \in L(|\psi\rangle)} \mathcal{Y} = \mathcal{H}$$

Example

Let AP be the 2-dimensional subspaces of $\mathcal{H}$.

If $n = 2$,

$$\bigcap_{\mathcal{Y} \in L(|\psi\rangle)} \mathcal{Y} = \mathcal{H}$$

and $|\psi\rangle \models \mathcal{X}$ if and only if $\mathcal{X} = \mathcal{H}$

Example

Let AP be the 2-dimensional subspaces of $\mathcal{H}$.

If $n = 2$,

$$\bigcap_{\mathcal{Y} \in L(|\psi\rangle)} \mathcal{Y} = \mathcal{H}$$

and $|\psi\rangle \models \mathcal{X}$ if and only if $\mathcal{X} = \mathcal{H}$

If $n > 2$,

$$\bigcap_{\mathcal{Y} \in L(|\psi\rangle)} \mathcal{Y} = span\{|\psi\rangle\}$$

Example

Let AP be the 2-dimensional subspaces of $\mathcal{H}$.

If $n = 2$,

$$\bigcap_{\mathcal{Y} \in L(|\psi\rangle)} \mathcal{Y} = \mathcal{H}$$

and $|\psi\rangle \models \mathcal{X}$ if and only if $\mathcal{X} = \mathcal{H}$

If $n > 2$,

$$\bigcap_{\mathcal{Y} \in L(|\psi\rangle)} \mathcal{Y} = span\{|\psi\rangle\}$$

and $|\psi\rangle \models \mathcal{X}$ if and only if $|\psi\rangle \in \mathcal{X}$.

Linear time properties

- A property P is a set of finite and infinite sequences of sets of atomic propositions.

Linear time properties

- A property P is a set of finite and infinite sequences of sets of atomic propositions.
- $A \models P$ if the set of traces is contained in P.

Linear time properties

- A property P is a set of finite and infinite sequences of sets of atomic propositions.
- $A \models P$ if the set of traces is contained in P.
- Try to model check different types of linear time properties like safety, liveness, invariant, persistence and reachability.

Invariant

Check whether $\mathcal{A} \models \mathit{inv}\ \mathcal{X}$

Invariant

Check whether $\mathcal{A} \models inv\ \mathcal{X}$

Can do liveness and persistence similarly.

Meta-propositional logic

Can have complement of subspaces and union of two subspaces (not necessarily a subspace).

Meta-propositional logic

Can have complement of subspaces and union of two subspaces (not necessarily a subspace).

$$\mathcal{A} \models \mathbf{I}\ f$$

Decidability results

Can check $\mathbf{G}\ f, \mathbf{U}\ f$ and $\mathbf{I}\ f$ if f is positive.

Decidability results

Can check $\mathbf{G}\ f, \mathbf{U}\ f$ and $\mathbf{I}\ f$ if f is positive.

Undecidability: if f is negative or for $\mathbf{F}\ f$.

What else?

- QMC
- QMDPs

What else?

- QMC
- QMDPs
- Quantum AI and ML

What else?

- QMC
- QMDPs
- Quantum AI and ML
- Quantum Annealing

Ideas

- Quantum algorithm for reachability in a graph.

Ideas

- Quantum algorithm for reachability in a graph.
- Ranking functions for quantum programs.

Ideas

- Quantum algorithm for reachability in a graph.
- Ranking functions for quantum programs.
- Define strategy in a different way for QMDPs.

Ideas

- Quantum algorithm for reachability in a graph.
- Ranking functions for quantum programs.
- Define strategy in a different way for QMDPs.
- Can try to solve some optimization problems using quantum annealing.

Ideas

- Quantum algorithm for reachability in a graph.
- Ranking functions for quantum programs.
- Define strategy in a different way for QMDPs.
- Can try to solve some optimization problems using quantum annealing.
- Quantum AI and ML.

Thank You!